APPLICATIONS OF NONCLASSICAL LOGIC METHODS FOR PURPOSES OF KNOWLEDGE DISCOVERY AND DATA MINING[1]

Vladimir Jotsov, Vassil Sgurev, and Adil Timofeev

Abstract: Methods for solution of a large class of problems on the base of nonclassical, multiple-valued, and probabilistic logics have been discussed. A theory of knowledge about changing knowledge, of defeasible inference, and network approach to an analogous derivation have been suggested. A method for regularity search, logic-axiomatic and logic-probabilistic methods for learning of terms and pattern recognition in the case of multiple-valued logic have been described and generalized. Defeasible analogical inference and new forms of inference using exclusions are considered. The methods are applicable in a broad range of intelligent systems.

Introduction

The classical binary logic is related to formalizing strictly correct (formal) arguments. Still the object field that is the background for the basic concepts and conclusions possesses an incomplete, inaccurate, contradictory, and frequently variable information [1-7]. So there is a necessity to use and develop new non-classical methods for formalizing intelligent processes and information technologies.

At present we have a mighty big variety of different non-classical logics [2,3,7]. Yet the methods for application of these logics in tangible problems are poorly developed. Besides the potential of these logics (e.g. the K-valued logics) does not perfectly satisfy the necessities that originate during the elaboration of intelligent systems and technologies.

The statistical approach to data analysis and making optimal decisions remains popular at present. However it requires a representativeness of the output data, and is not functioning in knowledge-poor environments. Practically the training data sets from which the knowledge is found and the intelligent decisions are formulated are very limited and therefore they are not statistically representative.

This paper describes methods of application for multiple-valued and probabilistic logics to solutions of intelligent systems' problems (particularly, to problems of machine learning and search of regularities on an example of three-valued logics). Some approaches to the creation of conclusions are used: inference by analogy, logic-axiomatic and logic-probabilistic methods, and modeling of network flows. It has been shown that the application of non-classic logic tools allows a significant widening both of the application area and also of the theoretical basis for development even in such a developed area as inference using exceptions – the notion defeasible inference is used below.

The suggested methods allow the cooperation between logic and probabilistic approaches and also to obtain preferences from each of them.

1. Basic Characteristics of Defeasible Inference

Let the unity of classes V is comprised by the subsets S_1, S_2, ... and $S \in V$. Every subset of type S includes elements $x_{s:1}$;, $x_{s;2}$, ..., that form a new model. The original set S is related to one of the classes $S_i \in V$. The final result from the analysis S is idenitified with one of the classes S_i in U. The output is an answer of the type V_s = (T; F;?) with three values: "true", "false" and "uncertainty". In the case with an answer V_s =? or V_s=F the set S may be identified with more than a single known class S_{i1}, S_{i2}, ... ($i1 \neq i2$ ...). The answer V_s=T is received if and only if the examined class S coincides with S_i.

Amongst the classes S_i there exists an interdependence of the type "ancestor - successor", (e.g. S_i – an ancestor of S_{i1}). Thus it is possible to form simple types of semantic nets – with one type of relation. It is necessary to note that the elements $x_{si1;1}$, $x_{si1;2}$, ... produce the differences between the class S_{i1} and the other successors of the

[1] The work is partially supported by Grant no. NIP917 of the Ministry of Science and Education – Republic of Bulgaria.

common ancestor S_i. All differences that appear in the comparison process of S_{i1} with other classes that are not direct successors of S_i are determined after the application of the heredity mechanism.

The conclusion (response) V_s=T is formed when for all ancestors S_i and also for S_{i1} the corresponding conjunction terms are of the following form: $A_1' \Lambda A_2' \Lambda..A_n'$, where A_k is x_k or $\neg$ x_k; A_k' may coincide with A_k or it may include (using a disjunction) A_k and analogical terms for other variables.

Let rules of a Horn type describe some domain:

$$B \leftarrow \underset{i \in I}{\Lambda} A_i . \tag{1}$$

During the usage of a binary logic in the referred rules if at least a single variable A_i is not "true" then the truth of B is indefinite i.e. B may mean "true" or "false". In the case when the corresponding exclusion from the conjunction (1) of the rule is based on the inclusion of a term with any A_k ($k \in I$) then the inference procedure changes. In the case if the exclusion E (C, A_k) and C is true and A_k is false then the right side of rule B may be true (as an exception).

The extended inference models with exclusions were introduced and generalized in formalized ones in [9,10] in the following form.

$$\frac{B \leftarrow \overset{z}{\underset{i=1}{\Lambda}} A_i, C, E(C, A_k), \neg A_k \leftarrow C}{B \leftarrow A_1 \wedge A_2 \wedge \ldots A_{k-1} \wedge \neg A_k \wedge \ldots A_z}, \tag{2}$$

$$\frac{C,\ B \leftarrow \overset{z}{\underset{i=1}{\Lambda}}\ \mathrm{A}_i\ , E(C, A_k)}{B \leftarrow A_1 \wedge \ldots A_{k-1} \Lambda A_{k+1} \Lambda \ldots A_z}, \tag{3}$$

$$\frac{C,\ B \leftarrow \overset{z}{\underset{i=1}{\Lambda}}\ \mathrm{A}_i,\ \ E(C, A_k)}{B \leftarrow A_1 \wedge \ldots A_{k-1} \Lambda (A_k \mathrm{V} C) \Lambda A_{k+1} \ldots A_z} \tag{4}$$

It is clear from formulas (2)-(4) that the exclusions are a kind of special-rules inclusions with their effective fields. The interpretation of formula (2) is based on the following: if there exists an exclusion E(C, A_k) that is related to one of the rules with a conclusion B and A_k is its effect then the conjunct A_k must be replaced by $\neg A_k$. In the case when C is not "true" then the corresponding replacement is impossible. The application of the Modus Ponens rule means that the relation between B and $\neg A_k$ leads to a formal logical contradiction.

Therefore the formation of exclusions of the type E(C, A_k) may lead to a contradictory result that is provoked by an incompleteness in the description of the object field. In the case when C is true then the exclusion E(C, A_k) includes this meaning in the conjunct A_k to defeat the meaning of the last conclusions. The result is that A_k is replaced by C because the test of its meaning does not influence the output. In the case when C is true then the corresponding conjunct A_k is directly replaced by C.

Rules of type (1) are united in systems:

$$\begin{cases} B_1 \leftarrow \underset{i \in I}{\Lambda} A_{1i} . \\ B_2 \leftarrow \underset{j \in I}{\Lambda} A_{2j} . \\ \ldots \end{cases} \tag{1A}$$

$$\begin{cases} B_1 < - \underset{i \in I}{\Lambda} A_{1i} . \\ B_2 < - \underset{j \in I}{\Lambda} A_{2j} . \\ \ldots \end{cases} \tag{1B}$$

In the general case the causal-effective relation may be realized using non-classical operations of successions that are denoted '<-' in the paper and B_i may be presented as combinations of sophisticated logical relations (see formula (1B)).

The usage of exclusions (2) up to (4) may be applied also in systems (1A) or (1B); in the general case it reflects the interrelations between different parts of the causal-effective relations influenced by a new information (an exclusion that is attached to one or other group of relations). The new information may influence the mutual relation between the elements of rule (1) or of systems (1A); (1B). In this case the relations of a causal-effective type are defeated or they are strengthened due to an additional information that is contained in the exclusions. The rest of the paper does not include versions (1A) and (1B) because in the majority of our practical applications it is sufficient to confine ourselves to rules (1) thus the algorithmic complexity of the used combination of methods is significantly lowered. By their nature the presented exclusions are an enlarged version of defeasible inferences that is widely used in the intelligent systems. It is a difference from the classical inference with exclusions that in the presented work it is possible not only to exclude the exclusion A_k that is contained in and tailored to the rule but also that we may include in the rule a new formula e.g. $\neg A_k$ in formula (2) or an interrelation between A_k and C in (4). The research also includes versions of formulas using a non-classical negation ~, versions with exclusions of implications influenced by exclusions, etc.:

$$\frac{B \leftarrow \overset{z}{\underset{i=1}{\Lambda}} A_i C, E(C, A_k), \sim A_k \leftarrow C}{B \leftarrow A_1 \wedge A_2 \wedge \ldots A_{k-1} \wedge \sim A_k \wedge A_{k1} \wedge A_{k+1} \wedge \ldots A_z}, \tag{2A}$$

$$\frac{C,\ B \leftarrow \overset{z}{\underset{i=1}{\Lambda}} A_i, E(C, A_k)}{A_1 \wedge \ldots A_{k-1} \wedge A_{k+1} \wedge \ldots A_z}, \tag{3A}$$

$$\frac{C,\ B \leftarrow \overset{z}{\underset{i=1}{\Lambda}} A_i, E(C, A_k)}{A_1 \wedge \ldots A_{k-1} \Lambda A_k \Lambda A_{k+1} \Lambda \ldots A_z}, \tag{3B}$$

where A_{k1} is an additional condition for the transition from $\sim A_k$ to $\neg A_k$. The investigation includes schemas with multi-argument exclusions $E(C,A_k,A_l,\ldots A_s)$ that lead to the simultaneous change of several parts of the rule. The introduced method leads to three basic results: the truth of parts of the rule is altered influenced by the exclusion (if the conditions for activation of the exclusion are enabled), formulas are included in or excluded out of the rule or the rule itself is defeated as it is shown in (3A) or (3B). The results from the research led to a great number of inference versions with exclusions; a part of them is included in our bibliography list.

As it was already shown we introduced a generalized concept of defeating that is based on the following facts. Object scope modeling is a dynamic process. In the act of scope-field completion by the system the old relations between separate parts of the knowledge and/or between different knowledge may be eliminated, changed or their effect may be redirected. This is accomplished influenced by the new knowledge that complete or correct the primary existing knowledge or the interrelations in it. The processes are formalized in the following way.

We did a research of the situations that appear after the addition of new knowledge to the existing knowledge basis and we grouped them in 11 basic groups. Let P is the part of the new knowledge that influences one or more formulas (e.g. see (1) up to (4)).

I. P 'nullifies' A_k: it defeats its relation to the conclusion B. As a result of the defeat A_k has a meaning of 0 and no matter whether it is true or false the true of the conclusion does not change.

$$\frac{B \leftarrow \overset{z}{\underset{i=1}{\Lambda}} A_i, P}{B \leftarrow A_1 \wedge A_2 \wedge \ldots A_{k-1} \wedge A_{k+1} \wedge \ldots A_z, \neg(B \leftarrow \overset{z}{\underset{i=1}{\Lambda}} A_i)},$$

where in difference with defeasible inference schemes the first rule format existing before the appearance of P becomes false.

II. This is an extreme version of the situation from group I when all the atoms in the antecedent are defeated. Now rule (1) turns into a fact: B $\leftarrow$.

$$\frac{B \leftarrow \overset{z}{\underset{i=1}{\Lambda}} A_i, P}{B, \neg(B \leftarrow \overset{z}{\underset{i=1}{\Lambda}} A_i)},$$

III. P changes the true of A_k from true to false or v.v.

$$\frac{B \leftarrow \overset{z}{\underset{i=1}{\Lambda}} A_i, P}{B \leftarrow A_1 \wedge A_2 \wedge \ldots A_{k-1} \wedge \neg A_k \wedge \ldots A_z, \neg(B \leftarrow \overset{z}{\underset{i=1}{\Lambda}} A_i)},$$

IV. P defeats the existing meaning of A_k and increases it to 1. The meaning of the other parts of the antecedent of (1) duly drops down to 0. Independently on the way (conjunctively or disjunctively) they are related to A_k in this situation they are defeated by the antecedent of rule (1).

$$\frac{B \leftarrow \overset{z}{\underset{i=1}{\Lambda}} A_i, P}{B \leftarrow A_k, \neg(B \leftarrow \overset{z}{\underset{i=1}{\Lambda}} A_i)},$$

V. P redirects the relation between the rule and the other knowledge in the domain.

The causal-effective relations are not exhausted by the classical implication and the next example will show that even by formal means it is possible to present different causal-effective relations. Let us have the following two rules:

R_1: B $\leftarrow$ A; R_2: N $\leftarrow$ M.

Let both rules initially be related to the object X. Let also after the appearance of the new set of conclusions P R_1 is related to Y and R_2 to the former object X. In this case the first rule is preserved but its effect is redirected to another object.

For example it is known that by nature a disease is provoked either by a virus or by a bacteria. However let us have a case when a patient manifests simultaneous symptoms of an illness both from a virus and from a bacteria. The sequent investigation (P) shows that the symptoms of a virus-provoked disease are related to the patient's throat and that the bacterial symptoms are related to the patient's lungs. The redirecting of the conclusion that contradicts to the rule from the example and the discovery of the second disease solve the problem from this example. It is possible to redirect whole rules as an analogy to the presented example.

VI. P breaks or amplifies the relation between the rule and the other knowledge in the domain.

The difference with the previous situation V now is either the elimination of the existing relations or the addition of new relations between the existing rules. The very rules are preserved at that.

For example every chess-player must have a good physical condition so that he/she can present himself/herself well in the tournaments. If however the 'examined' chess-player is a computer program – this is the effect from the new information P – then the already said does not at all concern this program.

VII. P influences the conclusion from one or from a group of rules: from R_1: B $\leftarrow$ A into R'_1: $B^* \leftarrow$ A. In this way the old conclusion P is defeated or it is replaced by the new one B^*.

$$\frac{B \leftarrow \overset{z}{\underset{i=1}{\Lambda}} A_i, P}{B^* \leftarrow \overset{z}{\underset{i=1}{\Lambda}} A_i, \neg(B \leftarrow \overset{z}{\underset{i=1}{\Lambda}} A_i)},$$

VIII. The appearance of P changes the antecedent of the examined rule (1). It imports a new atom on the place of A_k, before or after the chosen one A_k. In the last two cases the new atom is conjunctively or disjunctively related to A_k, e.g.

$$\frac{B \leftarrow \bigwedge_{i=1}^{z} A_i, N(P,J)}{B \leftarrow A_1 \wedge A_2 \wedge \ldots A_{k-1} \wedge J \wedge \ldots A_z, \neg(B \leftarrow \bigwedge_{i=1}^{z} A_i)},$$

This situation can be named specifying the antecedent as a result from the new information P.

IX. R_1 is replaced by R_2 influenced by P:

R_1: $B \leftarrow A$; R_2: $N \leftarrow Q$.

The difference from the previous situation here is in the provoked by P complete replacement of the rule in accordance with the a priori defined concepts.

$$\frac{B \leftarrow \bigwedge_{i=1}^{z} A_i, P}{N \leftarrow Q, \neg(B \leftarrow \bigwedge_{i=1}^{z} A_i)},$$

X. We have a situation from I to IX but the obtained consequences may not be used in the antecedents of the other rules. The reasons for similar constraints are different e.g. limiting an insecure information along long chains of rules, etc.

XI. The atoms of the investigated rule (1) remain the same but some of the logical operations are changed affected by P, e.g.

$$\frac{B \leftarrow A_1 \wedge A_2 \wedge \ldots A_{k-1} \wedge \neg A_k \wedge \ldots A_z, N(P,J)}{B \leftarrow A_1 \wedge A_2 \wedge \ldots A_{k-1} \wedge \sim A_k \wedge \ldots A_z, \neg(B \leftarrow \bigwedge_{i=1}^{z} A_i)},$$

A characteristic example of a similar situation is the transformation of the strong classical negation '¬' into a weak paraconsistent negation '~'.

Let us discuss the following illustrative example. On principle it is not possible that a single man is a teacher and a student at the same time. Let us denote that 'John is a teacher' by the variable Q. Then it will not be an error if we denote that 'John is a student' by ¬Q.

This is valid in the prevailing number of situations but it is inapplicable on condition (P) that John is a student in one subject in one school but he is a teacher in other subject in other e.g. sports school. After the advent of the new information P it is not possible to say that 'John is a student' is ¬Q; now it is correct to use the weak negation and ~Q will lead to a contradiction only in the cases when definite conditions hold – in the example the conditions are the subject for teaching and also the location for teaching.

The described situations from I to XI present a research for the influence of the new information P over different parts and relations between existing conclusions. In the majority of the cases the discussed situations may be used contemporary mechanisms for defeasible inference. The difference is just in the fact that P totally changes the existing a priori situation. But if P replaces the literal in the first argument in the exception $E(C, A_k)$ then the exclusion does not change the action progress for the existing up to the advent of P things and it adds to them a new scheme that is activated if and only if when P is false. The present chapter does not contain formalizations of all the possible realizations of the situations from I to XI because the number of their combinations in all the possible realizations is too great.

We propose the application of inference by analogy to increase the effectiveness of searching. This method is viewed in details in [8-10].

2. Analogical Inference Using the Defeasible Schemes

Graph models and network flows play an important role in intelligent systems. Let graph G(N, U) has a set of arcs U and a set of nodes N. It is shown in [10] that the inference by analogy may be presented as a network flow on a graph. The geometric interpretation of this presentation is depicted in fig. 1

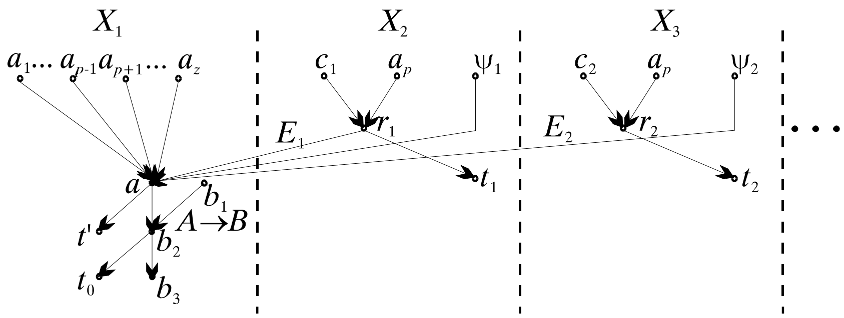

Fig. 1.

Fig. 1 is separated in different regions by dotted lines. Each region contains data corresponding approximately to a single object Xj. The set S contains all elements A_i of the exclusion E and ψ_i. The set of conclusions T contains all t_v, t_0 and t'. Then the following interrelations hold for all $X,y \in N$:

$$f(y,X)-f(X,y)=\begin{cases} v(a_j), \text{ iff } y\in S, \\ 0, \quad \text{ iff } y\notin S,T, \\ v(t_k), \text{ iff } y\in T. \end{cases} \qquad (5)$$

Here A_i corresponds to the stream function $f(a_i,a)$ and C_v, A_p and $E(C_v,A_p)$ to the functions $f(c_i,r_i)$, $f(a_p,r_i)$, $f(r_i,a)$ respectively. The function $f(\psi_i,a)$ has an initial value of 1 if $\psi(X_j,X_1)<T$ or a value of 0 otherwise. Some corresponding exclusions $E(C_v,A_p)$, $i\geq j$ may be included in the knowledge about the object X_j. The pairs of input arcs are disjunctively connected in the nodes r_i and they are conjunctively connected in the nodes *a* and b_2. The functional dependency v has the following form:

$$v(a_j)=f(a_j,a),\ v(tj)=f(a,t_j). \qquad (6)$$

The conjunction of all A_i and ψ_i is denoted with A and it corresponds to the arc (a,b_2). The implication $A\rightarrow B$ is a set of arcs (b_1,b_2) and the result of the inference $f(b_2,b_3)$ possesses a meaning of truth B. Then the inference by analogy may be presented by the following system of equalities and inequalities [6-10]:

$$f(a,b_2)-f(a_i a)\leq 0;\ i=1,...z, \qquad (7)$$

$$f(a,b_2)-f(r_j,a)\leq 0;\ j=1,...n, \qquad (8)$$

$$f(a,b_2)-f(r_j,a)\leq 0;\ j=1,...n, \qquad (9)$$

$$f(r_j,a)-f(c_j,r_j)\geq 0;\ j=1,...n, \qquad (10)$$

$$2f(r_j,a)-f(c_j,r_j)-f(a_p,r_{1j})=0;\ j=1,...n, \qquad (11)$$

$$(2n+z-1)f(a,b_2)-\sum_{i=1}^{z-1} f(a_i,a)-\sum_{j=1}^{n} f(r_j,a)-\sum_{j=1}^{n} f(\psi_j,a)=0\,, \qquad (12)$$

$$2f(b_2,b_3)-f(a,b_2)-f(b_1,b_2)=0, \qquad (13)$$

$$f(r_j,a) = 0 \ \text{ or } \ 1, \qquad (14)$$

$$f(x,y)\geq 0; \quad (x,y)\in U, \qquad (15)$$

$$f(r_j,t_j)\leq 1, \qquad (16)$$

$$f(a,t')\leq 2n+z-2, \qquad (17)$$

$$f(b_2,t_0)\leq 1.$$

In this way the problem of inference by analogy is reduced to a problem of linear programming with a goal function of the kind:

$$\sum_{(x,y)\in D} f(x,y) \to max \tag{18}$$

with constraints (7) – (17). This problem is viewed in details in [10].

The purposes of the analogical defeasible reasoning are two: check-up significance of the selected set of hypotheses and knowledge acquisition by analogy. The idea of the considered scheme of reasoning is to transfer such knowledge from the base into the goal of the transformation that this proposition reduces the significance of the considered part of formula to zero. Before the transfer, the propositions have to pass 'filters'. After filtration of wrong or insignificant information, the resulting information is applicable for the defeasible reasoning or elsewhere.

It follows from the scheme above that the elaborated by us defeasible analogy uses one of the already presented defeasible inferences combined with the inference by analogy with a goal defeating or confirming intermediate results – hypotheses that are inferred by analogy.

3. Logic Derivation in Problems of Search for Regularities and Pattern Recognition

Let us suppose that the information about some plant area has been defined in the form of a database that is interpreted as a learning sample for search (extraction) of logic regularities connecting these data. Let the set $Z=\{X_i,Y_i\}^{m}_{i=1}$ is some database (learning sample) and the data are connected by an unknown dependence of the kind:

$$Y = f(X), \tag{19}$$

where X and Y are multiple-valued predicates. It is required to define a dependence (regularity) (19) on the learning database Z of power m.

First let us see a case of coding for the learning sample by two-valued predicates. In this case the initial object area may be described by rules of productions like

$$\Lambda^{n}_{j=1} x_{ij} \to y_i , j = 1,..., m \ . \tag{20}$$

Every rule of production is an implication, so it may be presented as a perfect disjunctive normal form (PDNF). In the case of a two-valued logic rule the transformation of implication to DNF is executed by the formulas:

$$A \to B = \neg A V B \to \tag{21}$$

Therefore in the case of knowledge coding by two-valued predicates every suggestion may be presented by the rule of production and transformed to PDNF like

$$V^{n}_{i=1} x_i^{\sigma_{ij}} V y_i \ , \tag{22}$$

where $\sigma_{ij\,i}$ is equal 0 or 1.

Further it is required to unite all formulas for the learning sample in a single logic function or system for functions, giving one-valued interpretation of the initial object area. Thus the unknown dependence Y=f(X) may be reconstructed simultaneously on the learning sample Z.

Any logic function that is written in the kind of PDNF may be reduced. Therefore the system of logic knowledge may be also reduced as a rule. Then reducing PDNF corresponding to the logic function may be interpreted as minimizations of the initial database.

We suggest the following algorithm for the PDNF reduction with an account of the object-area speciality:

1. If DNF has single-letter disjunctions x and $\neg x$, DNF is generally significant;

2. If some variable is in DNF with one sign, then delete all disjunctions containing this variable (this variable is non-informative);

3. If DNF has some single-letter disjunction x, then execute the following actions:

a) delete all disjunctions of the kind x Λ ... (rule of absorption);

б) substitute disjunctions of the kind ¬x Λ s ... on disjunctions of the kind s Λ p ... .

As a result of such reduction we obtain “the strongest” logic rules, describing the initial object area.

The described method may be used for learning of concepts (classes) in problems of pattern recognition. The synthesized concepts may be interpreted as axioms of classes (patterns) $A_k(\omega)$ in the object area defined by the learning database. Then the problem for pattern recognition is reduced to a search of a logic derivation using the Robinson method for resolutions or the Maslov back method [11].

The problem of identification for an image $\hat{\omega}$ of k-th class (pattern) on the complex image ω with a logic description D(ω) is reduced to a formula derivation:

$$D(\omega) \rightarrow \exists\hat{\omega}A_k(\omega)\,,\ \hat{\omega} \in \omega \tag{23}$$

The meaning of this formula is in the following: a complex image ω with a logic description D(ω) contains an image $\hat{\omega}$ of k-th class on which the axiom $A_k(\hat{\omega})$ is true. It allows to identify automatically and localize (select) the image of k-th class (pattern) on a complex image containing images (patterns) from M different classes $S_1,S_2,\ldots,S_M$.

Multiple applications of the logic-axiomatic method with every k=1,2,…,M allow to recognize (classify) all images of all classes, located on the complex image [11].

4. Multiple-Valued and Probabilistic Logics in Problems for Learning and Search of Regularities

The described method for search of logic regularities may be generalized on a case of multiple-valued coding for back samples and a search for multiple-valued regularities. The use of multiple-valued logics is complicated by the ambiguity of interpretation for functions of negation, implication, etc. Therefore let us discuss the most general variant in a case of use of three-valued logic.

Let a set of values for truth has the kind {0 1 2} with the following interpretation:

x=0 – false, x=1 – nonsense (indefinite), x=2 – truth.

Then let us introduce the concept of inversion as ¬x= 1V0, i.e. negation of truth may be either false or nonsense. This concept is defined by Table 1. This definition of inversion provides the inclusion of all possible interpretations of inversion in different logics.

Table 1

X	¬X
0	1V2
1	0V2
2	0V1

Some functions of three-valued logics are introduced. The most important of them are the characterizing functions, defined in the following way:

$$I_i(x) = \begin{cases} k-1, & if\ \ x = i, \\ 0, & if\ \ x \neq i, \end{cases} \tag{24}$$

$$J_i(x) = \begin{cases} 1, & at\ \ x = i, \\ 0, & at\ \ x \neq i, \end{cases} \tag{25}$$

The main rules of operation with these functions have the kind:

$$I_\sigma(x)I_\tau(x) = \begin{cases} I_\sigma(x), & if\ \sigma = \tau, \\ 0, & if\ \sigma \neq \tau, \end{cases} \tag{26}$$

$$\sigma \wedge \tau = \min(\sigma,\tau), \qquad \sigma \vee \tau = \max(\sigma,\tau). \tag{27}$$

Let us use also a two-valued analogue of implication in the discussed three-valued logic, i.e.

$$A \to B = \neg A \vee B = I_0(A) \vee I_1(A) \vee B \tag{28}$$

The form (28) as a negation is an extension that includes in itself a series of possible implications of a three-valued logic. Such a wide definition of main functions of logic is convenient for modelling intelligent systems in cases when it is not possible to describe intelligent processes by some concrete multiple-valued logic.

Let us return to the solution of the initial problem in the terms of the three-valued logics. Also let every line in the learning sample be described by rules of production:

$$\Lambda^{n}_{j=1} X_{ij} \to Y_i, \; i = 1,..., m \;. \tag{29}$$

Then the analogue of PDNF will be the following function of three-value logic:

$$I_0(I_l(x)) = \begin{cases} I_1(x).. \vee I_{l-1}(x) \vee I_{l+1}(x).. \vee I_m(x), & if \quad x = l, \\ 0 \quad if \quad x \neq l & . \end{cases} \tag{30}$$

$$V^{m}_{j=1} I_0(I_i(x)) \vee Y_i, \tag{31}$$

Because every regularity (knowledge) corresponding to the learning sample may be written in the kind of the suggested function of three-value logic, we want to have the possibility to present all regularities, forming the database, by a function or a system function of three-valued logic.

Single-value correspondence is easy to obtain if, for example, we multiply logically the rules of productions. It corresponds to discussions of the following type: we know partial (local) rules and thus we know all local rules (regularities) determining the global knowledge base built by the learning sample.

As a result we will obtain a three-valued function that determines the desired regularity. This function can be obtained if we use an adapted version of a reducing algorithm for multiple-valued logic as follows:

1. If some variable is in DNF with one sign ($I_j(x)$, j=const, in all disjunctions), then delete all disjunctions, containing this variable (this variable is non-informative);

2. If DNF has some single-letter disjunction $I_j(x)$, then execute the following actions:

   a) delete all the disjunctions of the kind $I_j(x) \Lambda$ ... (rule of absorption);

   b) substitute disjunctions of the kind $I_i(x) \Lambda$ s... ($i \neq j$) by the disjunctions of the kind s Λ p ... .

The result of the algorithm is a multiple-valued function built by the initial learning sample, characterizing it by a single value and giving a set of the most significant rules (regularities) defining the initial knowledge area.

By the addition of a new rule of production (new knowledge) we check if a given rule may be derived from the already existing ones or not. If it is possible to derive this rule then the function remains the same. Otherwise the knowledge base shall be enlarged adding a new rule (regularity) by a multiple-valued logic multiplying of the existing function and a new production written in the kind of a multiple-valued PDNF.

The other method of learning for concepts and search of multiple-valued regularities on defined databases is based on local-optimal logic-probabilistic algorithms [12,13]. It provides automatic synthesis, optimization (by precision) and complexity minimization for knowledge bases in terms of multiple-valued predicates with a non-defined valuation by learning databases. It allows the interpretation and the realization of synthesized knowledge (regularities) in the form of three-layer or multi-layer neural networks of a polynomial type with a self-organizing architecture [14,15].

Conclusions

An approach is introduced for inference by analogy based on three-valued logics and network flows. The approach is oriented at applications in systems of artificial intelligence and maintenance of decision making. The discussion fixes the peculiarities and the general characteristics of different types of inference by analogy.

A method is elaborated where the suppressed proof is formalized as a network flow. This approach reduces the problems of logic programming to the corresponding problems of linear programming.

The difference is investigated between the logics of the type 'knowledge about changing knowledge' and logics using different types of exclusions (defeasible inference).

The multiple-valued logic approach may be applied to solutions of learning problems and regularities searches in databases permitting the identical description of the object area, to structural analyses of the initial information, to reductions of it and to its changes by a measure of forming a new knowledge that is not derived from the initial data.

Logic-axiomatic and logic-probabilistic learning methods for concepts and pattern recognition have been generalized on a case of a multiple-valued logic. It is shown that synthesized concepts and recognizing rules may be realized in the kind of multiple-valued neural networks of a polynomial type and used in systems of intelligent and neural control [13-14].

The work is done under a partial support of the grants no. NIP917 of the Ministry of Science and Education – Republic of Bulgaria, RFBR N03-01-00224 - Russia, and the grant RHSF N03-06-12019v-Russia.

Bibliography

[1] P.Dung, P. Moncarella, Production systems with negation as failure, IEEE, Trans/ Knowledge and Data Engineering, vol. 14, no.2, pp. 336-352, April 2002.

[2] N. Rescher, Many Valued Logics, Basil Blackwell, Oxford 1979.

[3] Яблонский С.В. Введение в дискретную математику. Москва. Высшая Школа, 2001

[4] V. Gladun, Process of new knowledge formation. Pedagog 6, Sofia, 1994.

[5] Z. Markov, Inductive Machine Learning Methods, TEMPUS JEN 1497, Softex, Sofia, 1996.

[6] V. Jotsov, Knowledge discovery and data mining in number theory: some models and proofs, Proc. Methods and Algorithms for Distributed Information Systems Design. Institute for Information Transmission Problems of RAS, 1997, pp.197-218.

[7] K. Kowalski, Logic for Problem Solving, North-Holland Inc., Amsterdam, 1979.

[8] Halpern, J., M. Rabin, A logic to reason about likelihood, Artificial Intelligence, 1987, no. 32, pp. 379-405.

[9] V.Sgurev, V.Jotsov, Some characteristic features of the application of three-valued logical schemes in intelligent systems, Proc. First International IEEE Symp. 'Intelligent Systems' (T. Samad and V. Sgurev, Eds.), Vol. I., Varna, Bulgaria, September 10-12, 2002, pp. 171-176.

[10] V.Sgurev, V.Jotsov, Some defeasible inference methods by using network flows, J. Problems of Engineering Cybernetics and Robotics, 2001, no. 51, pp. 110-116.

[11] A.Timofeev, T.Kossovskaya, Logic-Axiomatical Method for Knowledge Representation and Recognition in Intelligent Manufacturing Systems, Elsevier, Amsterdam – Oxford – New York, 1990, pp. 3-6.

[12] A.Timofeev, Z.Shibzuchov, The Synthesis and Optimization of Knowledge Bases Based on the Local-Optimization Logic-Probabilistic Algorithms. Int. Journal of Information Theories and Applications, 1995, vol. 3, no. 2.

[13] A.Timofeev, R.Yusupov, Evolution of Intelligent Control in Adaptive Systems. Int. Journal of Adaptive Control and Signal Processing, 1992, vol. 6, pp.193-200.

[14] Тимофеев А.В., Шеожев А.В., Шибзухов З.М. Синтез нейросетевых архитектур по многозначному дереву решений. – Нейрокомпьютеры: разработка и применение, 2002, № 5-6, с. 44-49.

[15] Timofeev A.V. Polynomial Neural Networks with Self-Organizing Architecture, Int. Journal on Optical Memory and Neural Networks, 2004, v.1.

Authors' Information

Vasil Sgurev, Vladimir Jotsov – Institute of Information technologies of the Bulgarian Academy of Sciences; P.O.Box 161, Sofia 1113, Bulgaria; sgurev@bas.bg, jotsov@ieee.org

Adil Timofeev – Saint-Petersburg Institute for Informatics and Automation of RAS, 199178, Saint-Petersburg, 14-th Line, 39; tav@iias.spb.su