ARGUMENTA OECONOMICA

1 • 1995

Academy of Economics in Wrocław
Wrocław 1995

TABLE OF CONTENTS

ARGUMENTA OECONOMICA
No 1 • 1995
PL ISSN 1233-5835

Andrzej J. Baborski

INDUCTIONAL METHODS OF KNOWLEDGE DISCOVERY IN SYSTEMS OF ARTIFICIAL INTELLIGENCE

1. PRINCIPAL CONCEPTS

Modelling is one of the principal domains of scientific activity even when it is not explicitly mentioned. The notions of mathematical models, database models, and other applications of modelling are mentioned in extensive literature (among others Baborski 1980 and McFadden, Hoffer 1988). When dealing with the problems of artificial intelligence (AI) we must take this literature into consideration, as AI makes use of earlier research achievements. Let us begin with the formulation of the notion of a model. It is formulated in a different way. For our purpose, I think the following definition will fit the best:

Model is the description of some fragment of reality, expressed in specific language.

The fragment of reality mentioned above, must be determined by the user (researcher) to suit the specific problem. In view of the above definition, knowledge is a model. This description serves simultaneously as a base for explaining phenomena occurring in this fragment of reality.

When we evaluate a model, it is very important to determine the range in which it can be applied. A model can be applicable in different ranges.

Range of applicability of a model is such a fragment of reality, where description retains relevance.

The bigger the range of applicability the more general is a model, i.e. it can be applied to broader scope of events. Since the essence of science is the description and explanation of nature, then models are at its roots.

There are two ways of model building: inductive and deductive. Deduction is drawing conclusions based on some assumptions deemed generally true (axioms). It is therefore mandatory to have prior knowledge to the inference process. If this knowledge proves to be general enough, then what is true in the whole range is true in every instance. Deduction lies at the foundation of the vast majority of presently operating systems with knowledge base. The knowledge base is obtained most frequently from experts and every specific system is destined to operate in the range of applicability of this base.

In this article though we take another approach. Very often a situation is encountered when we do not have an expert to consult in the domain in question. There is though usable database containing documentation of the real life process. It is apparent that every database record is the documentation of a fact. Therefore it carries implicit knowledge on the process and it is important to extract this knowledge into explicit form. Such an approach is called knowledge discovery.

What is discovery? There are two approaches to this notion. Mathematicians treat discovery as **creation** of a new noncontroversial system of axioms and conclusions, being (as Lebesgue has said) designate of some reality. Physicists and people doing research in applied science define discovery in other way.

> **Discovery is new description** of known facts, which explain them better and more generally than hitherto known descriptions.

Let us recall that Columbus discovered America but was not aware of that. He was, to the end of his days, convinced that he had discovered a new way to India and the people who originally inhabited this land were (and still are) called Indians. Only when controversies over the facts became apparent, did America become America. Normal approach in science is that initially the facts are documented and only later descriptions (models) to fit these facts are created. Such was the situation when a new disease – AIDS – appeared. *Hypotheses non possumus fingere.*

The problem of discovery can be formulated in the following way with given:

- observation results (facts), situation descriptions and the like, which will be denoted as F;
- other implicit assumptions resulting from general state of knowledge etc. (see Michalski 1990);

inductive assertion (hypothesis) $H : H \Rightarrow F$ must be formulated. This implication can be strong – when the sentence $H \Rightarrow F$ is always true – or weak, when there exist only beliefs, that facts result from hypothesis. Because hypothesis is obtained from assertion, it is not reasoning preserving truth like *modus ponens*, but is retaining false. From that, indirectly, results the necessity of checking machine-made discoveries by an expert.

A fact (model) can be presented as the conjunction of features together with their values. Formally, a fact can be expressed as

$$F \equiv (X_1 = a) \cap (X_2 = b) \cap \ldots (X_n = h).$$

For a set of facts

$$\mathcal{F} = \{F_i : i = 1, 2 \ldots k\},$$

answers for three following questions must be found:

- whether all facts are characterized by the same features $X_1, \ldots X_n$;
- whether all features are characterized by the same domains;
- whether in different facts are features with the same values.

If answers for the two first questions are positive, then a general model is the simple generalization

$$\mathcal{F}^* \subseteq X_1 \times X_2 \times \ldots X_k,$$

and subsets' delimitation is determined by sets of values related to specific features. A model or fact can be named, similar labeling can be done in case of aggregated description. For instance:

Good credit taker ≡ (Liquidity = big) ∩ (Business plan = good)
Poor credit taker ≡ (Liquidity = small) ∩ (Business plan = good)
Bad credit taker ≡ (Liquidity = small) ∩ (Business plan = poor)
Credit taker ≡ Good credit taker ∪ Poor credit taker ∪ Bad credit taker

Let us note that two descriptions of facts are not only characterized by the same features but values of the features are identical. In such a case giving features with identical values is superfluous and the only discriminating factor between good and poor credit taker is liquidity. Identical values are rather rare, more common are near values of a feature.

On the basis of the above assumptions, the following rules of generalization can be stated:

1. If some features and values are identical, then they carry no information and their mentioning in the model is not necessary.

2. If the features are the same but have different values, they should be ranked according to some utility criterion. Such criterion can be the frequency of a value, variability measure, gains from information or belief on utility (subjective utility according to Dempster-Shafer theory).

3. If the features are different, then the description should be formulated on the base of common features and then its relevance tested. If the description obtained in this way is not relevant enough, then it is advised to resign from a global description in favour of several descriptions for subsets or dendrite structure, with root for common features and branches for more specific features. The scheme of this method is presented in Fig. 1.

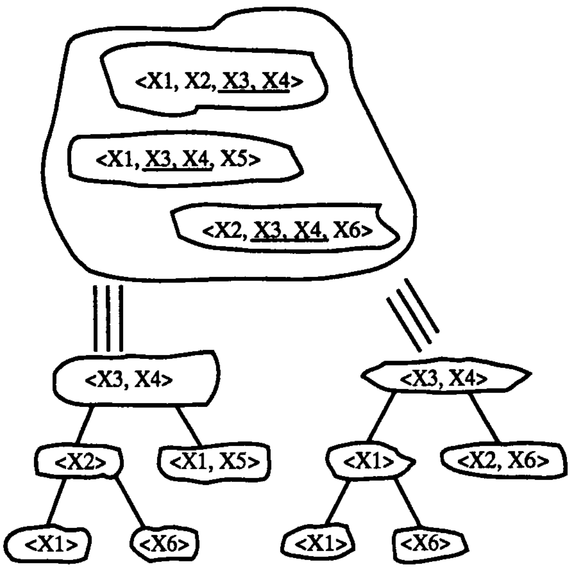

Fig. 1

As a result of analysis, rules are obtained. Here I must point to one frequently neglected problem. It is intuitive to generate, on the basis of the above example, the following rule:

if (Liquidity = big), (Business plan = good) **then** Good credit taker

so that fulfilment of assumptions implies the conclusion. If the assumptions are not fulfilled, then a conclusion cannot be drawn. It is a simplification, because in a real life situation an implication can be fulfilled also with false assumptions (there can be good credit takers with small liquidity). The possibility of fulfilment of implication at false assumptions calls for some caution in creating rules. Like in statistics, **two** kinds of errors should be taken into account:

type I – conclusion true with false assumptions,

type II – conclusion false with true assumptions.

In typical shells like Guru or Kappa only the complement of error type II is given as chances. Considering both errors could enable us to construct expert systems with more reliable inference.

In analysing data we apply known methods of statistics and mathematics. The process of analysis can be greatly facilitated by the use of a computer. On the other hand, a computer cannot do all the job, because it lacks understanding of the problem. In this publication I would like to propose computer aided method of knowledge discovery. It is based on the analysis of modal values. There are two reasons for such an approach:

– mode and approximate values group the most frequent occurrencies of data in a set. Therefore in analysis of mode we have to deal with larger samples than in analysis of any other value;

– mode is very easy to compute and therefore the procedures are fast and simple;

– mode is not influenced by other values in a set, therefore this statistic is very robust.

2. INDUCTIVE KNOWLEDGE ACQUISITION

With inductive knowledge acquisition we have to deal with every case, where we have a description of facts and on this basis we want to acquire knowledge. In my opinion it is the basis of all applied science. In all these sciences we gather facts and then a theory (model) which will explain further facts of that type, already in the way of deductive reasoning. Here lots of methods exist, among which worth mentioning are

– estimation of functional models,

– determination of correlational dependencies,

– aggregation of data into taxonomical types,

– generation of rules

and others which I will not mention. These models are extensively described in literature (e. g. *Logika...* 1987 and Pawłowski 1967). Inductive knowledge acquisition is made in two principal ways:

– supervised (or aided) experiment,

– unsupervised experiment.

Supervised acquisition can be made with the use of training or directly on the data. In any case a knowledge engineer and an expert are involved. The most popular presently is Quinlan's ID3 algorithm, where hypotheses are presented as decision trees. Literature on algorithm ID3 is very rich (Bundy 1990). Inductive methods also comprise reasoning by analogy, i. e. case based reasoning (CBR). In these methods possibility of extrapolation of the model obtained on one set of data to other sets. Like all algorithms based on assertion, these algorithms require a good degree of time invariance in the problem modeled, otherwise reliability can be poor as in the case of stock exchange forecasts or weather forecasts.

Unsupervised experiment is based on direct (real or test) data analysis. Such analysis leads to the generation of rules and it is better to treat it in categories of knowledge discovery. In this paper we propose a method of knowledge discovery based on analysis of modes. The method comprises of two types of work:

– generation of rules,

– knowledge modification.

3. MODE AND ITS DETERMINATION

Mode is determined on the basis of analysis of data occurrencies in a file. There can be proposed two ways of such analysis (for brevity let us denote that we analyze the variable X):

i. The file is sorted with respect to the values of X. In Fig. 2 one of the possible graphic presentations of results obtained for a continuous variable is shown. It resembles spectrometer photography and indeed this technique has much in common with the analysis of radiation or absorption spectra. The graph or sorted table is then scanned with 'scanning window' S as shown in Fig. 2 as analogous to densitometer probe. If the width of the scanner is h then an estimate of frequency $f(x_0)$ is the number of occurrencies fulfilling the criterion $x_0 \le x \le x_0 + h$.

Fig. 2

ii. It is assumed that the distribution of values falls into n classes and the set of classes is prepared before the start of the analysis. Then the data file is read and data are counted in appropriate classes according to a simple classification algorithm as e. g. in (Wirth 1980). The outcome of such classification is shown in Fig. 3.

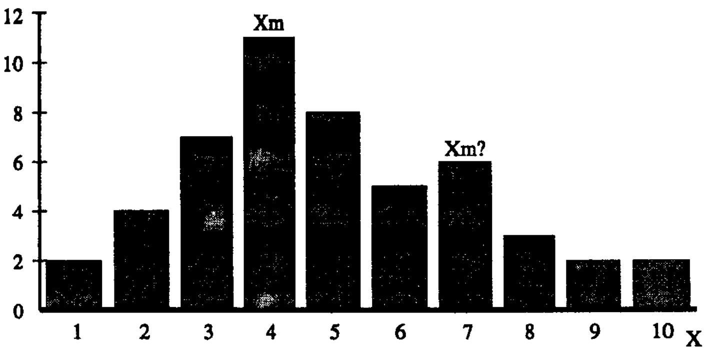

Fig. 3

The problem with this method is that frequently mode, especially in case of continuous X, is near the border of two classes and there is skewness in classes.

Special measures must be taken to avoid skewness, by means of dividing distribution into uneven classes with reduced skewness.

The situation as presented in Fig. 3 in case of $X = 4$ is apparent. Frequently though there is a question about existence of a mode (as in the case of $X = 7$). To solve the problem we must test hypothesis

$$H_0: P(X = x_m) = P(X \neg = x_m)$$

and its rejection proves that there is a mode. In case of independent events the distribution of occurrencies in class interval is binomial with $p = \frac{1}{n}$ where n is the number of classes. Problems emerging in case of correlated events are now being studied.

In case of multimodal distribution the following algorithm of mode determination can be proposed.

1. Determine the modal value x_m.
2. Extract from distribution all values of $X = x_m$.
3. If nonparametric test for uniformity of distribution cannot be rejected, then **stop**, else **go to 1.**

This algorithm gives a sequence of modal values, sorted according to descending number of occurrencies. Each mode is tested for relevance.

4. GENERATION OF RULES

Having determined modes of variable X, we can investigate their relations to other variables. Relation can be easily shown on conditional distributions. Let us note, that if such a relation exists between X and Y it can have the form of an implication

$$X = x_m \Rightarrow Y = y_m,$$

in case of unimodal distributions. Such a situation does not often occur and can lead to trivial conclusions. More often is the situation described by the implication

$$X = x_m \Rightarrow Y = y_{m1} \cup Y = y_{m2} \cup \dots Y = y_{mk}$$

applicable in case of multimodal conditional distribution. In this case one cause can lead to different conclusions, as presented in Fig. 4.

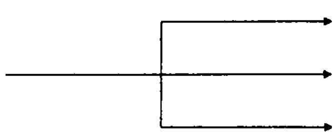

Fig. 4

The third case of implication has the form

$$X = x_{m1} \cup X = x_{m1} \cup \ldots \Rightarrow Y = y_m.$$

It describes the situation with many causes of one result, shown in Fig. 5. Existence of this dependence makes abduction an uncertain tool of inference. The most general form is one step Markov process with transitions from i-th state, $1 \le i \le k$ of variable X to j-th state $1 \le j \le m$ of variable Y. The probabilities of these transitions form $k \times m$ matrix. This topic goes beyond this publication.

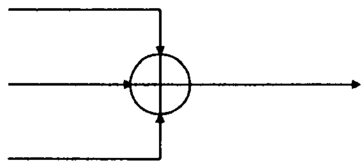

Fig. 5

Probability of an implication can be calculated from the formula

$$P\,(X = x_m \Rightarrow Y = y_m) = P\,(X = x_m) \cdot P\,(Y = y_m \mid X = x_m). \qquad (1)$$

Properly speaking it is the probability $P\,(X = x_m \cap Y = y_m)$ but in artificial intelligence one is not interested in void implications.

Rules in knowledge base have the form

if A **then** B **chance** C (2)

where A is assumption, B conclusion and C is given by (1). It is impossible to know, by merely reading the data, which of them can be associated to assumption and which to conclusion; it calls for knowledge, which can be supplied only by an expert. But once it is supplied, the process of rules generation is straightforward. It is based on the modes of a variable assumed to represent the cause. For the generation of rules we can propose the following algorithm:

1. Determine the variable X being characteristic of cause and variable(s) Y being characteristic of result(s);

2. For X find relevant modal values x_{mi}, $i = 1, \ldots k$; relevance is tested by nonparametric hypotheses described above;

2. For each x_{mi} determine the subset of database $D = \{d : X = x_{mi}\}$. Count the number of occurrencies n_{xi};

4. In the subset D determine relevant modal values y_{mj}, $j = 1, \ldots, m$ of the variable characterizing the result. Count the number of occurrencies n_{yj};

5. Calculate probability estimate for implication

$$P\,(X = x_m \Rightarrow Y = y_m) = \frac{n_{xi}}{n} \cdot \frac{n_{yj}}{n_{xi}} = \frac{n_{yj}}{n};$$

6. Generate inference rules in the form (2).

It can be questioned whether the results thus obtained is knowledge as defined in knowledge bases. In AI it is assumed that knowledge is composed of data and the rules of inference on the basis of these data and that knowledge base enables not just inference but also explanation of the conclusion obtained. Also knowledge base should give the possibility of explanation to questions posed to the user. But since the rules describe causal regularities existing in the real process, then they can be treated as knowledge about the process. Explanations must be supplied by experts since they cannot be gathered from merely reading the file.

Seeking modal values can be started with causes or consequences. Above we started with causes as naturally grounded. The reverse seeking is equally justified and preferable if the situation depicted in Fig. 4 is suspected. In this case, consequences are represented with bigger samples and therefore more relevant.

5. KNOWLEDGE MODIFICATION

Normally, as described in literature, the construction of an expert system is finished when the knowledge base is created. We must bear in mind that an expert's knowledge, unless it pertains to the fundamental laws of nature, is valid only in some time span and must be submitted to verification. Verification can be done in the same way as knowledge discovery. The problem of verification can be formulated as follows: let us be given some set of rules

if A_1 then B_1 with p_1,

if A_2 then B_2 with p_2,

....................,

if A_N then B_N with p_N.

In order to verify, it is necessary to generate rules anew, this time though not seeking modal values but assuming them as a set. In this situation generation algorithm begins with Step 3 and ends with Step 5. As a result of execution of this algorithm estimates $p_1', p_2', ..., p_N'$ of respective rules are obtained. Presently for every $1 \leq i \leq N$, a hypothesis:

$$H_{0i}' : p_i = p_i'$$

must be tested. If there is no reason to reject this hypothesis, then pass to the next i. If the hypothesis would be rejected, a hypothesis

$$H_i'' : p_i' = 0$$

is to be tried. If this hypothesis cannot be rejected, the rule must be removed from the base. In the other case p_i is substituted by p_i'. If, during modification of the knowledge base some rules have been removed, suspicion arises that some new ones could appear. In such a case it is necessary to perform Step 2 of the algorithm in order to check whether new modal values exist. In this way the limited time of validity of knowledge is taken into account.

6. PROGRAMMED INDUCTION AND NEURAL NETWORKS

Neural networks, being the hardware or software realization of parallel computers, are the alternative with respect to algorithms written in programming languages for von Neumann type machines. Presently the majority of neural networks are simulators of parallel computers, because of the big cost of multiprocessor machines. There is ample literature on the relative advantages and disadvantages of both approaches (e. g. Lawrence 1993; Mooney et al. 1990). Naturally comparison is possible only when both types can be applied for the solution of the same problems. The basis for comparison is in the first instance the cost of operation as a common measure of both types of systems.

The cost of both, an expert system and neural network, is composed of many components. Their full list is long and frequently difficult to determine. The most important ones, with respect to which consensus was reached are:

- cost of expert and knowledge engineer,
- cost of system training,
- cost (time) of problem solutions,
- cost of hardware and software,
- gains (losses) on expertise obtained from the system.

Not all costs can be precisely determined, especially a priori. Cost of an expert is, to a large extent, the result of negotiations. Cost of system learning is mainly time spent on learning (multiplied by cost of unit of time) and time spent on training data preparation. Time of training exerts also an influence on costs related to expert. Only the last three components are fully comparable, because they pertain to the same magnitudes and are easy to measure. That is the reason why in publications on comparisons of expert systems and neural networks authors usually avoid categorical, numeric evaluation, in favour of averages from many samples and general indices.

As can be seen from the data in Table 1, the amount of common applications of both systems is not large. Comparison is possible only in the applications where 'competition' is possible. Coming to the last component of cost evaluation most frequently we have in mind the percentage of right answers as the

Table 1
Comparison of applications of neural networks and expert systems

Application	Neural networks	Expert systems
Economics		
Cost evaluation	+	+
Financial trends forecasting	+	+
Analysis of credit applications	+	+
Real estate price evaluation	+	
Urban planning		+
Management and law		
Intelligent documents retrieval		+
Planning	+	+
Investment strategy		+
Identification of fingerprints	+	
Determination of legal status	+	+
Information security		+
Engineering		
Chemistry		
Forecasting chemical reactions	+	
Identification of carbohydrates	+	+
Industrial applications		
Engine diagnoses	+	+
Hardware configuration		+
Repair instructions		+
Modelling of control processes	+	+
Quality control	+	+
Medicine and biology		
Bacteria identification	+	+
Identification of cancerous cells	+	
Medical diagnosis	+	+
Medical verdicts		+
Other		
Identification of signals and targets	+	
Education – scoring		+
Contextual identification of words	+	
Natural language processing		+
Translation of speech into text	+	
Sport results forecasting	+	+
Manuscripts identification	+	
Graphical characters identification	+	

Source: developed on the basis of (Lawrence 1993).

principal characteristic of system in the sense of its reliability. Also good measure is comparison of previous costs with cost of a human expert.

From Table 1 we can see that both these approaches are different solutions fit for different applications. Expert systems are better justified in well-structured problems, whose solution involves multistep logical reasoning and dialog with the user. As an example we can quote an expert system for medical diagnoses. Neural networks on the other hand are better fit for the problems analysed with use of taxonomic analysis and one step classification or pattern recognition.

Neural network have one thing in common with the above described method for generation of rules. Their very nature is based on **typical** i.e. frequent occurrencies. Training of neural networks involves feeding them with large amounts of data in which, by proper choice of weights, typical situations are determined. Therefore analysis of modal values is being implicitly applied also in neural networks to generate plausible solutions. Therefore they can be used for knowledge discovery and that knowledge can be later transformed into rules.

7. APPLICATIONS

A characteristic feature of the proposed method of knowledge discovery is the 'purification' of information. Presently we frequently encounter situation with not lack but abundance of information. In such situations the method proposed can find the best application. Also relevant are situations where, as mentioned on page 22, there is no expert to gain knowledge from. Such is for instance the situation in decision making about granting a bank loan. Commercial activity of banks in Poland began five years ago and the majority of enterprises have even shorter history. Therefore the majority of enterprises do not have a credit history which could be studied when credit standing is validated.

It can seem that the activity of a credit officer is an art not subject to structuring. Such was the opinion of some managers of American banks I met with. In these banks future credit officers are trained for five years before being admitted to credit decisions. But, if after such training they are able to make similar decisions for similar cases, then it means they gain knowledge making possible to draw controllable conclusions.

REFERENCES

Baborski A. (1980): *Theory of Formal Languages in Modelling of the Dynamic Systems.* Published by Academy of Economics, Wrocław.

Bundy A., ed. (1990): *Catalogue of Artificial Intelligence Techniques.* Springer-Verlag.

Dietterich T. G. (1990): *Learning at the Knowledge Level,* [in:] *Readings in Machine Learning,* ed. Shawlik J. W. Morgan Kaufmann Publishers Inc.

Kibler D., Langley P. (1990): *Machine Learning as an Experimental Science,* [in:] *Readings in Machine Learning,* ed. Shawlik J. W. Morgan Kaufmann Publishers Inc.

Lawrence J. (1993): *Introduction to Neural Networks.* California Scientific Software Press.

Logika formalna. Zarys encyklopedyczny [*Formal Logic Ecyclopaedic Outline*]. (1987). PWN, Warsaw.

Martin J. (1988): *Data Types and Data Structures*. Prentice-Hall.

McFadden J., Hoffer J. A. (1988): *Data Base Management*. Benjamin/Cumings.

Michalski R. S. (1990): *A Theory and Methodology of Inductive Learning*, [in:] *Readings in Machine Learning*, ed. Shawlik J. W. Morgan Kaufmann Publishers Inc.

Mooney R., Shawlik J., Towell G., Grove A. (1990): *An Experimental Comparison of Symbolic and Connectionist Learning Algorithms*, [in:] *Readings in Machine Learning*, ed. Shawlik J. W. Morgan Kaufmann Publishers Inc.

Pawłowski Z. (1967): *Ekonometria* [*Econometrics*]. PWN, Warsaw.

Smith J. C., Gelbart D., Graham D. (1992): *Building Expert Systems in Case-Based Law*. 'Expert Systems with Applications' Vol. 4 No 4.

Wirth N. (1980): *Algorytmy + struktury danych = programy* [*Algorithms + data structures = programmes*]. WNT, Warsaw.