KYBERNETIKA — VOLUME 40 (2004), NUMBER 3, PAGES 317 - 332

AN ILP MODEL FOR A MONOTONE GRADED CLASSIFICATION PROBLEM

PETER VOJTÁŠ, TOMÁŠ HORVÁTH, STANISLAV KRAJČI
AND RASTISLAV LENCSES

Motivation for this paper are classification problems in which data can not be clearly divided into positive and negative examples, especially data in which there is a monotone hierarchy (degree, preference) of more or less positive (negative) examples. We present a new formulation of a fuzzy inductive logic programming task in the framework of fuzzy logic in narrow sense. Our construction is based on a syntactical equivalence of fuzzy logic programs FLP and a restricted class of generalised annotated programs. The induction is achieved via multiple use of classical two valued induction on α-cuts of fuzzy examples with monotonicity axioms in background knowledge, which is afterwards again glued together to a single annotated hypothesis. Correctness of our method (translation) is based on the correctness of FLP. The cover relation is based on fuzzy Datalog and fixpoint semantics for FLP. We present and discuss results of ILP systems GOLEM and ALEPH on illustrative examples. We comment on relations of our results to some statistical models and Bayesian logic programs.

Keywords: graded classification, ILP, annotated programs
AMS Subject Classification: 68T37, 03B70, 03B50

1. INTRODUCTION AND MOTIVATION

In a standard logical framework, we are restricted to representing only facts that are true absolutely. Thus, this framework is unable to represent and reason with imperfect (uncertain, vague, noisy,...) information. This is a significant gap in the expressive power of the framework, and a major barrier to its use in many real-world applications. Imperfection is unavoidable in the real world: our information (and particularly our classification) is often inaccurate and always incomplete, and only a few of the "rules" that we use for reasoning are true in all (or even most) of the possible cases.

This limitation, which is critical in many domains (e.g., medical diagnosis), has led over the last decade to the resurgence of probabilistic reasoning in artificial intelligence. Probability theory models uncertainty by assigning a probability to each of the states of the world that an agent considers possible (see [7]).

Many valued logic. Besides probabilistic models there is an extensive study of these phenomena in many valued logic. In this paper we concentrate especially on fuzzy logic programs FLP and generalised annotated programs GAP.

The basic syntactic concept of FLP is $A \cdot x$ an atom graded by a real number $x \in [0,1]$. In GAP it is an atom with an annotation term $A : t$ where $t \in [0,1]$.

The semantics in both approaches are given by a mapping $f : B_L \longrightarrow [0,1]$ where B_L is the Herbrand base. The semantical satisfaction of atoms in both FLP and GAP is the same: $f \models A \cdot x$ (resp. $A : t$) iff $f(A) \geq x$ (resp. $f(A) \geq t$).

FLP is a truth functional logic (i. e. we work in Hajek's fuzzy logic in a narrow sense, see [8])). The truth value of a formula is calculated using truth value functions of many valued logical connectives. In contrast, GAP builds complex formulas from annotated atoms using two valued logical connectives but the structure of annotation terms is more complex.

We can say roughly, that what is hidden in many valued logical connectives of FLP the same expressive power is hidden in the structure of the annotation terms of GAP (see [17]). In [16] it is shown that FLP is more suitable for deductive data models and the purpose of this paper is to show that GAP is more suitable for induction.

Inductive logic programming. In a two valued logic the Inductive logic programming (ILP) task is formulated as follows:

Given is a set of examples, i. e., tuples that belong to the target relation p (positive examples) and tuples that do not belong to p (negative examples). Given are also background relations (or background predicates) q_i that constitute the background knowledge and can be used in the learned definition of p. Finally, a hypothesis language, specifying syntactic restrictions on the definitions of p is also given (either explicitly or implicitly). The task is to find a definition of the target relation p that is consistent and complete. Informally, it has to explain all the positive and none of the negative tuples.

Fuzzy Inductive logic programming. In many valued logic (especially in fuzzy logic) formulas (facts and rules) are assigned a truth value (typically) from the unit interval of real numbers $[0,1]$. This gives the logic a comparative notion of truth. The explicit numerical value of truth assignement is often of small importance. Important is, that a fact or rule has truth degree bigger than the other one.

Motivation for this paper are classification problems in which data can not be clearly divided into positive and negative examples, especially data in which there is a monotone hierarchy (degree, preference) of more or less positive (negative) examples. This corresponds to fuzzy set of examples $E : \{p(c), c \in \mathrm{dom}(p)\} \longrightarrow [0,1]$. We assume also on the side of background knowledge a monotone graded (comparative) notion of fulfillment. This corresponds to fuzzy background knowledge in the form of a definite logic program (without negation) in which rules are assigned a truth value. We consider two illustrative examples of this sort.

Example 1. Our first example is from the psychological praxis and considers the problem of graded classification of pairs of men and women regarding the chance of forming a good marriage (whatever this means and how it is measured, it is just a subjective preference given by an expert).

The set of examples is a set of ground atoms divided into four classes indicating the height (degree) of experts certainty in the success of their (eventual) marriage. The lowest assigned with 0, the second with 0.33, higher with 0.66 and the best chance with 1. E. g. experts confidence in the success of the marriage between woman $w1$ and man $m4$ is $gm(w1, m4) = 0.66$ and hence higher than 0.33 for partnership of $w1$ with $m1$. So, our example set is a mapping E : ManWomenPairs $\longrightarrow$ $\{0, 0.33, 0.66, 1\}$. Nevertheless we have to emphasize that it is not a classification problem of four disjoint classes, it is about classification of a monotone descending (by inclusion) chain of classes expressing success at least 0.33 (or bigger), at least 0.66 or 1.

Background knowledge contains a crisp part (i. e. two values yes-no statements) and also a graded part. The crisp part consists of predicates **a**ge(Person, AgeInYears), **o**ccupation(Person, Occupation) and **h**obby(Person, Hobby) (predicates will be abbreviated to highlighted part e. g. age(m1,33) to a(m1, 33)).

The fuzzy (graded) part of background knowledge is expressing (possibly another) expert's opinion of relevant degrees of similarity (compatibility) of **c**ompatibility**a**ge ca(Age1, Age2), **c**ompatibility**o**ccupation co(Occupation1, Occupation2) and **c**ompatibility**h**obby ch(Hobby1, Hobby2).

There e. g. if one loves to read books and the second to play/watch football, the degree of compatibility of their hobbies $ch(\text{books}, \text{football}) = 0.1$, that is much smaller than 0.8 for the combination on book lover with a cinema fan and equal for his/her combination with swimmer (remember that numerical values are just a way of describing the order).

Further, the background knowledge table for compatibility of occupation assigns co(driver,teacher) = 0.6, that is slightly more that 0.5 = co(director,teacher).

In our example, compatibility for age (in years) is calculated by a formula $ca(y1, y2) = \max(0, 1 - |y1 - y2|/20)$

We expect our graded ILP system will provide us with rules guaranteeing the success of marriage at certain (or higher) level of the form, e. g.

IF h(P1,H1), h(P2,H2), o(P1,O1), o(P2,O2), a(P1,A1), a(P2,A2)
AND compatibility of occupation co(O1,O2) is at least 0.5
AND compatibility of ages ca(A1,A2) is at least 0.4
AND compatibility of hobby ch(H1,H2) is at least 0.6
THEN we can guarantee the success of an eventual marriage of person P1 with person P2 in a degree at least 0.5 (or higher).

Notice that we assume a positive (monotonic, increasing) influence of background factors on the degree of classification (this will correspond to definite logic programming).

Example 2. Second example is from a high school system containing information on grades of students in several topics (in Slovak and Czech Republic 1 denotes the

best and 5 denotes the lowest achievement). Here the monotonicity comparative notion of grades is present, who knows math in degree at least 2 (or better with 1) he/she fulfills also requirements for evaluation 3, 4 and 5 (weaker).

We would like to induce rules describing dependencies between students achievements in different topics, say in a class or a bigger group. We expect our model to be able to extract rules of the form

IF the grade from chemistry is at least 2
AND the grade from biology is at least 1 (the best)
THEN the grade from physics is at least 2 (or better 1).

Here, grades from all subjects form a fuzzy set, e. g. Physics : Students $\longrightarrow \{1,2,3,4,5\}$. In one ILP task the fuzzy set *Physics* plays the role of graded examples and evaluations from all other subjects will form fuzzy background knowledge. In an second task the role of graded classification to be explained is represented by the fuzzy set of grades from e. g. biology and all other, including physics play the role of the background knowledge.

2. DEFINITE FUZZY LOGIC PROGRAMMING

Fuzzy sets. Having a relational schema $r(A_1,\ldots,A_n)$ with attribute domains $D_1,\ldots,D_n$ a fuzzy extension of this schema has instances of the form $R : D_1 \times \ldots \times D_n \longrightarrow [0,1]$.

A standard equivalence relation over D is a transitive,reflexive and symmetric binary relation over D. Here we do not go into discussion what does it mean e. g. transitive in fuzzy case. For our purpose an arbitrary fuzzy binary relation given by expert will play the role, and we call it fuzzy compatibility relation.

Fuzzy logic. Our language $\mathcal{L}$ has two types of syntactical objects: logical and quantitative. The logical part consists of a many-sorted predicate language without function symbols. The quantitative consists of some/all real numbers from the unit interval [0,1].

As motivated in [28], our language has finitely many different connectives of each sort, conjunctions, disjunctions, implications and aggregations. The truth function of a connective is denoted by a dot(circle) in the upper right index. A truth function for a conjunction $\wedge$ is a conjunctor $\wedge^\bullet : [0,1]^2 \to [0,1]$ and for disjunction $\vee$ a disjunctor $\vee^\bullet : [0,1]^2 \to [0,1]$ which are assumed to extend the respective two valued connectives and are order preserving in both coordinates.

Truth function for an j-ary aggregation @ is an aggregation operator $@^\bullet : [0,1]^j \to [0,1]$ which fulfills $@^\bullet(0,\ldots,0) = 0$ and $@^\bullet(1,\ldots,1) = 1$.

Truth function for an implication $\to$ is an implicator $\to^\bullet$: $[0,1]^2 \to [0,1]$, which is non-increasing in the first (body) coordinate and non-decreasing in the second (head) coordinate and extends the two valued implication.

Interpretations are based on fuzzy relations. Our logic is truth functional, meaning that every interpretation of the language can be extended to all formulas calculating the truth value of formulas along their complexity from truth values of atoms using truth functions of connectives.

The main syntactical object of our language are graded formulas (φ, β), where φ is a formula and β is a rational number from [0,1].

An interpretation f is a model of (φ, β) if $f(\varphi) \geq \beta$.

The theory of fuzzy logic programming. We formulate properties of residuation.

Definition. Let C a conjunctor and I be an implicator. In what follows, b, h, r are universaly quantified and range through $[0, 1]$. We define following properties of C and I:

(a)(C, I)	$r \leq I(b, h)$	iff	$C(b, r) \leq h$
$\Phi 2(C, I)$	$C(b, I(b, h)) \leq h$		
$\Phi 3(C, I)$	$r \leq I(b, C(b, r))$.		

Observation. (a)(C, I) iff ($\Phi 2(C, I)$ and $\Phi 3(C, I)$).

Observation. Assume (a)(C, I) then $I(b, h) = \sup\{r : C(b, r) \leq h\}$ (denote this implicator by I_C and call it the residual implicator to the conjunctor C) and $C(b, r) = \inf\{h : I(b, h) \geq r\}$ (denote this conjunctor by C_I and call it the residual conjunctor to the implicator I).

Observation. Given C, then there is an I such that (a)(C, I) iff C is left continuous in the rule coordinate.

Observation. Given I, then there is a C such that (a)(C, I) iff I is right continuous in the head coordinate.

Proofs of these observations are outside the scope of this paper.

In our computational model, we have conjunctors $C_1, \ldots, C_n$ which are residual to above implications. These need not be truth functions of any conjunctions in our language. We assume conjunctors are left continuous.

Any formula built from atoms using conjunctions, disjunctions and aggregations is called a body. Every composition of conjunctors, disjunctors and aggregation operators is again an aggregation operator. Hence, without a loss of generality, we can assume that each body is of the form $B = @(B_1, \ldots, B_n)$. A rule of FLP is a graded implication $H \longleftarrow @(B_1, \ldots, B_n).r)$, where H is an atom, $@(B_1, \ldots, B_n)$ is a body and $r \in [0, 1]$ is a number. A fact is a graded atom $(A.a)$.

A definite (i. e. without negation) fuzzy logic program is a partial mapping P : Formulas $\to (0, 1]$ with the domain of P, dom(P) consisting only of atoms and logical parts of FLP rules. The quantitative part of the rule is $r = P(H \leftarrow @(B_1, \ldots, B_n))$. Let B_L be the Herbrand base. A mapping $f : B_L \to [0, 1]$ is said to be a fuzzy Herbrand interpretation. Satisfaction for rules means that

$$f(H \longleftarrow @(B_1, \ldots, B_n)) = \longleftarrow^{\bullet} (f(H), @^{\bullet}(f(B_1), \ldots, f(B_n))) \geq r.$$

Recall the many-valued modus ponens (a correct rule in fuzzy logic) reads as

$$\frac{(B \cdot b), (H \leftarrow_i B \cdot r)}{(H.C_i(b,r))}.$$

We base our procedural semantics on the "backward usage of modus ponens" (no refutation nor resolution is applied here). A computation looks like

$?-H$
$C_i(@^\bullet(B_1,\ldots,B_n),r)$
$\ldots$
$C_i(@^\bullet(b_1,\ldots,b_n),r).$

Namely, a query H is (after a successful unification with the head of a rule) replaced by a mixed expression $C_i(@^\bullet(B_1,\ldots,B_n),r)$ – the initial segment of the term calculating the truth value of the answer $C_i(@^\bullet(\_),y)$ with embedded atoms from the body $B_1,\ldots,B_n$ not computed so far. After all truth values $b_1,\ldots,b_n$ of all atoms $B_1,\ldots,B_n$ are calculated, we evaluate the whole expression.

We know by the residuality of C_i that this is a sound rule (see [8, 28]).

We define the corresponding production (datalog or cover) operator T_P operator (for $f : B_P \to [0,1]$) by

Definition. Assume P is a fuzzy logic program. Then

$$T_P(f)(A) = \max\{\sup\{C_i(f(B),r) : (A \leftarrow_i B \cdot r)$$

is a ground instance of a rule in the program $P, \sup\{a : (A \cdot a)$ is a ground instance of a fact in the program $P\}\}$.

We know that this operator is continuous (under our conditions, more on continuity see [28] and [17]) and it's fixpoint is the minimal model of the definite program P. Hence we can base the cover relation of our ILP system on this fixpoint semantics

$$\mathrm{Cover}(P) = T_P^\omega(0).$$

3. GENERALIZED ANNOTATED LOGIC PROGRAMMING

Kifer and Subrahmanian [14] introduced generalized annotated logic programs (GAP) that unify and generalizes various results and treatments of multivalued logic programming. The whole theory of GAP is developed in a general setting for lattices. We restrict ourselves to the unit interval of real numbers $[0,1]$.

Definition. A function $a : [0,1]^i \to [0,1]$ is an annotation function if it is left continuous and order preserving in all variables.

The language of annotated programs consists of a usual language of predicate logic as in FLP and of the quantitative part of the language. The quantitative part of the language has annotation variables and a set of basic annotation terms of different arity. Every annotation term ρ is a composition of annotation functions.

Notice, that $\rho^\bullet$ can be considered as the truth function of an aggregation operator. If A is an atomic formula and α is an annotation term, then $A : \alpha$ is an annotated atom.

Definition. If $A : \rho$ is a possibly complex annotated atom and $B_1 : \mu_1, \ldots, B_k : \mu_k$ are variable annotated atoms, then $A : \rho \longleftarrow B_1 : \mu_1 \wedge \ldots \wedge B_k : \mu_k$ is an annotated clause. We assume that variables occurring in the annotation of the head also appear as annotations of the body literals and different literals in the body are annotated with different variables.

Definition. Let B_L be the Herbrand base. A mapping $f : B_L \to [0,1]$ is said to be a Herbrand interpretation for annotated logic. Note that interpretation for fuzzy logic and interpretations for annotated logic coincide.

The satisfaction is defined differently (all variables (object and annotation) are implicitly universally quantified). Suppose f is an Herbrand interpretation. Then,

f satisfies a ground atom $A : \rho$ iff $\rho \leq f(A)$

f satisfies $(F \wedge G)$ iff f satisfies F and f satisfies G

f satisfies $(F \vee G)$ iff f satisfies F or f satisfies G

f satisfies $(F \leftarrow G)$ iff f satisfies F or f does not satisfy G.

Definition. (FLP and GAP transformations) Assume $C = A : \rho \longleftarrow B_1 : \mu_1 \wedge \ldots \wedge B_k : \mu_k$ is an annotated clause. Then $flp(C)$ is the fuzzy rule $A \longleftarrow \rho(B_1, \ldots, B_n) \cdot 1$, here ρ is understood as an n-ary aggregator operator.

Assume $D = A \longleftarrow_i @(B_1, \ldots, B_n).r$ is a fuzzy logic program rule. Then $gap(D)$ is the annotated clause $A : C_i(@^\bullet(x_1, \ldots, x_n), r) \longleftarrow B_1 : x_1, \ldots, B_n : x_n$.

Theorem. (See [17].) Assume C is an annotated clause, D is a fuzzy logic program rule and f is a fuzzy Herbrand interpretation. Then

f is a model of C iff f is a model of $flp(C)$

f is a model of D iff f is a model of $gap(C)$.

This theorem is the main tool in our formal model of fuzzy ILP.

4. A FUZZY ILP MODEL

The classical ILP can be described more formally: given is a set of examples $E = E^+ \cup E^-$, where E^+ contains positive and E^- negative examples, and background knowledge B. The task is to find a hypothesis H such that $\forall e \in E^+ : B \cup H \models e$ (H is complete) and $\forall e \in E^- : B \cup H \not\models e$ (H is consistent). This setting, introduced by Muggleton [21], is also called learning from entailment. In an alternative setting proposed by De Raedt and Džeroski [25], the requirement that $B \cup H \models e$ is replaced by the requirement that H be true in the minimal Herbrand model of $B \cup \{e\}$: this setting is called learning from interpretations (see Džeroski, Lavrač [6]).

Fuzzy logic has the logical part identical to that of classical (two valued logic). A straight forward way how to fuzzify different models, is to consider the same formulas as in the specification of the classical model just the truth values range through a bigger set (e. g. the unit interval of real numbers $[0,1]$), truth functions of connectives should be specified more carefully, and we have a fuzzy model and we can study it. It is possible to formulate the fuzzy ILP problem this way, consider two fuzzy sets μ_{E^+} and μ_{E^-}, translate the definition of their disjointness, and so on. In our case, there is no E^+ nor E^-, there is only one fuzzy set E. We try to justify our model of FILP by real world examples. At the end we will discuss and compare it to other fuzzy ILP models and also probabilistic.

We transfer the problem of fuzzy ILP with fuzzy background knowledge and fuzzy set of examples to several crisp ILP problems. First, assume B is a fuzzy background knowledge consisting of fuzzy facts (so far we do not discuss the case of fuzzy rules in B). Then $c(B)$ is the crisp knowledge acquired from B by adding an additional attribute for the truth value to each predicate (e. g. we transform $ch(\text{football, theatre}) = 0.3$ to $ch(\text{football, theatre}, 0.3)$. Second, for every $\alpha \in [0,1]$ in the range of our fuzzy set of examples, we put E^+_α to be the upper ($\geq$) and E^-_α to be the lower ($<$) α cut of the fuzzy set E. For each α consider the classical ILP task with E^+_α, E^-_α and $c(B)$ and returns a crisp set of hypothesis H_α . Our aim is to have a formal model allowing us correctly to answer the fuzzy ILP problem (E, B) with the fuzzy set of hypothesis H, such that for all α, the α cut of H is exactly H_α.

Classical (crisp) ILP systems in the two valued logic use the language of Horn clauses. Notice, that in fuzzy logic the implication $A \to B$ need not be always equivalent with $A \vee \neg B$ and in our model of FLP the rules are implications. Nevertheless, working with α cuts the ILP cover turns to greater or equal value in fuzzy hypothesis glued from single H_α together.

An ILP system GOLEM. In order to search the space of relational rules (program clauses) systematically, it is useful to impose some structure upon it, e. g. an ordering. One such ordering is based on subsumption (clause C subsumes C' if there exist a substitution θ, such that $C\theta \subseteq C'$). The space of hypothesis ordered by a subsumption is a lattice. The rules for computing the operations are outlined in [6]. System GOLEM uses background knowledge B restricted to ground facts. To search the space of hypothesis uses a derived notion of least general generalisation relative to background knowledge B ([10]).

Using data from Example 1 the accuracy of GOLEM was below 100 % and due to the fact that this systems allows only atoms in B, an interesting phenomenon appeared. On the level $\alpha = 0.66$ the system encountered (besides others) following rules, e. g.:

$$gm(A,B) \leftarrow h(A,C), h(B,D), ch(C,D,0.70).$$
$$gm(A,B) \leftarrow h(A,C), h(B,D), ch(C,D,0.80).$$
$$gm(A,B) \leftarrow h(A,C), h(B,D), ch(C,D,0.90).$$

from which the last two rules are useless, because the many valued model of the later rules is also a model of the first rule. Another problem is, that usual ILP systems understand these numbers like a syntactic objects and they do not distinguish the ordering between them.

Practically, for different granulation of the interval $[0,1]$ we added to B the less than relation

$\ldots leq10(0.4,0.5) \cdot leq10(0.5,0.6) \cdot leq10(0.6,0.7) \cdot leq10(0.7,0.8)\ldots$

$\ldots$

$\ldots leq100(0.4,0.45) \cdot leq100(0.45,0.5) \cdot leq100(0.5,0.55) \cdot leq100(0.55,0.6)\ldots$

$\ldots$

and monotonicity rules

$$co(A,B,C) \leftarrow leq10(C,D), co(A,B,D).$$
$$ca(A,B,C) \leftarrow leq100(C,D), ca(A,B,D).$$
$$ch(A,B,C) \leftarrow leq10(C,D), ch(A,B,D).$$

For this purpose we have used the ILP system ALEPH [9], which is based on the inverse entailment [21]. In this case, the background knowledge can contain rules.

The search of the lattice of hypothesis (depending on example) starts at an saturation element influenced by our monotonicity rules in background knowledge reducing the search, and hence also increasing accuracy.

Results of the ILP system ALEPH for good marriage example had 100 % accuracy and following rules were produced (lower index is α, the upper is numbering)

$R^1_{1.0} = gm(A,B) \leftarrow a(A,C),\ a(B,D),\ ca(C,D,.9),\ h(A,E),\ h(B,F),\ ch(E,F,.8).$
$R^2_{.66} = gm(A,B) \leftarrow a(A,C),\ a(B,D),\ ca(C,D,.45),\ h(A,E),\ h(B,F),\ ch(E,F,.6).$
$R^3_{.66} = gm(A,B) \leftarrow o(A,C),\ o(B,D),\ co((C,D,.9).$
$R^4_{.66} = gm(A,B) \leftarrow a(A,C),\ a(B,C).$
$R^5_{.66} = gm(A,B) \leftarrow h(A,C),\ h(B,D),\ ch(C,D,.7).$
$R^6_{.66} = gm(A,B) \leftarrow a(A,C),\ a(B,D),\ ca(C,D,.8),\ h(A,E),\ h(B,F),\ ch(E,F,.3).$
$R^7_{.33} = gm(A,B) \leftarrow h(A,C),\ h(B,D),\ ch(C,D,.1).$

These hypotheses found by ALEPH are complete and consistent (correct), while in the case of GOLEM they were not covering all positive examples (no negative were covered by either of them). Adding the monotonicity rule into background knowledge substantially reduced the search space of ALEPH and hence also the accuracy of ALEPH was 100 %, often much higher than that of GOLEM.

Glueing hypothesis together.

Rules from the ALEPH contain in the body a block of two valued predicates

$block = h(A,C) \wedge h(B,D) \wedge a(A,E) \wedge a(B,F) \wedge o(A,G) \wedge o(B,H)$

with bindings of attributes to persons A and B and are always true (this illustrates the multirelational character of our application). Conditions on compatibility of attributes of persons are hidden in crisp representation of fuzzy background knowledge in form of

$ch(C,D,x) \wedge ca(E,F,y) \wedge co(G,H,z)$.

If some of these compatibility conditions is not present we can interpret it as having value 0.

Moreover rule obtained on the level α guarantees the result in degree α, so it corresponds to a fuzzy logic program rule with truth value α (because in body there

are crisp predicates and the boundary condition of our conjunctors fulfill $C(x,1) = x$. The first rule $R^1_{1.0}$ corresponds to fuzzy rule

$gm(A,B) \leftarrow block \wedge ch(C,D,0.8) \wedge ca(E,F,0.9) \wedge co(G,H,0).1$

The second rule $R^2_{0.66}$ says

$gm(A,B) \leftarrow block \wedge ch(C,D,0.6) \wedge ca(E,F,0.45) \wedge co(G,H,0).0.66$

and so on.

Here we see limitations of fuzzy logic programming in the induction, we are not able to glue them to one hypothesis.

On the other side, these rules define a single annotation term – a function of three real variables – $a(x,y,z)$ defined as follows:

If the system for selected α has induced a rule

$gm(A,B) \leftarrow block \wedge ch(C,D,x) \wedge ca(E,F,y) \wedge co(G,H,z).$

then the function $a(x,y,z) = \alpha$.

If there is no such rule then the function is the smallest monotone function extending those points, i. e. $a(x,y,z) = \max\{a(x_1,y_1,z_1) : x_1 \leq x, y_1 \leq y, z_1 \leq z\}$.

Another challenging problem is to learn the function a, methods of [29] could be appropriate.

Result and formulation of the fuzzy/annotated ILP task for graded examples.

Given the fuzzy set of examples E from Example 1 and fuzzy background knowledge B containing positive literals. If f is a fuzzy Herbrand interpretation which is a model of the fuzzy theory B and the single GAP rule forms a (one element) hypothesis

$H = gm(A,B) : a(x,y,z) \leftarrow block \wedge ch(C,D) : x \wedge ca(E,F) : y \wedge co(G,H) : z$

then f is a model of the fuzzy theory E. Moreover for the minimal fuzzy model f_m of $B \cup H$ we have for all e_1 and e_2 the following order preserving approximation condition:

$E(e_1) < E(e_2)$ if and only if $f_m(e_1) < f_m(e_2)$

and

$E(e_1) = E(e_2)$ if and only if $f_m(e_1) = f_m(e_2)$.

On more complex data this would be a too strong condition. The idea of our formulation of fuzzy and/or annotated ILP task is to have a weak order preservation of learned function – namely examples can be glued together, can be treated off but should not be inverted – that is the learned hypothesis should not give a higher preference to an example e_1 than to e_2 in the case when in input data e_2 has higher preference than e_1.

A weak order preserving fuzzy/annotated ILP problem.

Given a fuzzy set of examples E and fuzzy background knowledge B containing positive literals, we look for a fuzzy definite logic program and/or annotated program of hypothesis H such that

$$B \cup H \models E$$

and for the minimal model f_m of $B \cup H$ we have for all e_1 and e_2 the following weak order preserving approximation condition:

$E(e_1) \leq E(e_2)$ implies $f_m(e_1) \leq f_m(e_2)$

and

$E(e_1) \geq E(e_2)$ implies $f_m(e_1) \geq f_m(e_2)$

with a possibility that the granulation of range of f_m is coarser than that of the range of the fuzzy set of examples E.

Comparison with some probabilistic and statistical induction methods.

Although our example is multirelational, it can be propositionalized (paying a big price on complexity) and after some preprocessing the function can be learned also by statistical methods.

A multivariate polynomial regression MPR from [19] gives the following polynome

$gm = +0.379169 * ca + 0.208618 * co + 0.53312 * ch - 0.0892687.$

In what follows, there are a decision and regression tree using DTREE from [4] and consequently rules learned from it.

dtree(gm) =
{(*ch*|0.45)
  <: {(*ca*|0.775)
    <: {(*ch*|0.05)
      <: {0 ~ 0[2]},
      >: {0.33 ~ 0[18]}},
    >: {(*co*|0.8)
      <: {(*ch*|0.15)
        <: {0.33 ~ 0[4]},
        >: {**0.594** ~ 0.132[5]}}, monotonicity violation, low accuracy
      >: {0.66 ~ 0[3]}}},
  >: {(*ca*|0.875)
    <: {(*ch*|0.65)
      <: {(*co*|0.55)
        <: {**0.44** ~ 0.155563[3]}, monotonicity violation, low accuracy
        >: {0.66 ~ 0[4]}},
      >: {0.66 ~ 0[17]}},
    >: {(*ch*|0.75)
      <: {0.66 ~ 0[2]},
      >: {1 ~ 0[6]}}}}

we see the function learned by tree violates the principle of order preserving approximation and the accuracy is lower than that of the GAP rule (having accuracy 100 % learned by ALEPH).

The DTREE rules on data from Example 1 look as follows:

$gm = 0 \leftarrow ca < 0.775 \wedge ch < 0.05;$
$gm = 0.33 \leftarrow ca < 0.775 \wedge ch > 0.05 \wedge ch < 0.45;$
$gm = 0.33 \leftarrow co < 0.8 \wedge ca > 0.775 \wedge ch < 0.15;$

$gm = \mathbf{0.44} \leftarrow co < 0.55 \wedge ca < 0.875 \wedge ch > 0.45 \wedge ch < 0.65;$
$gm = \mathbf{0.594} \leftarrow co < 0.8 \wedge ca > 0.775 \wedge ch > 0.15 \wedge ch < 0.45;$
$gm = 0.66 \leftarrow co > 0.8 \wedge ca > 0.775 \wedge ch < 0.45;$
$gm = 0.66 \leftarrow co > 0.55 \wedge ca < 0.875 \wedge ch > 0.45 \wedge ch < 0.65;$
$gm = 0.66 \leftarrow ca < 0.875 \wedge ch > 0.65;$
$gm = 0.66 \leftarrow ca > 0.875 \wedge ch > 0.45 \wedge ch < 0.75;$
$gm = 1 \leftarrow ca > 0.875 \wedge ch > 0.75.$

Notice, that rules use also negative dependence on values, this is violating requirement of definite logic program, which is necessary for monotonicity of the production operator and fixpoint existence, and this again for the cover relation (highlighted values out of range). DTREE rule with $gm = \mathbf{0.44}$ is in conflict with the ALEPH rule $R^6_{0.66}$ and DTREE rule with $gm = \mathbf{0.594}$ is in conflict with the ALEPH rules $R^2_{0.66}$ and $R^7_{0.33}$ and weak order preservation condition is violated.

Example 2 task gave similar results and (for different set of fuzzy examples) we obtained rules as following for example:

$biology(A) : 1 \leftarrow physics(A) : 2, chemistry(A) : 1, practice\_nursing(A) : 1$
$chemistry(A) : 1 \leftarrow history(A) : 2, math(A) : 2$
$physics(A) : 2 \leftarrow chemistry(A) : 2, biology(A) : 1$
$informatics(A) : 1 \leftarrow math(A) : 2, anatomy(A) : 4, nursing(A) : 2$

we see here we have recursion in the set of hypothesis.

Bayesian logic programs.

Project APRIL [2] – Application of Probabilistic Inductive Logic Programming addresses the problem of integrating probabilistic reasoning, first order logical representations and machine learning. This integration is one of the key open questions in artificial intelligence. An adequate answer to this open question is likely to result in new technologies that are applicable across a wide range of applications.

As already quoted [7] in the introduction: probability theory models uncertainty by assigning a probability to each of the states of the world that an agent considers possible.

Joint probability distribution over all possible worlds leads to exponentially many instantiations. Key insight, leading to Bayesian networks is the locality of influence. A certain limitation of Bayesian networks is its propositional character which leads often to exponential updating problem. Next step of development (behind the propositional horizon) are various object-attribute probabilistic models, especially that of Bayesian Logic Programming of K. Kersting and L. deRaedt which we start from.

In [13] they present a BLP model motivated by conditional probability distribution of height of person depending on height of parents and possibly some genetic information. An BILP learning based on maximum likelihood estimation and hill climbing algorithm is presented.

In [12] Kersting and DeRaedt proved (besides other results) that the classical two valued logic programming can be interpreted in the framework of BLP.

Note, that character of our application examples is totally different, in BLP there is a conditional probability dependency of different attributes (blood group, height,

genetical information), in our examples it is about classification, preference given by an expert, teacher, without any uncertainty in the model.

Using ideas of Kersting and De Raedt, it can be shown that annotated (and hence also fuzzy) LP can be also interpreted in the framework of BLP. Surprisingly (or not see for different character of applications) the quantitative part of GAP (FLP) is interpreted in attributes "truth value" understood qualitatively.

For instance, the annotation term $a(ca, ch, co) = gm$ from Example 1 by rule $R^2_{0.66}$ takes value $a(0.45, 0.6, 0) = 0.66$, this induces the BLP rule with same logical part and assigned cpd - conditional probability distribution being

$P(gm=1|ca=0.45, ch=0.6, co=0) = 0$
$P(gm=0.66|ca=0.45, ch=0.6, co=0) = 1$
$P(gm=0.33|ca=0.45, ch=0.6, co=0) = 0$
$P(gm=0|ca=0.45, ch=0.6, co=0) = 0$

that is that the probability is 1 only in the case of

$$P(gm=a(x,y,z)|ca=x, ch=y, co=z) = 1.$$

Again from $R^7_{0.33}$ we get

$$P(gm=0.33|ca=0, ch=0.1, co=0) = 1, P(gm=0.66|\ldots) = 0, P(gm=1|\ldots) = 0$$

and

$$P(gm=0|ca=0, ch=0.1, co=0) = 0$$

with combination rule of BLP being the maximum. Although in BLP there is no concept of model, we can prove certain soundness properties.

What is different, BLP does not allow recursion in background knowledge, whereas we have a fixpoint semantic based on a continuous operator in both FLP and GAP programs guaranteeing good behaviour of programs with negation (so far this aspect is not implemented in ILP).

5. RELATED RESEARCH

There are several papers on fuzzy ILP devoted to implementation of new search strategies and/or to combination with other tools.

To create a more suitable set of rules using ILP in [5] developed an algorithm called FS FOIL, that extends the original FOIL algorithm (described in [24]). While FOIL was developed to find Horn clauses, they modified it to be able to handle first order fuzzy predicates where cover compares confidence and support of fuzzy predicates.

In [26] another fuzzy variant of the ILP method FOIL is used for a crisp classification of "good arch' in civil engineering using vague linguistic hedges. Their system FCI uses min-max logic with Lukasiewicz implication and creates only crisp hypothesis.

Our system enables to describe more general dependencies (our function $a(x, y, z)$ from the annotation of gm is not expressible using min, max).

FCI search of hypothesis tries to cover positive examples with degree at least μ^+ and avoid covering of negative examples with degree below μ^- . We have tried to

model this in our approach. Running classical ILP with $E^+ = \{c : E(c) \geq \mu^+\}$ and $E^- = \{c : E(c) < \mu^-\}$ we get following results:

For $\mu^+ = 1$ and $\mu^- = 0.66$ we got

$gm(A, B) \leftarrow h(A, C), h(B, D), ch(C, D, 0.8)$.

with accuracy on positive examples 6/6 and 0/27 on negative.

For $\mu^+ = 1$ and $\mu^- = 0.33$ we got

$gm(A, B) \leftarrow o(A, C), o(B, D), co((C, D, 0.6)$.

again with full accuracy, and similarly for $\mu^+ = 0.66$ and $\mu^- = 0.33$ we got

$gm(A, B) \leftarrow h(A, C), h(B, D), ch(C, D, 0.1)$.

The only problem is the interpretation of these results in the framework of fuzzy logic in narrow sense. We cannot assign those rules truth value from the interval [0,1]. A solution would be to extend the set of truth values to a lattice containing the interval [0,1] as a chain and some subintervals. But this is out of the scope of this paper.

Another sort of learning under uncertainty are Neuro-Fuzzy Systems. The basic idea of combining fuzzy systems and neural networks is to design an architecture that uses a fuzzy system to represent knowledge in an interpretable manner and the learning ability of a neural network to optimize its parameters. The drawbacks of both of the individual approaches – the black box behavior of neural networks, and the problems of finding suitable membership values for fuzzy systems – could thus be avoided. A combination can constitute an interpretable model that is capable of learning and can use problem-specific prior knowledge. Therefore, neuro-fuzzy methods are especially suited for applications, where user interaction in model design or interpretation is desired (see [15], [18] and [22]).

A propositional fuzzy decision tree systems was presented also in [3]. Comparison with statistical methods of learning functions and rules were provided (after some preprocessing). Those do not act in a multirelational setting.

6. CONCLUSIONS

In this paper we have presented a new formulation of a fuzzy inductive logic programming task in the framework of fuzzy logic in narrow sense with formal definition of satisfaction and fixpoint semantics which gives a cover relation for theories (this emphasis on formal model is a feature which is distinguishing our approach from most of other approaches). We used a syntactical equivalence of fuzzy logic programs and a restricted class of generalised annotated programs. The induction is achieved via syntactical translation of our learning problem to multiple use of classical two valued induction, which is afterwards again glued together to a single annotated hypothesis interpreted as an annotation rule. Correctness of our method (translation) is based on the correctness of FLP in a sense, that negative part of the ILP task for $e \in E$ is $B \cup H \not\models e$) is replaced by a requirement that the minimal model of $B \cup H$ has some weak order preservation properties on examples (this is a feature which is specific for our approach and is not present in other systems). This is important for a learning of a graded monotone classification problem, where preservation of

the comparative notion present in example and background knowledge is important for it's intended meaning in the real world. A comparison with statistical methods showed they do not posses our weak order preservation property. Further study of these interrelations and more effective search strategies for monotone graded classification problem are a challenge. We will further experiment with connections to Bayesian networks and some applications in soft computing (see [1]) and the use of abduction in protocol security (see [11]).

ACKNOWLEDGEMENTS

The work has been supported by Slovak grant VEGA 1/0385/03 and COST 274.

(Received November 11, 2003.)

REFERENCES

[1] G. Andrejková and J. Jirásek: Neural network topologies and evolutionary design. Neural Network World *6* (2001), 547–560.

[2] Project APRIL – Applications of Probabilistic Inductive Logic Programming: http://www.informatik.uni-freiburg.de/~ml/april/

[3] B. Bouchon-Meunier and Ch. Marsala: Improvement of the interpretability of fuzzy rules constructed by means of fuzzy decision tree based systems. In: Abstracts of FSTA 2002, Liptovský Ján, Slovakia 2002.

[4] Decision and regression tree induction system DTREE: http://fuzzy.cs.uni-magdeburg.de/~borgelt/software.html#dtree

[5] M. Drobics, U. Bodenhofer, and W. Winiwarter: Interpretation of self-organizing maps with fuzzy rules. In: ICTAI 2000, IEEE.

[6] S. Džeroski and N. Lavrač: An introduction to inductive logic programming. In: Relational Data Mining (S. Džeroski and N. Lavrač, eds.) Springer–Verlag, Berlin 2001, pp. 48–73.

[7] L. Getoor et al: Learning probabilistic relational models. In: Relational Data Mining (S. Džeroski and N. Lavrač, eds.), Springer–Verlag, Berlin 2001, pp. 307–335.

[8] P. Hájek: Metamathematics of Fuzzy Logic. Kluwer, Dordrecht 1999.

[9] ILP system ALEPH: http://www.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/aleph.pl

[10] ILP system GOLEM: cs.cmu.edu/project/ai-repository-8/ai/areas/learning/systems/golem

[11] E. Jenčušová and J. Jirásek: Formal methods of security protocols. Tatra Mt. Math. Publ. *25* (2002), 1–10.

[12] K. Kersting and L. De Raedt: Towards combining Inductive Logic Programming with Bayesian Networks. In: Proc. ILP 2001 (C. Rouveirol and M. Sebagh, eds., Lecture Notes in Artificial Intelligence 2157), Springer–Verlag, Berlin 2001, pp. 118–131.

[13] K. Kersting and L. De Raedt: Adaptive Bayesian Logic Programs. In: Proc. ILP 2001 (C. Rouveirol and M. Sebagh, eds., Lecture Notes in Artificial Intelligence 2157), Springer–Verlag, Berlin 2001, pp. 104–117.

[14] M. Kifer and V. S. Subrahmanian: Theory of generalized anotated logic programming and its applications. J. Logic Programming *12* (1992), 335–367.

[15] A. Klose, A. Nürnberger, D. Nauck, and R. Kruse: Data Mining with Neuro-Fuzzy Models. In: Data Mining and Computational Intelligence (A. Kandel, H. Bunke, and M. Last, eds.), Physica–Verlag, Heidelberg 2001, pp. 1–36.

[16] S. Krajči, R. Lencses, and P. Vojtáš: A data model for annotated programs. In: ADBIS'02-Research Com. (Y. Manolopoulos and P. Návrat, eds.), Vydavatelstvo STU, Bratislava 2002, pp. 141–154.
[17] S. Krajči, R. Lencses, and P. Vojtáš: A comparison of fuzzy and annotated logic programming. Fuzzy Sets and Systems *144* (2004), 173–192.
[18] C.-T. Lin and C.-C. Lee: Neural Fuzzy Systems. A Neuro-Fuzzy Synergism to Intelligent Systems. Prentice Hall, New York 1996.
[19] Multivariate polynomial regression system MPR: http://fuzzy.cs.uni-magdeburg.de/~borgelt/software.html#regress
[20] S. Muggleton: Inductive logic programming. New Gen. Comp. *8* (1991), 295–318.
[21] S. Muggleton: Inverse entailment and Progol. New Gen. Comp. *13* (1995), 245–286.
[22] D. Nauck, F. Klawonn, and R. Kruse: Foundations of Neuro-Fuzzy Systems. Wiley, Chichester 1997.
[23] J. R. Quinlan: Learning logical definitions from relations. Mach. Learning *5* (1990), 239 266.
[24] J. R. Quinlan and R. M. Cameron-Jones: FOIL: A midterm report. In: Proc. 6th European Conference on Machine Learning (P. Brazdil, ed., Lecture Notes in Artificial Intelligence 667), Springer–Verlag, Berlin 1993, pp. 3-20.
[25] L. De Raedt and S. Džeroski: First order jk-clausal theories are PAC-learnable. Artificial Intelligence *70* (1994), 375–392.
[26] D. Shibata et al: An induction algorithm based on fuzzy logic programming. In: Proc. PAKDD'99 (Ning Zhong and Lizhu Zhou, eds., Lecture Notes in Computer Science 1574), Springer–Verlag, Berlin 1999, pp. 268–273.
[27] A. Srinavasan: The Aleph Manual. TRC Lab. Oxford Univ. 2000 at web.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/aleph.html
[28] P. Vojtáš: Fuzzy logic programming. Fuzzy Sets and Systems *124* (2001), 361–370.
[29] F. Železný: Learning functions from imperfect positive data. In: Proc. ILP 2001 (C. Rouveirol and M. Sebag, eds., Lecture Notes in Computer Science 2157), Springer–Verlag, Berlin 2001, pp. 248–259.

Peter Vojtáš, Institute of Informatics – P. J. Šafárik University, Košice, Slovak Republic, Institute of Computer Science – Academy of Sciences of the Czech Republic, Prague, Czech Republic and Mathematical Institute – Slovak Academy of Sciences, Bratislava. Slovak Republic.

Tomáš Horváth, Stanislav Krajči, and Rastislav Lencses, Institute of Informatics – P. J. Šafárik University, Košice. Slovak Republic.

e-mails: vojtas@kosice.upjs.sk; thorvath,krajci,lencses@science.upjs.sk