Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
European Journal of Operational Research 154 (2004) 184–190
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
www.elsevier.com/locate/dsw

Computing, Artificial Intelligence and Information Technology

Decisions rules and flow networks

Zdzisław Pawlak *

Institute of Theoretical and Applied Informatics, Polish Academy of Sciences, ul. Bałtycka 5, 44 100 Gliwice, Poland

Received 9 September 2002; accepted 9 December 2002

For late Professor Henryk Greniewski, my mentor

Abstract

This paper, which is continuation of a series of the author's papers on the relationship between decision algorithms and Bayes' theorem, is related to Łukasiewicz's ideas concerning the relationship between multivalued logic, probability and Bayes' theorem. We propose in this paper a new mathematical model of a flow network, different from that introduced by Ford and Fulkerson. Basically, the presented model is intended to be used rather as a mathematical model of decision processes than as a tool for flow optimization in networks. Moreover, it concerns rather flow of information than material media. Branches of the network are interpreted as decision rules with elementary conditions and decisions in the nodes, whereas the whole network represents a decision algorithm. It is shown that a flow in such networks is governed by Bayes' formula. In this case, however, the formula describes deterministic information flow distribution among branches of the network, without referring to its probabilistic character. This leads to a new look on Bayes' formula and many new applications.

Keywords: Decision rules; Flow networks; Bayes' theorem

1. Introduction

This paper is an extension of the article [7] and is a continuation of ideas presented in author's previous papers on rough sets, Bayes' theorem and decision tables [8].

In [4] flow optimization in networks has been introduced and studied. The model was intended to capture the nature of flow in transportation or communication networks.

In this paper, we present another kind of mathematical model for flow networks, which may be interpreted rather as a model of a deterministic, steady state flow in a plumbing network—than a transportation network. Essentially, the model is intended to be used as a description of decision processes and not as a description of flow optimization. Branches of the network are interpreted as decision rules with elementary conditions and decisions in the nodes, whereas the network is supposed to describe a decision algorithm. It is shown that a flow in such a network is governed by Bayes' rule. Furthermore, this interpretation brings to light another understanding of Bayes'

* Address: ul. Zuga 29, Warszawa 01-806, Poland. Tel.: +48-22-8345659; fax: +48-22-8251635.
E-mail address: zpw@ii.pw.edu.pl (Z. Pawlak).

0377-2217/$ - see front matter
doi:10.1016/S0377-2217(03)00004-3

rule: the rule may be interpreted entirely in a deterministic way, without referring to its probabilistic nature, inherently associated with classical Bayesian philosophy. This leads to new philosophical and practical consequences. Some of them will be discussed in this paper.

The plan of the paper is the following: First, the definition of the flow graph is introduced, and next basic properties of flow graphs are given and discussed. Further, simplification of flow graphs is formulated and analyzed. Finally, the relationship between flow graphs and decision algorithms (tables) is presented. A simple tutorial example is used to illustrate the ideas presented in the paper.

2. Flow graphs

A flow graph is a *directed*, *acyclic*, *finite* graph $G = (N, \mathcal{B}, \varphi)$, where N is a set of *nodes*, $\mathcal{B} \subseteq N \times N$ is a set of *directed branches*, $\varphi : \mathcal{B} \to R^+$ is a *flow function* and R^+ is the set of non-negative reals.

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 follows:

$$I(G) = \{x \in N : I(x) = \emptyset\},$$
$$O(G) = \{x \in N : O(x) = \emptyset\}.$$

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

If $(x,y) \in \mathcal{B}$ then $\varphi(x,y)$ is a *troughflow* from x to y.

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

$$\varphi_+(y) = \sum_{x \in I(y)} \varphi(x,y),$$
$$\varphi_-(x) = \sum_{y \in O(x)} \varphi(x,y),$$

respectively.

Similarly, we define an inflow and an outflow for the whole flow graph, which are defined as

$$\varphi_+(G) = \sum_{x \in I(G)} \varphi_-(x),$$
$$\varphi_-(G) = \sum_{x \in O(G)} \varphi_+(x),$$

respectively.

We assume that for any internal node x, $\varphi_+(x) = \varphi_-(x) = \varphi(x)$, where $\varphi(x)$ is a *troughflow* of node x.

Obviously, $\varphi_+(G) = \varphi_-(G) = \varphi(G)$, where $\varphi(G)$ is a *troughflow* of graph G.

The above formulas can be considered as *flow conservation equations* [4].

3. Strength, certainty and coverage factors

With every branch (x,y) we associate its strength defined as

$$\sigma(x,y) = \varphi(x,y)/\varphi(G).$$

Obviously, $0 \leqslant \sigma(x,y) \leqslant 1$ and it can be considered as a normalized flow of the branch (x,y). The strength of a branch expresses simply the percentage of a total flow through the branch.

We define now two important coefficients assigned to every branch of a flow graph—the *certainty* and the *coverage factors*.

The *certainty* and the *coverage* of (x,y) are defined as $\mathrm{cer}(x,y) = \sigma(x,y)/\sigma(x)$, and the $\mathrm{cov}(x,y) = \sigma(x,y)/\sigma(y)$, respectively, where $\sigma(x)$ is the *normalized troughflow* of x, defined as

$$\sigma(x) = \sum_{y \in O(x)} \sigma(x,y) = \sum_{y \in I(x)} \sigma(y,x)$$

and $\sigma(y)$ is defined in a similar way.

The certainty factor describes outflow distribution between outputs of a node, whereas the coverage factor reveals how inflow is distributed between inputs of the node.

The below properties are immediate consequences of definitions given in the preceding section:

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

$$\sum_{x \in I(y)} \mathrm{cov}(x,y) = 1, \tag{2}$$

$$\mathrm{cer}(x,y) = \mathrm{cov}(x,y)\sigma(y)/\sigma(x), \qquad (3)$$

$$\mathrm{cov}(x,y) = \mathrm{cer}(x,y)\sigma(x)/\sigma(y). \qquad (4)$$

It is easily seen that the strength, certainty, coverage factors and consequently properties (1)–(4) have a probabilistic flavor. In particular, Eqs. (3) and (4) are well known Bayes' formulas. However, in our case the properties are interpreted without referring to their probabilistic character. They simply describe some features of flow distribution among branches in the network.

Let us also observe that Bayes' formula is, in our setting, expressed by means of the strength coefficient. This leads to very simple computations and gives also new insight into the meaning of Bayesian methodology.

Remark. It is worthwhile to mention that the certainty and coverage factors have been for a long time used in another context in machine learning and data bases, see e.g., [9,10]. In fact these coefficients have been first used by Łukasiewicz in connection with his study of logic, probability and Bayes' theorem [5].

Example. Suppose that cars are painted in two colors y_1 and y_2 and that 60% of cars have color y_1, whereas 40% cars have color y_2. Moreover, assume that colors y_1 and y_2 can be obtained by mixing three paints x_1, x_2 and x_3 in the following proportions:

- y_1 contains 20% of x_1, 70% of x_2 and 10% of x_3,
- y_2 contains 30% of x_1, 50% of x_2 and 20% of x_3.

We have to find demand of each paint and its distribution among cars y_1 and y_2.

Employing terminology introduced in previous section we can represent our problem by means of flow graph shown in Fig. 1.

Thus in order to solve our task first we have to compute strength of each branch. Next, we compute demand of each paint. Finally, we compute the distribution of each paint among colors of cars.

The final result is presented in Fig. 2.

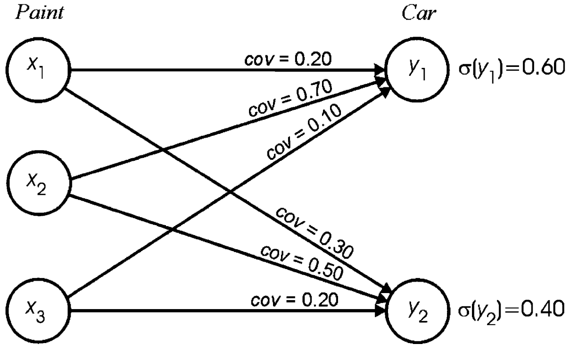

Fig. 1. Paint demand.

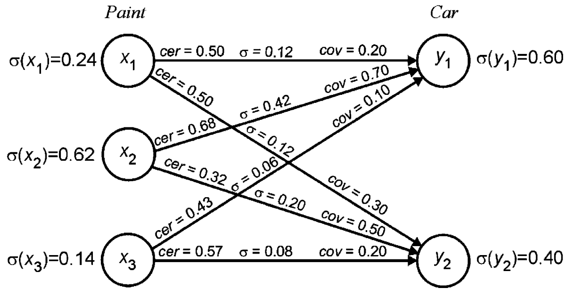

Fig. 2. Paint distribution.

Suppose now that the cars are produced by three manufacturers z_1, z_2 and z_3, in proportions shown in Fig. 3, i.e.,

- 50% of cars y_1 are produced by manufacturer z_1,
- 30% of cars y_1 are produced by manufacturer z_2,
- 20% of cars y_1 are produced by manufacturer z_3

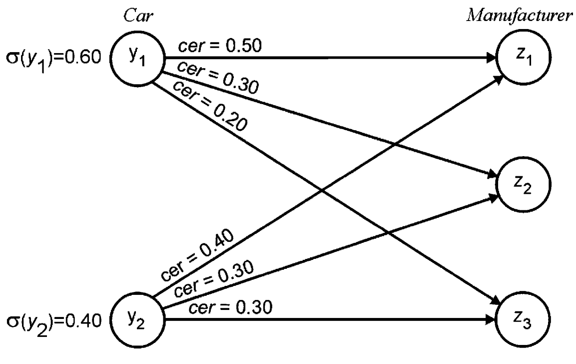

Fig. 3. Car production distribution.

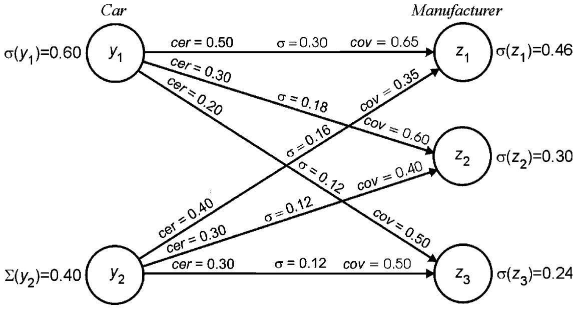

Fig. 4. Manufactures distribution.

and

- 40% of cars y_2 are produced by manufacturer z_1,
- 30% of cars y_2 are produced by manufacturer z_2,
- 30% of cars y_2 are produced by manufacturer z_3.

Employing the technique used previously, we can compute car production distribution among manufacturers as shown in Fig. 4, e.g., manufacturer z_1 produces 65% of cars y_1 and 35% of cars y_2 etc. Finally, the manufacturer z_1 produces 46% of cars, manufacturer z_2—30% of cars and manufacturer z_3—24% of cars.

We can combine graphs shown in Figs. 2 and 4 and we obtain the flow graph shown in Fig. 5.

The graph shows clearly the flow structure of the whole production process. From this graph it is easily seen how the flow of decisions is structured.

4. Paths and connections

For many applications we will need generalization of the strength, the certainty and coverage factors, which will be discussed next.

A (*directed*) *path* from x to y, $x \neq y$ denoted $[x, y]$, is a sequence of nodes $x_1, \ldots, x_n$ such that $x_1 = x$, $x_n = y$ and $(x_i, x_{i+1}) \in \mathscr{B}$ for every i, $1 \leqslant i \leqslant n-1$.

Now, we extend the concept of certainty, coverage and strength from single branch to a path, as shown below.

The *certainty* of a path $[x_1, x_n]$ is defined as

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

the *coverage* of a path $[x_1, x_n]$ is the following

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

the *strength* of a path $[x, y]$ is

$$\sigma[x, y] = \sigma(x)\text{cer}[x, y] = \sigma(y)\text{cov}[x, y].$$

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

We will also need extension of the above coefficients for connections (i.e., sub-graphs determined by nodes x and y) as shown in what follows:

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

$$\text{cer}\langle x, y \rangle = \sum_{[x,y] \in \langle x,y \rangle} \text{cer}[x, y],$$

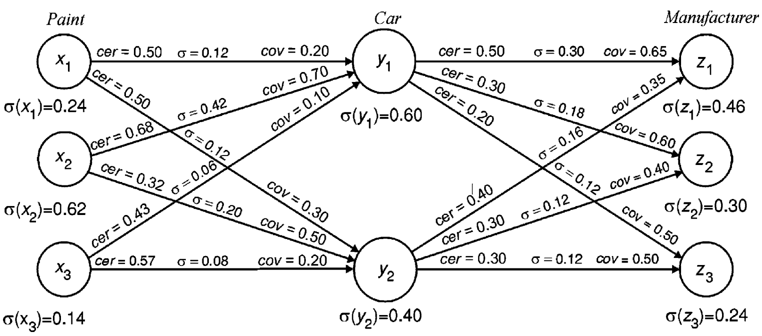

Fig. 5. Combined flow graph.

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

$$\mathrm{cov}\langle x, y\rangle = \sum_{[x,y]\in\langle x,y\rangle} \mathrm{cov}[x, y],$$

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

$$\sigma\langle x, y\rangle = \sum_{[x,y]\in\langle x,y\rangle} \sigma[x, y].$$

Let x, y $(x \neq y)$ be nodes of G. If we substitute the sub-graph $\langle x, y\rangle$ by a single branch (x, y), such that $\sigma(x, y) = \sigma\langle x, y\rangle$, then $\mathrm{cer}(x, y) = \mathrm{cer}\langle x, y\rangle$, $\mathrm{cov}(x, y) = \mathrm{cov}\langle x, y\rangle$ and $\sigma(G) = \sigma(G')$, where G' is the graph obtained from G by substituting $\langle x, y\rangle$ by (x, y).

5. Flow graph and decision rules

With every branch (x, y) we associate a decision rule $x \to y$, read *if x then y*.

Thus every path $[x_1, x_n]$ determine 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 replaced by a single decision rule $x_1x_2, \ldots, x_{n-1} \to x_n$, in short $X_i \to x_n$, such that

$$\mathrm{cer}(X_i, x_n) = \mathrm{cer}[x_1, x_n],$$

$$\mathrm{cov}(X_i, x_n) = \mathrm{cov}[x_1, x_n]$$

and

$$\sigma(X_i, x_n) = \sigma(x_1)\mathrm{cer}[x_1, x_n] = \sigma(x_n)\mathrm{cov}[x_1, x_n].$$

Similarly, with every connection $\langle x, y\rangle$ we will associate a decision rule $x \to y$, such that

$$\mathrm{cer}(x, y) = \mathrm{cer}\langle x, y\rangle,$$

$$\mathrm{cov}(x, y) = \mathrm{cov}\langle x, y\rangle$$

and

$$\sigma(x, y) = \sigma(x)\mathrm{cer}\langle x, y\rangle = \sigma(y)\mathrm{cov}\langle x, y\rangle.$$

The definitions given above allow us to associate with every flow graph a decision algorithm (decision table).

Let $[x_1, x_n]$ be a path such that x_1 is an input and x_n an output of the graph respectively. Such path will be called *complete*.

The set of all decision rules $x_{i_1}x_{i_2}\ldots x_{i_{n-1}} \to x_{i_n}$ associated with all complete paths $[x_{i_1}, x_{i_n}]$ in G will be called a *decision algorithm* of G.

Example (*cont.*). The decision algorithm, presented in the form of a decision table, associated with flow graph shown in Fig. 5, is given in Table 1.

Remark. The sum of strengths in the decision table is 0.99. This is due the round-off errors.

In the decision table *paint* and *car* are *condition attributes*, *manufactures* is the *decision attribute*, whereas x_i, y_i and z_i are values of these attributes, respectively.

Let us notice that all combinations of condition attributes value are present in the decision table. In general case this is not necessary.

We can also look at the relationship between flow graph and decision algorithms differently. We can be interested in replacing all connections of each complete path by a single decision rule, according to definitions given previously. In this case we obtain a decision algorithm, which is a kind of composition of constituent algorithms. This will be explained in more details in the example which follows.

Table 1
Decision table induced by the flow graph

Rule no.	Paint	Car	Manufacturer	Strength
1	x_1	y_1	z_1	0.06
2	x_1	y_1	z_2	0.04
3	x_1	y_1	z_3	0.02
4	x_2	y_1	z_1	0.21
5	x_2	y_1	z_2	0.12
6	x_2	y_1	z_3	0.08
7	x_3	y_1	z_1	0.03
8	x_3	y_1	z_2	0.02
9	x_3	y_1	z_3	0.01
10	x_1	y_2	z_1	0.05
11	x_1	y_2	z_2	0.04
12	x_1	y_2	z_3	0.04
13	x_2	y_2	z_1	0.08
14	x_2	y_2	z_2	0.06
15	x_2	y_2	z_3	0.06
16	x_3	y_2	z_1	0.03
17	x_3	y_2	z_2	0.02
18	x_3	y_2	z_3	0.02

Example (*cont.*). We can ask what is the paint demand by each manufacturer. To this end we have to replace each complete connection by a single branch as shown in Fig. 6.

It is easily seen from the flow graph how paint supply is distributed among manufacturers and what the demand for each paint by every manufacturer is.

For example, supply of paint x_1 is distributed among manufacturers z_1, z_2 and z_3 in the proportions 46%, 29% and 25%, whereas demand for paints x_1, x_2 and x_3 by manufacturer z_1 is 24%, 65% and 13%, respectively.

The flow graph presented in Fig. 5 can be depicted by two decision tables given in Tables 2 and 3, respectively.

The decision table corresponding to the flow graph shown in Fig. 6 is given in Table 4.

This table can be understood as a result of operation performed on the constituent decision tables Tables 2 and 3.

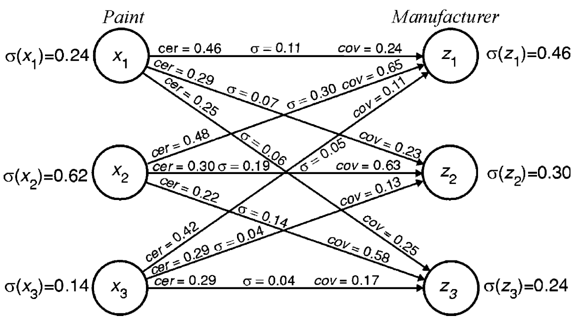

Fig. 6. Paint demand by manufacturers.

Table 2
Paint distribution

Rule no.	Paint	Car	Strength
1	x_1	y_1	0.12
2	x_1	y_2	0.12
3	x_2	y_1	0.42
4	x_2	y_2	0.20
5	x_3	y_1	0.06
6	x_3	y_2	0.08

Table 3
Car manufacturer distribution

Rule no.	Car	Manufacturer	Strength
1	y_1	z_1	0.30
2	y_1	z_2	0.18
3	y_1	z_3	0.12
4	y_2	z_1	0.16
5	y_2	z_2	0.12
6	y_2	z_3	0.12

Table 4
Paint demand by manufacturers

Rule no.	Paint	Manufacturer	Strength
1	x_1	z_1	0.11
2	x_1	z_2	0.07
3	x_1	z_3	0.06
4	x_2	z_1	0.30
5	x_2	z_2	0.19
6	x_2	z_3	0.14
7	x_3	z_1	0.05
8	x_3	z_2	0.04
9	x_3	z_3	0.04

6. Conclusions

We presented in this paper a new approach to flow networks. This approach is basically meant as a new tool for modeling a flow of information represented by a set of decision rules. It may be useful for knowledge-based decision support. It is also shown that the flow in the flow graph is governed by Bayes' formula, however the meaning of the Bayes' formula has entirely deterministic character and does not refer to any probabilistic interpretation. Thus our approach is entirely free from the mystic flavor of Bayesian reasoning raised by many authors, e.g., [1–3]. Besides, it gives clear interpretation of obtained results and simple computational algorithms.

It seems that the presented ideas could be generalized along the lines presented in [6].

Acknowledgements

Thanks are due to Professor Andrzej Skowron and Professor Roman Słowiński for critical remarks.

References

[1] J.M. Bernardo, A.F.M. Smith, in: Bayesian Theory, Wiley Series in Probability and Mathematical Statistics, John Wiley & Sons, Chichester, 1994.

[2] G.E.P. Box, G.C. Tiao, Bayesian Inference in Statistical Analysis, John Wiley and Sons, Inc., New York, 1992.

[3] M. Berthold, D.J. Hand, Intelligent Data Analysis—An Introduction, Springer-Verlag, Berlin, 1999.

[4] L.R. Ford, D.R. Fulkerson, Flows in Networks, Princeton University Press, Princeton, NJ, 1962.

[5] J. Łukasiewicz, Die logishen Grundlagen der Wahrscheinilchkeitsrechnung. Kraków (1913), in: L. Borkowski (Ed.), Jan Łukasiewicz—Selected Works, North Holland Publishing Company & Polish Scientific Publishers, Amsterdam & Warsaw, 1970.

[6] S. Greco, Z. Pawlak, R. Słowiński, 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, vol. 2475, Springer-Verlag, Berlin, 2002, pp. 93–104.

[7] Z. Pawlak, In pursuit of patterns in data reasoning from data—the rough set way, in: J.J. Alpigini, J.F. Peters, A. Skowron, N. Zhong (Eds.), Rough Sets and Current Trends in Computing, Lecture Notes in Artificial Intelligence, vol. 2475, Springer-Verlag, Berlin, 2002, pp. 1–9.

[8] Z. Pawlak, Rough sets, decision algorithms and Bayes' theorem, European Journal of Operational Research 136 (2002) 181–189.

[9] S. Tsumoto, H. Tanaka, Discovery of functional components of proteins based on PRIMEROSE and domain knowledge hierarchy, in: T.Y. Lin, A.M. Wildberger (Eds.), Proceedings of the Workshop on Rough Sets and Soft Computing (RSSC-94), 1994, 1995, pp. 280–285.

[10] S.K.M. Wong, W. Ziarko, Algorithm for inductive learning, Bulletin of the Polish Academy of Sciences 34 (5–6) (1986) 271–276.