URL: http://www.elsevier.nl/locate/entcs/volume82.html 9 pages

Probability, Truth and Flow Graph

Zdzisław Pawlak [1]

University of Information Technology and Management
Newelska 6, 01-447 Warsaw, Poland

Abstract

In 1913 Jan Łukasiewicz proposed to use logic as mathematical foundations of probability. He claims that probability is "purely logical concept" and that his approach frees probability from its obscure philosophical connotation. He recommends to replace the concept of *probability* by the concept of a *truth value*, which can be regarded as a degree of truth, i.e., a number between 0 and 1, of propositional functions (called in his work *indefinite propositions*). Further he shows that all laws of probability can be obtained from a properly built logical calculus.

In this paper we show that the idea of Łukasiewicz can be also expressed differently. Instead of using truth values in place of probability, stipulated by Łukasiewicz, we propose, in this paper, using of deterministic flow analysis in flow networks (graphs). In the proposed setting, flow is governed by some probabilistic rules (e.g., Bayes' rule), or by the corresponding logical rules, proposed by Łukasiewicz, though, the formulas have entirely deterministic meaning, and need neither probabilistic nor logical interpretation. They simply describe flow distribution in flow graphs. However, flow graphs introduced here are different to those proposed by Ford and Fulkerson, for optimal flow analysis, because they model rather flow distribution in a plumbing network, then the optimal flow.

The flow graphs considered in this paper can be also used as a description of a decision algorithms, where branches of the graph are interpreted as decision rules. This feature causes that flow networks can be also used as a new tool for data analysis, and knowledge representation.

1 Introduction

In [3] Jan Łukasiewicz proposed to use logic as mathematical foundations of probability. He claims that probability is "purely logical concept" and that his approach frees probability from its obscure philosophical connotation. He recommends to replace the concept of *probability* by *truth values* of *indefinite propositions*, which are in fact propositional functions.

[1] Email: zpw@ii.pw.edu.pl

Let us explain this idea more closely. Let U be a non empty finite set, and let $\Phi(x)$ be a propositional function. The meaning of $\Phi(x)$ in U, denoted by $|\Phi(x)|$, is the set of all elements of U, that satisfies $\Phi(x)$ in U. The truth value of $\Phi(x)$ is defined as $card|\Phi(x)|/card U$. For example, if $U = \{1, 2, 3, 4, 5, 6\}$ and $\Phi(x)$ is the propositional function $x > 4$, than the truth value of $\Phi(x) = 2/6 = 1/3$. If the truth value of $\Phi(x)$ is 1, then the propositional function is true, and if it is 0, then the function is *false*. Thus the truth value of any propositional function is a number between 0 and 1. Further, it is shown that the truth values can be treated as probability and that all laws of probability can be obtained by means of logical calculus.

In this paper we show that the idea of Łukasiewicz can be also expressed differently. Instead of using truth values in place of probability, stipulated by Łukasiewicz, we propose, in this paper, using of deterministic flow analysis in flow networks (graphs). In the proposed setting, flow is governed by some probabilistic rules (e.g., Bayes' rule), or by the corresponding logical calculus proposed by Łukasiewicz, though, the formulas have entirely deterministic meaning, and need neither probabilistic nor logical interpretation. They simply describe flow distribution in flow graphs. However, flow graphs introduced here are different to those proposed by Ford and Fulkerson for optimal flow analysis, because they model rather, e.g., flow distribution in a plumbing network, than the optimal flow.

The flow graphs considered in this paper are basically meant not to physical media (e.g., water) flow analysis, but to information flow examination in decision algorithms. To this end branches of a flow graph are interpreted as decision rules. With every decision rule (i.e., branch) three coefficients are associated, the *strength, certainty and coverage factors*. In classical decision algorithms language they have probabilistic interpretation. Using Łukasiewicz's approach we can understand them as truth values. However, in the proposed setting they can be interpreted simply as flow distribution ratios between branches of the flow graph, without referring to their probabilistic or logical nature.

This interpretation, in particular, leads to a new look on Bayes' theorem, which in this setting, has entirely deterministic explanation.

The presented idea can be used, among others, as a new tool for data analysis, and knowledge representation.

We start our considerations giving fundamental definitions of a flow graph and related notions. Next, basic properties of flow graphs are defined and investigated. Further, the relationship between flow graphs and decision algorithms is discussed. Finally, a simple tutorial example is used to illustrate the consideration.

2 Flow Graphs

A 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*, $\sigma : \mathcal{B} \to < 0, 1 >$ is a *flow function*.

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

Input and *output* of a graph G, are defined by $I(G) = \{x \in N : I(x) = \emptyset\}$, $O(G) = \{x \in N : O(x) = \emptyset\}$, respectively.

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

With every node x of a flow graph G we associate its *inflow* and *outflow* defined as $\sigma_+(x) = \sum_{y \in I(x)} \sigma(y, x)$, $\sigma_-(x) = \sum_{i \in O(x)} \sigma(x, y)$, respectively. An inflow and an outflow of G are defined by $\sigma_+(G) = \sum_{x \in I(G)} \sigma_-(x)$, $\sigma_-(G) = \sum_{x \in O(G)} \sigma_+(x)$, respectively.

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

Obviously $\sigma_+(G) = \sigma_-(G) = \sigma(G)$, where $\sigma(G)$ is a *troughflow* of G. Moreover, we assume that $\sigma(G) = 1$.

The above formulas can be considered as *flow conservation equations* [2].

3 Properties of Flow Graphs

With every branch of a flow graph we associate the *certainty* and the *coverage factors.*

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

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

respectively, where $\sigma(x)$ is the *normalized troughflow* of x, defined by $\sigma(x) = \sum_{y \in O(x)} \sigma(x, y) = \sum_{y \in I(x)} \sigma(y, x)$. Immediate consequences of definitions given above are:

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

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

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

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

Obviously the above properties have a probabilistic character, e.g., equations (3) and (4) can be interpreted as Bayes' formulas. However, these properties can be interpreted in deterministic way and they describe flow distribution among branches in the network.

Notice that Bayes' formulas given above have a new mathematical form. Bayes' theorem is expressed by means of strength of decision rules, which simplifies essentially computations.

4 Paths and Connections

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

$$cer[x_1 \ldots x_n] = \prod_{i=1}^{n-1} cer(x_i, x_{i+1}), \tag{6}$$

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

$$cov[x_1 \ldots x_n] = \prod_{i=1}^{n-1} cov(x_i, x_{i+1}), \tag{7}$$

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

$$\sigma[x \ldots y] = \sigma(x) cer[x \ldots y] = \sigma(y) cov[x \ldots y]. \tag{8}$$

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

The *certainty* of connections $< x, y >$ is

$$cer < x, y >= \sum_{[x \ldots y] \in < x, y >} cer[x \ldots y], \tag{9}$$

the *coverage* of connections is $< x, y >$

$$cov < x, y >= \sum_{[x \ldots y] \in < x, y >} cov[x \ldots y], \tag{10}$$

and the *strength* of connections is $< x, y >$

$$\sigma < x, y >= \sum_{[x \ldots y] \in < x, y >} \sigma[x \ldots y]. \tag{11}$$

Let x, y $(x \neq y)$ be nodes of G. If we substitute the sub-graph $< x, y >$ by a single branch (x, y) such that $\sigma(x, y) = \sigma < x, y >$ then $cer(x, y) = cer < x, y >$, $cov(x, y) = cov < x, y >$ and $\sigma(G) = \sigma(G')$, where G' is the graph obtained from G by substituting in G (x, y) instead of the subgraph $< x, y >$.

5 Decision Algorithms

With every branch (x, y) we associate a decision rule $x \to y$, read if x *then* y; x will be referred to as a *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

(12) $$cer(x^*, x_n) = cer[x_1 \ldots x_n],$$

(13) $$cov(x^*, x_n) = cov[x_1 \ldots x_n],$$

and

(14) $$\sigma(x^*, x_n) = \sigma(x_1) cer[x_1 \ldots x_n] = \sigma(x_n) cov[x_1 \ldots x_n].$$

Similarly, every connection $< x, y >$ can be interpreted as a single *decision rule* $x \to y$ such that:

(15) $$cer(x, y) = cer < x, y >,$$

(16) $$cov(x, y) = cov < x, y >,$$

and

(17) $$\sigma(x, y) = \sigma(x) cer < x, y >= \sigma(y) cov < x, y > .$$

Let $[x_1 \ldots x_n]$ be a path such that x_1 is an input and x_n an output of the flow graph G, respectively. Such a path and the corresponding connection $< x_1, x_n >$ will be called *complete*.

The set of all decision rules $x_{i_1} x_{i_1} \ldots x_{i_{n-1}} \to x_{i_n}$ associated with all complete paths $x_{i_1} \ldots x_{i_n}$ will be called a *decision algorithm* induced by the flow graph.

The set of all decision rules $x_{i_1} \to x_{i_n}$ associated with all complete connections $< x_{i_1}, x_{i_n} >$ in the flow graph, will be referred to as the *combined decision algorithm* determined by the flow graph.

6 Inference in Flow Graphs

Reasoning in deductive logic consists in using *inference rules*, which are implications in the form, *if* Φ *then* Ψ, where Φ is called the *premises* (*reason*) and Ψ – the *consequence* of the rule. Inference rules allow us to obtain true consequences from true premises. Fundamental rules of inference are *modus ponens* and *modus tollens*.

Modus ponens has the following form:

$$
\begin{array}{lcl}
\textit{if} & \Phi \rightarrow \Psi & \textit{is true} \\
\textit{and} & \Phi & \textit{is true} \\
\hline
\textit{then} & \Psi & \textit{is true}
\end{array}
$$

and *modus tollens* is as follows:

$$
\begin{array}{lcl}
\textit{if} & \Phi \rightarrow \Psi & \textit{is true} \\
\textit{and} & \sim \Psi & \textit{is true} \\
\hline
\textit{then} & \sim \Phi & \textit{is true}
\end{array}
$$

Modus tollens can be regarded as the inverse of *modus ponens*, i.e., gives reason for a consequence.

In reasoning about data (data analysis) the situation is slightly different. Instead of true sentences we consider propositional functions, which are true to a “degree”, i.e., they assume truth values which lie between 0 and 1, in other words, they are probable, not true. In the flow graph setting the concepts of truth (or probability) is replaced by the flow intensity in branches of the flow graph, and logical inference is boiled down to flow distribution analysis. Thus a flow graph can be regarded as schema of reasoning about data patterns – i.e., a network of decision rules, which lead from propositional functions expressing properties of initial data to other propositional functions about data.

This idea can be formulated more exactly as follows:

If $< x, y >$ is a connection in G, then

$$\sigma(y) = \frac{\sigma(x) cer < x, y >}{cov < x, y >} = \frac{\sigma(x, y)}{cov < x, y >}, \tag{18}$$

and

$$\sigma(x) = \frac{\sigma(y) cov < x, y >}{cer < x, y >} = \frac{\sigma(x, y)}{cer < x, y >}. \tag{19}$$

Formulas (17) and (18) are direct consequences of (3) and (4), respectively – consequently they are Bayes’ rules. Obviously, they play similar rule in data analysis to that played by *modus ponens* and *modus tollens* in logical reasoning.

Let us stress once more that formulas (17) and (18) can be interpreted in probabilistic or logical terms, however in our setting they simply describe deterministic flow distribution in flow graphs.

7 An Example

Now we will illustrate ideas introduced in the previous sections by means of a simple example concerning votes distribution of various age groups and social classes of voters between political parties.

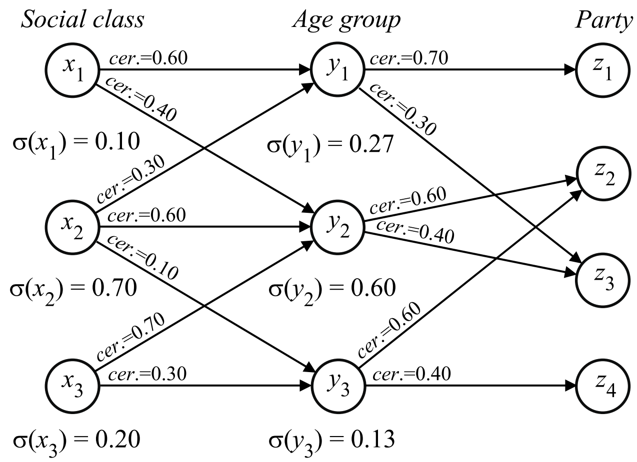

Fig. 1. Social class and Age group votes distribution

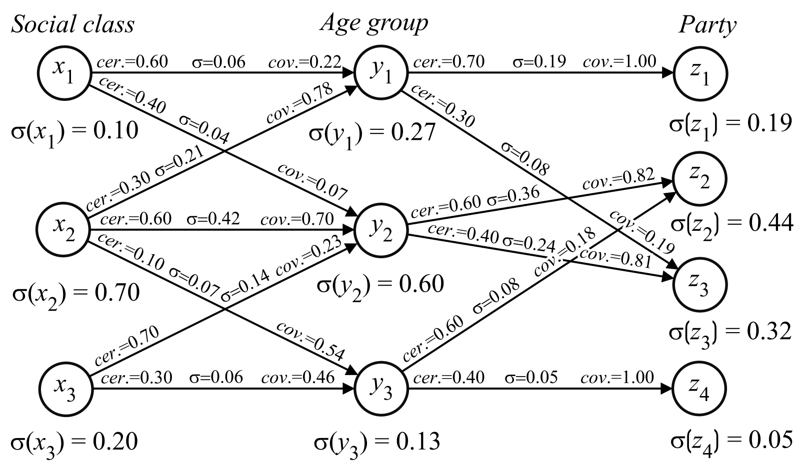

Fig. 2. Party votes distribution

Consider three disjoint age groups of voters y_1 (*old*), y_2 (*middle aged*) and y_3 (*young*) – belonging to three social classes x_1 (*high*), x_2 (*middle*) and x_3 (*low*). The voters voted for four political parties z_1 (*Conservatives*), z_2 (*Labour*), z_3 (*Liberal Democrats*) and z_4 (*others*).

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

First we want to find votes distribution with respect to age group. The result is shown in Fig. 2.

From the flow graph presented in Fig. 2 we can see that, e.g., party z_1 obtained 19% of total votes, all of them from age group y_1; party z_2 – 44% votes, which 82% are from age group y_2 and 18% – from age group y_3, etc.

If we want to know how votes are distributed between parties with respects to social classes we have to eliminate age groups from the flow graph. Employing the algorithm presented in section 5 we get results shown in Fig. 3.

From the flow graph presented in Fig. 3 we can see that party z_1 obtained 22% votes from social class x_1 and 78% – from social class x_2, etc.

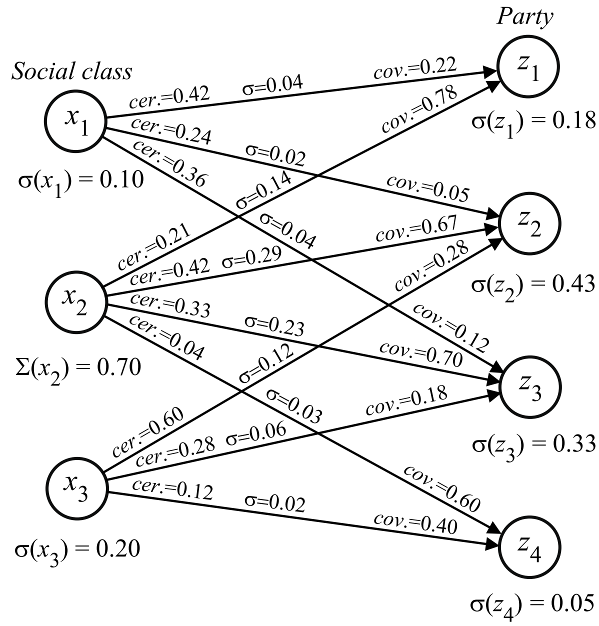

Fig. 3. Social class and Party relationship

Remark. Due to the round-off errors formulas (1) – (16) may not be always satisfied.

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

If Party (z_1) *then Age group* (y_1) (0.19)

If Party (z_2) *then Age group* (y_2) (0.36)

If Party (z_2) *then Age group* (y_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 (z_1) *then Social class* (x_1) (0.04)

If Party (z_1) *then Social class* (x_2) (0.14), etc

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

If Social class (x_1) *then Party* (z_1) (0.04)

If Social class (x_1) *then Party* (z_2) (0.02)

If Social class (x_1) *then Party* (z_3) (0.04), etc

From the examples given above one can easily see the relationship between the role of *modus ponens* and *modus tollens* in logical reasoning and using flow graphs in reasoning about data.

8 Conclusions

In this paper we have shown a new mathematical model of a flow networks, which can be used to decision algorithm analysis. In particular it has been revealed a new interpretation on Bayes' theorem, where the theorem has entirely deterministic meaning, and can be used to decision algorithm study. In this paper we have shown a new mathematical model of a flow networks, which can be used to decision algorithm analysis. In particular it has been revealed a new interpretation on Bayes' theorem, where the theorem has entirely deterministic meaning, and can be used to decision algorithm study.

Acknowledgment

Thanks are due for Professor Andrzej Skowron for critical remarks.

References

[1] Box, G.E.P., Tiao, G.C.: Bayesian inference in statistical analysis. John Wiley and Sons, Inc., New York, Chichester, Brisbane, Toronto, Singapore (1992)

[2] Ford, L.R., Fulkerson, D.R.: Flows in Network. Princeton University Press, Princeton. New Jersey (1962)

[3] Łukasiewicz, J.: Die logishen Grundlagen der Wahrscheinilchkeitsrechnung. Krakow (1913), in: L. Borkowski (ed.), Jan Łukasiewicz – Selected Works, North Holland Publishing Company, Amsterdam, London, Polish Scientific Publishers, Warsaw (1970) 16-63

[4] Pawlak, Z.: In pursuit of patterns in data reasoning from data – The rough set way, Lecture Notes in Artificial Intelligence **2475** Springer-Verlag, Heidelberg (2002) 1-9

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

[6] Pawlak, Z.: Decision rules and flow networks (to appear)