Decision Trees and Flow Graphs

Zdzisław Pawlak

Institute for Theoretical and Applied Informatics
Polish Academy of Sciences
ul. Bałtycka 5, 44-100 Gliwice, Poland
and
Warsaw School of Information Technology
ul. Newelska 6, 01-447 Warsaw, Poland
zpw@ii.pw.edu.pl

Abstract. We consider association of decision trees and flow graphs, resulting in a new method of decision rule generation from data, and giving a better insight in data structure. The introduced flow graphs can also give a new look at the conception of probability. We show that in some cases the conception of probability can be eliminated and replaced by a study of deterministic flows in a flow network.

1 Introduction

Decision tree is a very useful concept in computer science [7,9], decision science [2], probability [11] and others.

In this paper, we propose to associate with a decision tree another kind of graph, called flow graph, which gives better insight in data structure than the corresponding decision tree and reveals very interesting novel properties of decision trees, not visible directly from the tree. They can be used in many ways and, particularly, enable an efficient generation of decision rules from data.

Besides, the introduced flow graphs can also be used as a new look at the conception of probability. Łukasiewicz [6] claimed that probability defined by Laplace [5] and used today, is not a well defined concept and he proposed to base probability calculus on logical ground, which gives to probability sound mathematical foundations. Similar ideas have been proposed independently many years after Łukasiewicz by Carnap [3], Adams [1], Reichebach [10] and others.

We go a little bit farther and intend to show that in some cases the conception of probability can be eliminated and replaced by a study of deterministic flows in a flow network. The proposed approach gives a new method of decision rule generation from data, and permits to study data structure in a new way.

The paper is a continuation of some author's ideas presented in [8].

2 An Example

First, we explain our basic ideas by means of a simple example. Consider the set U of play blocks having various shapes (e.g., square, round), sizes (e.g., large,

S. Greco et al. (Eds.): RSCTC 2006, LNAI 4259, pp. 1–11, 2006.

small) and colors (e.g., black, white). Assume that the relation between different play blocks is given by a decision tree as shown in Fig.1. We will use standard terminology concerning decision trees, like root, branches, paths, etc.

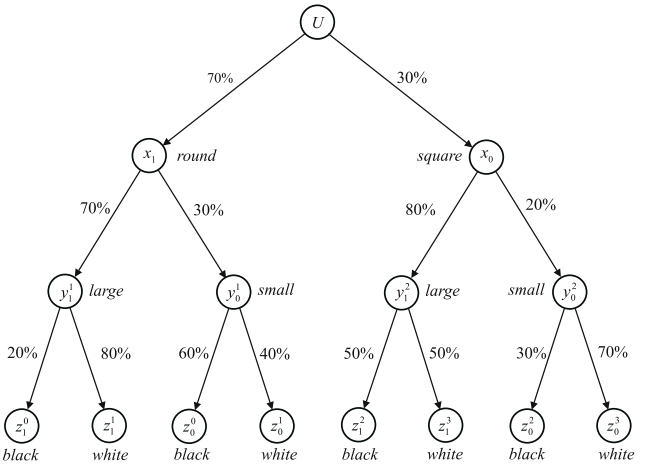

Fig. 1. Decision tree showing relations between different play blocks

The decision tree reveals statistical relationship between various types of play blocks. For example, the decision tree shows that there are 70% round and 30% square blocks in the set and among round blocks there are 70% large and 30% small blocks, whereas square blocks consist of 80% large and 20% small blocks. Moreover, the set of round and large blocks consists of 20% white and 80% black blocks, etc. In other words, the decision tree can be understood as a statistical data structure representation of the set U.

With every decision tree we can associate uniquely another graph, called a flow graph. The flow graph is an oriented graph obtained from a decision tree by removing the root and merging nodes labeled by the same "attribute", e.g. *small, large*, etc., as shown in Fig. 2.

The resulting flow graph is given in Fig. 3.

The flow graph reveals the relational structure among objects of the universe. For example, if the branch (*square, small*) is labeled by the number 0.06 it means that there are 6% objects in the universe which are square and small - the number 0.06 is computed from the data given in the decision tree.

Each path in the flow graph determines an "*if ..., then...*" decision rule. E.g., the path (*square, large, white*) determines a decision rule "*if square and large, then white*". In our approach, the number (percentage) associated with every branch can be interpreted as a flow intensity through the branch and used to study properties of decision rules. We can also interpret the flow graph in terms of probability, but we will refrain from this interpretation here and we claim that deterministic interpretation is more natural than the probabilistic one.

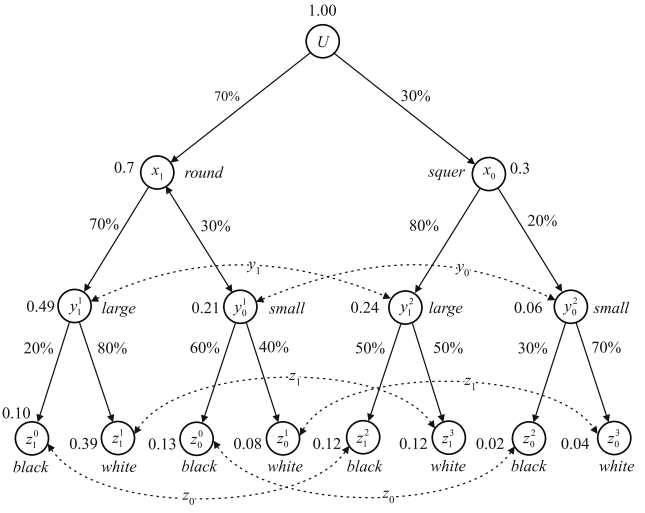

Fig. 2. Merging nodes labeled by the same "attribute"

In order to solve our problem we will analyze the structure of the flow graph in detail in the next section.

3 Flow Graphs – Basic Concepts

3.1 Flow Graphs

In this section we recall after [8] the fundamental concept of the proposed approach – a flow graph.

Flow graph is a *directed, acyclic, finite* graph $G = (N, \mathcal{B}, \sigma)$, where N is a set of *nodes*, $\mathcal{B} \subseteq N \times N$ is a set of *directed branches* and $\sigma : \mathcal{B} \to \langle 0, 1 \rangle$ is *a flow function* of (x, y) such that $\sigma(x, y)$ is a *strength* of (x, y). The strength of the branch expresses simply the percentage of a total flow through the branch.

Input of a node $x \in N$ *is the set* $I(x) = \{y \in N : (y, x) \in \mathcal{B}\}$; *output* of a node $x \in N$ is defined as $O(x) = \{y \in N : (x, y) \in \mathcal{B}\}$.

We will also need the concept of *input* and *output* of a graph G, defined, respectively, as: $I(G) = \{x \in N : I(x) = \emptyset\}$, $O(G) = \{x \in N : O(x) = \emptyset\}$.

Inputs and outputs of G are *external nodes* of G; other nodes are *internal nodes* of G.

If a flow graph G has only one input and every internal node of G has one input then such a flow graph with be called a *decision tree*.

Input of the decision tree will be referred to as *root*, whereas outputs - as *leaves* of the decision tree.

With every node x of a flow graph G we associate its *inflow* and *outflow* defined as

$$\sigma_+(x) = \sum_{y \in I(x)} \sigma(y, x) \tag{1}$$

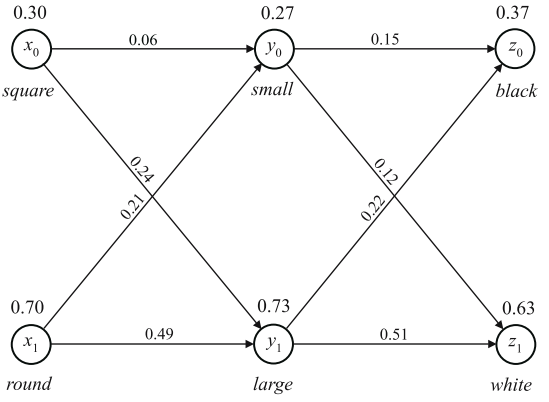

Fig. 3. Flow graph resulting from the decision tree

and

$$\sigma_-(x) = \sum_{y \in O(x)} \sigma(y, x). \tag{2}$$

For any internal node x, we have $\sigma_+(x) = \sigma_-(x) = \sigma(x)$, where $\sigma(x)$ is a *throughflow* of x. Moreover, let

$$\sigma_+(G) = \sum_{x \in I(G)} \sigma_-(x) \tag{3}$$

and

$$\sigma_-(G) = \sum_{x \in O(G)} \sigma_+(x). \tag{4}$$

Let us assume that $\sigma_+(G) = 1$, then $\sigma_+(G) = \sigma_-(G) = \sigma(G)$.

If we invert direction of all branches in G, then the resulting graph $G = (N, \mathcal{B}', \sigma')$ will be called an *inverted* graph of G. Of course, the inverted graph G' is also a flow graph and all inputs and outputs of G become inputs and outputs of G' , respectively.

3.2 Certainty and Coverage Factors

With every branch (x, y) of a flow graph G we associate the *certainty* and the *coverage factors*.

The *certainty* and the *coverage* of (x, y) are defined as

$$cer(x, y) = \frac{\sigma(x, y)}{\sigma(x)}, \tag{5}$$

and

$$cer(x,y) = \frac{\sigma(x,y)}{\sigma(y)}. \tag{6}$$

respectively.

Evidently, $cer(x,y) = cov(y,x)$, where $(x,y) \in \mathcal{B}$ and $(y,x) \in \mathcal{B}'$.

Certainty and coverage factors for the flow graph shown in Fig. 3 are presented in Fig. 4.

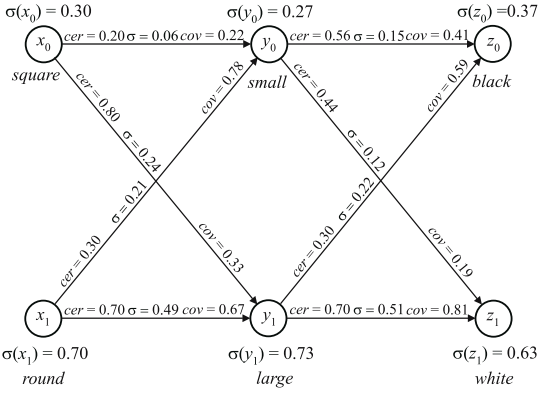

Fig. 4. Certainty and coverage factors

For every branch (x,y) of a decision tree $cov(x,y) = 1$.

Below, some properties, which are immediate consequences of definitions given above, are presented:

$$\sum_{y \in O(x)} cer(x,y) = 1, \tag{7}$$

$$\sum_{x \in I(y)} cov(x,y) = 1, \tag{8}$$

$$\sigma(x) = \sum_{y \in O(x)} cer(x,y)\sigma(x) = \sum_{y \in O(x)} \sigma(x,y), \tag{9}$$

$$\sigma(y) = \sum_{x \in I(y)} cov(x,y)\sigma(y) = \sum_{x \in I(y)} \sigma(x,y), \tag{10}$$

$$cer(x,y) = \frac{cov(x,y)\sigma(y)}{\sigma(x)}, \tag{11}$$

$$cov(x,y) = \frac{cer(x,y)\sigma(x)}{\sigma(y)}. \tag{12}$$

Obviously, the above properties have a probabilistic flavor, e.g., equations (9) and (10) have a form of total probability theorem, whereas formulas (11) and (12) are Bayes' rules. However, these properties in our approach are interpreted in a deterministic way and they describe flow distribution among branches in the network.

3.3 Paths, Connections and Fusion

A (*directed*) *path* from x to y, $x \neq y$ in G is a sequence of nodes $x_1, \ldots, x_n$ such that $x_1 = x$, $x_n = y$ and $(x_i, x_{i_1}) \in \mathcal{B}$ for every i, $1 \leq i \leq n-1$. A path from x to y is denoted by $[x \ldots y]$ and $n-1$ is called *length* of the path.

A flow graph is *linear* if all paths from node x to node y have the same length, for every pair of nodes x, y.

A set of nodes of a linear flow graph is called a *k-layer* if it consists of the set of all nodes of this graph linked by a path of the length k with some input node.

The set of all inputs of a flow graph will be called the *input layer* of the flow graph, whereas the set of all outputs of the flow graph is the *output layer* of the flow graph. For any input node x and output node y of a linear graph the length of the path $[x \ldots y]$ is the same. The layers different from the input layer and the output layer will be referred to as *hidden layers*.

In what follows we will interpret layers as attributes in an information system; input and hidden layers are interpreted as condition attributes, whereas output layer is interpreted as decision attribute.

The *certainty* of the path $[x_1 \ldots x_n]$ is defined as

$$cer[x_1 \ldots x_n] = \prod_{i=1}^{n-1} cer(x_i, x_{i+1}), \tag{13}$$

the *coverage* of the path $[x_1 \ldots x_n]$ is

$$cov[x_1 \ldots x_n] = \prod_{i=1}^{n-1} cov(x_i, x_{i+1}), \tag{14}$$

and the *strength* of the path $[x \ldots y]$ is

$$\sigma[x \ldots y] = \sigma(x) cer[x \ldots y] = \sigma(y) cov[x \ldots y]. \tag{15}$$

The set of all paths from x to $y (x \neq y)$ in G, denoted by $\langle x, y \rangle$, will be called a *connection* from x to y in G. In other words, connection $\langle x, y \rangle$ is a sub-graph of G determined by nodes x and y (see Fig. 5).

The *certainty* of the connection $\langle x, y \rangle$ is

$$cer\langle x, y \rangle = \sum_{[x \ldots y] \in \langle x, y \rangle} cer[x \ldots y], \tag{16}$$

the *coverage* of the connection $\langle x, y \rangle$ is

$$cov\langle x, y \rangle = \sum_{[x \ldots y] \in \langle x, y \rangle} cov[x \ldots y], \tag{17}$$

and the *strength* of the connection $\langle x, y\rangle$ is

$$\sigma\langle x, y\rangle = \sum_{[x\ldots y]\in\langle x,y\rangle} \sigma[x\ldots y] = \sigma(x)cer\langle x, y\rangle = \sigma(y)cov\langle x, y\rangle. \tag{18}$$

If we substitute simultaneously for every sub-graph $\langle x, y\rangle$ of a given flow graph G, where x is an input node and y an output node of G, a single branch (x, y) such that $\sigma(x, y) = \sigma\langle x, y\rangle$, then in the resulting graph G', called the *fusion* of G, we have $cer(x, y) = cer\langle x, y\rangle$, $cov(x, y) = cov\langle x, y\rangle$ and $\sigma(G) = \sigma(G')$.

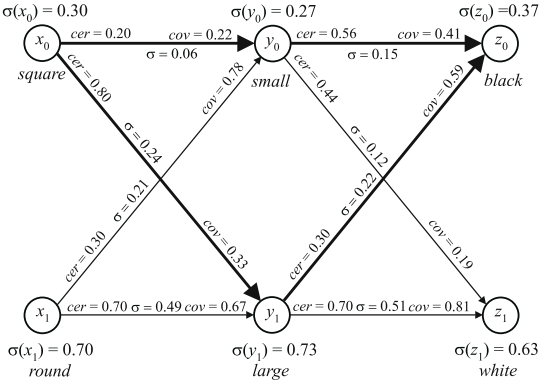

Fig. 5. Connection between x_0 and z_0

Thus fusion of a flow graph can be understood as a simplification of the graph and can be used to get a general picture of relationships in the flow graph (see Fig. 6).

3.4 Dependencies in Flow Graphs

Let x and y be nodes in a flow graph $G = (N, \mathcal{B}, \sigma)$, such that $(x, y) \in \mathcal{B}$.

Nodes x and y are *independent* in G if

$$\sigma(x, y) = \sigma(x)\sigma(y). \tag{19}$$

From (19) we get

$$\frac{\sigma(x, y)}{\sigma(x)} = cer(x, y) = \sigma(y), \tag{20}$$

and

$$\frac{\sigma(x, y)}{\sigma(y)} = cov(x, y) = \sigma(x). \tag{21}$$

If

$$cer(x, y) > \sigma(y), \tag{22}$$

or

$$cov(x, y) > \sigma(x), \tag{23}$$

x and y are *positively dependent* in G.

Similarly, if

$$cer(x, y) < \sigma(y), \tag{24}$$

or

$$cov(x, y) < \sigma(x), \tag{25}$$

then x and y are *negatively dependent* in G.

Relations of independency and dependencies are symmetric ones, and are analogous to those used in statistics.

For every branch $(x, y) \in \mathcal{B}$ we define a *dependency* (*correlation*) *factor* $\eta(x, y)$ defined as

$$\eta(x, y) = \frac{cer(x, y) - \sigma(y)}{cer(x, y) + \sigma(y)} = \frac{cov(x, y) - \sigma(x)}{cov(x, y) + \sigma(x)}. \tag{26}$$

Obviously, $-1 \leq \eta(x, y) \leq 1$; $\eta(x, y) = 0$ if and only if $cer(x, y) = \sigma(y)$ and $cov(x, y) = \sigma(x)$; $\eta(x, y) = -1$ if and only if $cer(x, y) = cov(x, y) = 0$; $\eta(x, y) = 1$ if and only if $\sigma(y) = \sigma(x) = 0$. Evidently, if $\eta(x, y) = 0$, then x and y are

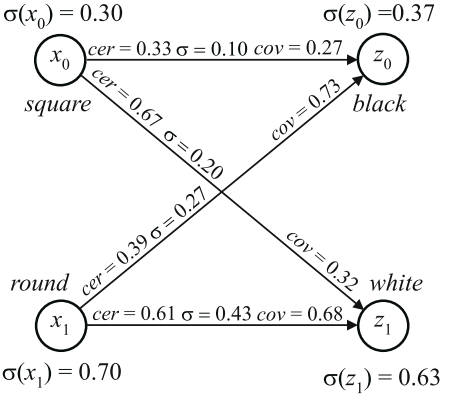

Fig. 6. Fusion of the flow graph

independent, if $-1 \leq \eta(x, y) < 0$, then x and y are negatively dependent, and if $0 < \eta(x, y) \leq 1$, then x and y are positively dependent (see Fig. 7). Thus, the dependency factor expresses a degree of dependency, and can be seen as a counterpart of correlation coefficient used in statistics.

4 Flow Graphs and Decision Algorithms

Flow graphs can be interpreted as decision algorithms. The most general case of this correspondence has been considered in [4].

Let us assume that the set of nodes of a flow graph is interpreted as a set of logical formulas. The formulas are understood as propositional functions and if x is a formula, then $\sigma(x)$ is to be interpreted as a truth value of the formula. Let us observe that the truth values are numbers from the closed interval $< 0, 1 >$, i.e., $0 \leq \sigma(x) \leq 1$.

These truth values can be also interpreted as probabilities. Thus $\sigma(x)$ can be understood as flow distribution ratio (percentage), truth value, or probability. We will stick to the first interpretation.

With every branch (x, y) we associate a decision rule $x \to y$, read as "*if x, then y*"; x will be referred to as *condition*, whereas y – *decision* of the rule. Such a rule is characterized by three numbers, $\sigma(x, y)$, $cer(x, y)$ and $cov(x, y)$.

Thus, every path $[x_1 \ldots x_n]$ determines a sequence of decision rules $x_1 \to x_2$, $x_2 \to x_3, \ldots, x_{n-1} \to x_n$.

From previous considerations it follows that this sequence of decision rules can be interpreted as a single decision rule $x_1 x_2 \ldots x_{n-1} \to x_n$, in short $x^* \to x_n$, where $x^* = x_1 x_2 \ldots x_{n-1}$, characterized by

$$cer(x^*, x_n) = \frac{\sigma(x^*, x_n)}{\sigma(x^*)}, \tag{27}$$

$$cov(x^*, x_n) = \frac{\sigma(x^*, x_n)}{\sigma(x_n)}, \tag{28}$$

and

$$\sigma(x^*, x_n) = \sigma(x^*) cer(x_{n-1}, x_n), \quad \sigma(x^*) = \sigma[x_1, \ldots, x_{n-1}]. \tag{29}$$

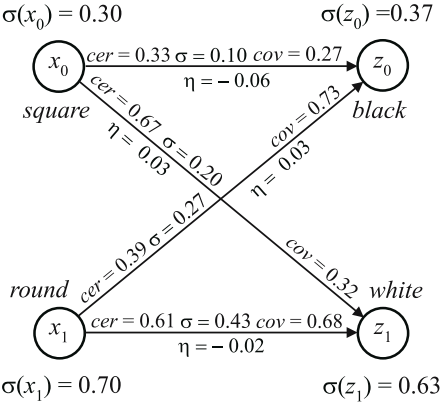

Fig. 7. Dependencies in the flow graph

The set of all decision rules $x_{i_1}x_{i_2}\ldots x_{i_{n-1}} \rightarrow x_{i_n}$ associated with all paths $[x_{i_n}\ldots x_{i_n}]$, such that x_{i_1} and x_{i_n} are input and output of the flow graph, respectively, will be called a *decision algorithm* induced by the flow graph.

The decision algorithm induced by the flow graph shown in Fig. 4 is shown in Table 1. The corresponding flow graph, and the dependency between conditions

Table 1. The decision algorithm induced by the flow graph

	certainty	coverage	strength
if square and small, then black	0.50	0.08	0.03
if square and small, then white	0.50	0.05	0.03
if square and large, then black	0.29	0.19	0.07
if square and large, then white	0.71	0.27	0.17
if round and small, then black	0.57	0.32	0.12
if round and small, then white	0.43	0.14	0.09
if round and large, then black	0.31	0.41	0.15
if round and large, then white	0.69	0.54	0.34

and decision in each decision rule are shown in Fig. 8.

It is interesting to compare diagrams shown in Fig. 1 and Fig. 8. Both diagrams show internal structure (relations) between various groups of play blocks. The decision tree reveals simple statistical structure of the relationship, whereas the flow graph gives much deeper insight into the relationship, and enables simple decision rule generation.

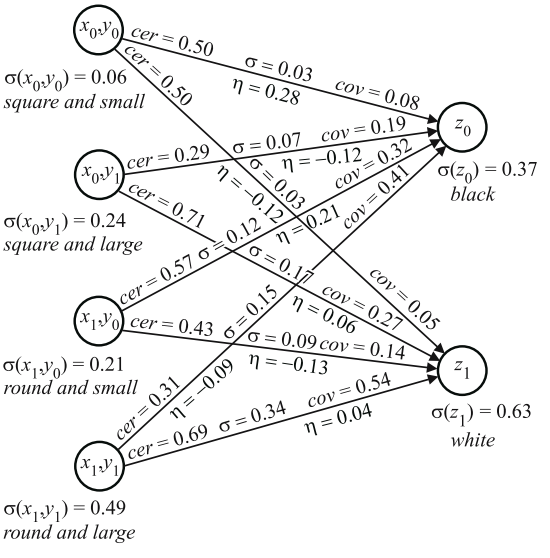

Fig. 8. Flow graph for the decision algorithm

5 Conclusions

Decision tree is an important concept, widely used in computer science, knowledge discovery from data, decision analysis, probability and others. In this paper, with every decision tree we associate another kind of graph, called a flow graph, which reveals deeper insight in data structure associated with a decision tree. This leads to novel methods of decision rule generation from data, and gives better look into decision process analysis. Besides, the proposed approach throws new light on the conception of probability.

References

1. Adams, E. A.: The Logic of Conditionals, an Application of Probability to Deductive Logic, D. Reidel Publishing Company, Dordrecht, Boston, 1975
2. Bernardo, J. M., M. Smith, A. F.: Bayesian Theory. Wiley series in probability and mathematical statistics. John Wiley & Sons, Chichester, New York, Brisbane, Toronto, Singapore, 1994
3. Carnap, R.: Logical Foundation of Probability, Routlege and Kegan Paul, London, 1950
4. Greco, S., Pawlak, Z., Słowiński, R.: Generalized decision algorithms, rough inference rules and flow graphs. In: J. J. Alpigini, J. F. Peters, A. Skowron, N. Zhong (eds.), Rough Sets and Current Trends in Computing. Lecture Notes in Artificial Intelligence 2475, Springer-Verlag, Berlin, 2002, pp. 93-104
5. Laplace, P. S.: Théorie Analytique des Probabilités, Paris, 1812
6. Łukasiewicz, J.: Die logischen Grundlagen der Wahrscheinilchkeitsrechnung. Kraków, 1913. In: L. Borkowski (ed.), Jan Łukasiewicz Selected Works, North Holland Publishing Company, Amsterdam, London, Polish Scientific Publishers, Warsaw, 1970, pp. 16-63
7. Moshkov, M.: On time complexity of decision trees. In: L. Polkowski, A, Skowron (eds.), Rough Sets in Knowledge Discovery 1, Physica-Verlag, Heidelberg, 1998, pp. 160-191
8. Pawlak, Z.: Flow graphs and data mining. In: J. F. Peters and A. Skowron (eds.), Transaction on Rough Sets III, LNCS 3400, Springer-Verlag, Berlin, 2005, pp. 1-36
9. Quinlan, J. R.: C4.5: Programs for machine learning. Morgan Kaufmann, San Mateo, CA, 1993
10. Reichenbach, H.: Wahrscheinlichkeitslehre: eine Untersuchung ber die logischen und mathematischen Grundlagen der Wahrscheinlichkeitsrechnung, 1935; (English translation: The theory of probability, an inquiry into the logical and mathematical foundations of the calculus of probability), University of California Press, Berkeley, 1948
11. Shafer, G.: The Art of Causal Conjecture, The MIT Press, Cambridge, Massachusetts, London, England, 1996