Decision Networks

Zdzisław Pawlak[1,2]

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

Abstract. A decision network is a finite, directed acyclic graph, nodes of which represent logical formulas, whereas branches - are interpreted as decision rules. Every path in the graph represents a chain of decision rules, which describe compound decision.
Some properties of decision networks will be given and a simple example will illustrate the presented ideas and show possible applications.

Keywords: decision rules, decision algorithms, decision networks

1 Introduction

The main problem in data mining consists in discovering patterns in data. The patterns are usually expressed in form of decision rules, which are logical expressions in the form "*if* Φ *then* Ψ", where Φ and Ψ are logical formulas (propositional functions) used to express properties of objects of interest. Any set of decision rules is called a *decision algorithm*. Thus knowledge discovery from data consists in representing hidden relationships between data in a form of decision algorithms. However, for some applications, it is not enough to give only set of decision rules describing relationships in the database. Sometimes also knowledge of relationship between decision rules is necessary in order to understand better data structures. To this end we propose to employ a decision algorithm in which also relationship between decision rules is pointed out, called a *decision network*.

The decision network is a finite, directed acyclic graph, nodes of which represent logical formulas, whereas branches – are interpreted as decision rules. Thus every path in the graph represents a chain of decisions rules, which will be used to describe compound decisions.

Some properties of decision networks will be given and a simple example will be used to illustrate the presented ideas and show possible applications.

2 Decision Networks and Decision Rules

Let U be a non empty finite set, called the *universe* and let Φ , Ψ be logical formulas. The meaning of Φ in U, denoted by $|\Phi|$, is the set of all elements

S. Tsumoto et al. (Eds.): RSCTC 2004, LNAI 3066, pp. 1–7, 2004.

of U, that satisfies Φ in U. The truth value of Φ denoted $val(\Phi)$ is defined as $card|\Phi|/card(U)$, where $card(X)$ denotes cardinaity of X.

By *decision network* over $S = (U, \mathcal{F})$ we mean a pair $N = (\mathcal{F}, \mathcal{R})$, where $\mathcal{R} \subseteq \mathcal{F} \times \mathcal{F}$ is a binary relation, called a *consequence relation* and $\mathcal{F}$ is a set of logical formulas.

Any pair $(\Phi, \Psi) \in \mathcal{R}, \Phi \neq \Psi$ is referred to as a decision rule (in N).

We assume that S is known and we will not refer to it in what follows.

A *decision rule* (Φ, Ψ) will be also presented as an expression $\Phi \rightarrow \Psi$, read *if* Φ *then* Ψ, where Φ and Ψ are referred to as *predesessor* (*conditions*) and *successor* (*decisions*) of the rule, respectively.

The number $supp(\Phi, \Psi) = card(|\Phi \wedge \Psi|)$ will be called a *support* of the rule $\Phi \rightarrow \Psi$. We will consider nonvoid decision rules only, i.e., rules such that $supp(\Phi, \Psi) \neq 0$.

With every decision rule $\Phi \rightarrow \Psi$ we associate its *strength* defined as

$$str(\Phi, \Psi) = \frac{supp(\Phi, \Psi)}{card(U)}. \tag{1}$$

Moreover, with every decision rule $\Phi \rightarrow \Psi$ we associate the *certainty factor* defined as

$$cer(\Phi, \Psi) = \frac{str(\Phi, \Psi)}{val(\Phi)} \tag{2}$$

and the *coverage factor* of $\Phi \rightarrow \Psi$

$$cov(\Phi, \Psi) = \frac{str(\Phi, \Psi)}{val(\Psi)}, \tag{3}$$

where $val(\Phi) \neq 0$ and $val(\Psi) \neq 0$.

The coefficients can be computed from data or can be a subjective assessment.

We assume that

$$val(\Phi) = \sum_{\Psi \in Suc(\Phi)} str(\Phi, \Psi) \tag{4}$$

and

$$val(\Psi) = \sum_{\Phi \in Pre(\Psi)} str(\Phi, \Psi), \tag{5}$$

where $Suc(\Phi)$ and $Pre(\Psi)$ are sets of all succesors and predecessors of the corresponding formulas, respectively.

Consequently we have

$$\sum_{Suc(\Phi)} cer(\phi, \Psi) = \sum_{Pre(\Psi)} cov(\Phi, \Psi) = 1. \tag{6}$$

If a decision rule $\Phi \rightarrow \Psi$ uniquely determines decisions in terms of conditions, i.e., if $cer(\Phi, \Psi) = 1$, then the rule is *certain*, otherwise the rule is *uncertain*.

If a decision rule $\Phi \rightarrow \Psi$ covers all decisions, i.e., if $cov(\Phi, \Psi) = 1$ then the decision rule is *total*, otherwise the decision rule is *partial*.

Immediate consequences of (2) and (3) are:

$$cer(\Phi,\Psi) = \frac{cov(\Phi,\Psi)val(\Psi)}{val(\Psi)}, \tag{7}$$

$$cov(\Phi,\Psi) = \frac{cer(\Phi,\Psi)val(\Phi)}{val(\Psi)}. \tag{8}$$

Note, that (7) and (8) are Bayes' formulas. This relationship first was observed by Łukasiewicz [1].

Any sequence of formulas $\Phi_1,\ldots,\Phi_n$, $\Phi_i \in \mathcal{F}$ and for every i, $1 \le i \le n-1$, $(\Phi_i,\Phi_{i+1}) \in \mathcal{R}$ will be called a *path* from Φ_1 to Φ_n and will be denoted by $[\Phi_1 \ldots \Phi_n]$.

We define

$$cer[\Phi_1 \ldots \Phi_n] = \prod_{i=1}^{n-1} cer[\Phi_i,\Phi_{i+1}], \tag{9}$$

$$cov[\Phi_1 \ldots \Phi_n] = \prod_{i=1}^{n-1} cov[\Phi_i,\Phi_{i+1}], \tag{10}$$

$$str[\Phi_1 \ldots \Phi_n] = val(\Phi_1)cer[\Phi_1 \ldots \Phi_n] = val(\Phi_n)cov[\Phi_1 \ldots \Phi_n]. \tag{11}$$

The set of all paths form Φ to Ψ, denoted $< \Phi,\Psi >$, will be called a *connection* from Φ to Ψ.

For connection we have

$$cer < \Phi,\Psi >= \sum_{[\Phi\ldots\Psi]\in<\Phi,\Psi>} cer[\Phi\ldots\Psi], \tag{12}$$

$$cov < \Phi,\Psi >= \sum_{[\Phi\ldots\Psi]\in<\Phi,\Psi>} cov[\Phi\ldots\Psi], \tag{13}$$

$$\begin{aligned} str < \Phi,\Psi > &= \sum_{[\Phi\ldots\Psi]\in<\Phi,\Psi>} str[\Phi\ldots\Psi] = \\ &= val(\Phi)cer < \Phi,\Psi >= val(\Psi)cov < \Phi,\Psi > . \end{aligned} \tag{14}$$

With every decision network we can associate a flow graph [2, 3]. Formulas of the network are interpreted as nodes of the graph, and decision rules – as directed branches of the flow graph, whereas strength of a decision rule is interpreted as flow of the corresponding branch.

3 Independence of Formulas

Independency of logical formulas considered in this section first was proposed by Łukasiewicz [1].

Let $\Phi \to \Psi$ be a decision rule. Formulas Φ and Ψ are *independent* on each other if

$$str(\Phi, \Psi) = val(\Phi)val(\Psi). \tag{15}$$

Consequently

$$\frac{str(\Phi, \Psi)}{val(\Phi)} = cer(\Phi, \Psi) = val(\Psi) \tag{16}$$

and

$$\frac{str(\Phi, \Psi)}{val(\Psi)} = cov(\Phi, \Psi) = val(\Phi). \tag{17}$$

If

$$cer(\Phi, \Psi) > val(\Psi) \tag{18}$$

or

$$cov(\Phi, \Psi) > val(\Phi), \tag{19}$$

then Φ and Ψ *depend positively* on each other. Similarly, if

$$cer(\Phi, \Psi) < val(\Psi) \tag{20}$$

or

$$cov(\Phi, \Psi) < val(\Phi), \tag{21}$$

then Φ and Ψ *depend negatively* on each other.

Let us observe that relations of independency and dependency are symmetric ones, and are analogous to that used in statistics.

For every decision rule $\Phi \to \Psi$ we define a *dependency factor* $\eta(\Phi, \Psi)$ defined as

$$\eta(\Phi, \Psi) = \frac{cer(\Phi, \Psi) - val(\Psi)}{cer(\Phi, \Psi) + val(\Psi)} = \frac{cov(\Phi, \Psi) - val(\Phi)}{cov(\Phi, \Psi) + val(\Phi)}. \tag{22}$$

It is easy to check that if $\eta(\Phi, \Psi) = 0$, then Φ and Ψ are independent on each other, if $-1 < \eta(\Phi, \Psi) < 0$, then Φ and Ψ are negatively dependent and if $0 < \eta(\Phi, \Psi) < 1$ then Φ and Ψ are positively dependent on each other. Thus the dependency factor expresses a degree of dependency, and can be seen as a counterpart of correlation coefficient used in statistics.

Another dependency factor has been proposed in [4].

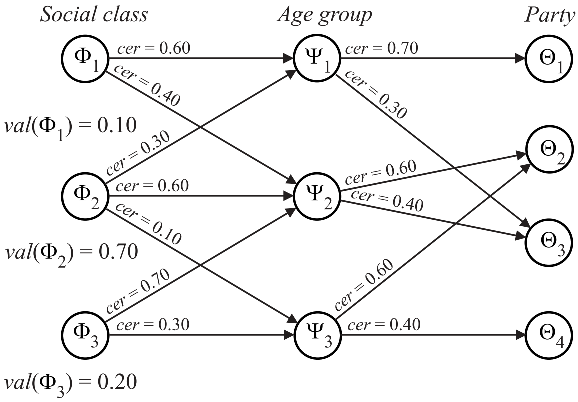

Fig. 1. Initial votes distribution.

4 An Example

Consider three disjoint age groups of voters Ψ_1 (*old*), Ψ_2 (*middle aged*) and Ψ_3 (*young*) – belonging to three social classes Φ_1 (*high*), Φ_2 (*middle*) and Φ_3 (*low*). The voters voted for four political parties Θ_1 (*Conservatives*), Θ_2 (*Labor*), Θ_3 (*Liberal Democrats*) and Θ_4 (*others*).

Social class and age group votes distribution is shown in Fig. 1.

First, we compute, employing formula (2), strength of each branch joining *Social class* and *Age group*. Having done this we can compute coverage factors for each *Age group* and using formula (5) we compute $val(\Psi_i)$.Repeating this procedure for *Age group* and *Party* we get results shown in Fig.2.

From the decision network presented in Fig. 2 we can see that, e.g., party Θ_1 obtained 19% of total votes, all of them from age group Ψ_1; party Θ_2 – 44% votes, which 82% are from age group Ψ_2 and 18% – from age group Ψ_3, etc.

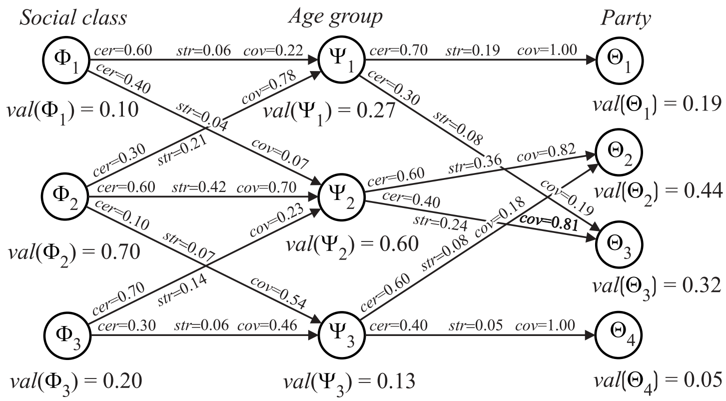

Fig. 2. Final votes distribution.

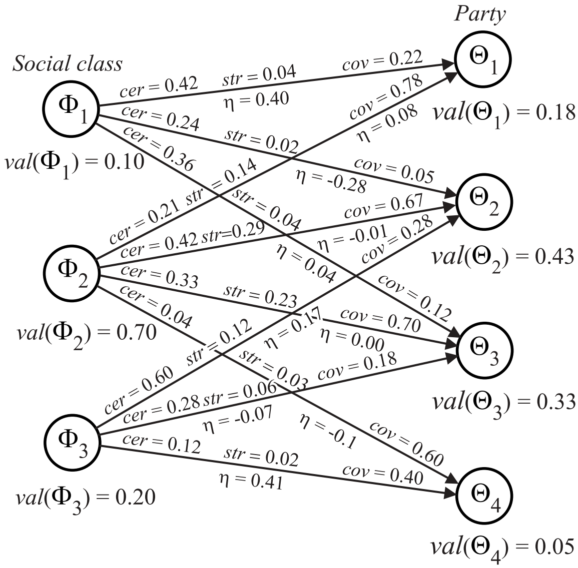

Fig. 3. Simplified decision network.

If we want to know how votes are distributed between parties with respects to social classes we have to eliminate age groups from the decision network. Employing formulas (9),...,(14) we get results shown in Fig. 3.

From the decision network presented in Fig. 3 we can see that party Θ_1 obtained 22% votes from social class Φ_1 and 78% from social class Φ_2, etc.

We can also present the obtained results employing decision algorithms. For simplicity we present only some decision rules of the decision algorithm. For example, from Fig.2 we obtain decision rules:

If Party (Θ_1) *then Age group* (Ψ_1) (0.19)
If Party (Θ_2) *then Age group* (Ψ_2) (0.36)
If Party (Θ_2) *then Age group* (Ψ_3) (0.08), etc.

The number at the end of each decision rule denotes strength of the rule. Similarly, from Fig.3 we get:

If Party (Θ_1) *then Soc. class* (Φ_1) (0.04)
If Party (Θ_1) *then Soc. class* (Φ_2) (0.14), etc.

We can also invert decision rules and, e.g., from Fig. 3 we have:

If Soc. class (Φ_1) *then Party* (Θ_1) (0.04)
If Soc. class (Φ_1) *then Party* (Θ_2) (0.02)
If Soc. class (Φ_1) *then Party* (Θ_3) (0.04), etc

In Fig. 3 values of dependency factors are also shown. It can be seen from the diagram that e.g., Φ_1 and Θ_1 are positively dependent ($\eta = 0.40$), whereas Φ_3 and Θ_3 are negatively dependent ($\eta = -0.07$). That means that there is relatively strong positive dependency between high social class and Conservatives, whereas there is very low negative dependency between low social class and Liberal Democrats.

5 Conclusion

In this paper a concept of decision network is introduced and examined. Basic properties of decision networks are given and their application to decision analysis is shown. Simple tutorial example at the end of the paper shows the possible application of the introduced ideas.

References

1. Łukasiewicz, J.: Die logishen Grundlagen der Wahrscheinilchkeitsrechnung. Kraków (1913), in: L. Borkowski (ed.), Jan Łukasiewicz – Selected Works, North Holland Publishing Company, Amsterdam, London, Polish Scientific Publishers, Warsaw (1970) 16-63
2. Pawlak, Z.: Probability, Truth and Flow Graphs, in: RSKD – International Workshop and Soft Computing, ETAPS 2003, A. Skowron, M. Szczuka (eds.), Warsaw (2003) 1-9
3. Pawlak, Z.: Flow graphs and decision algorithms, in: G. Wang, Q. Liu, Y. Y. Yao, A. Skowron (eds.), Proceedings of the Ninth International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing RSFDGrC'2003), Chongqing, China, May 26-29, 2003, LNAI 2639, Springer-Verlag, Berlin, Heidelberg, New York, 1-11
4. Słowiński, R., Greco, S.: A note on dependency factor (manuscript).