

Enriching Relational Learning with Fuzzy Predicates

Henri Prade, Gilles Richard, and Mathieu Serrurier

IRIT - Université Paul Sabatier
118 route de Narbonne 31062 Toulouse France
{henri.prade,serrurier}@irit.fr, grichard@ifi.edu.vn

Abstract. The interest of introducing fuzzy predicates when learning rules is twofold. When dealing with numerical data, it enables us to avoid arbitrary discretization. Moreover, it enlarges the expressive power of what is learned by considering different types of fuzzy rules, which may describe gradual behaviors of related attributes or uncertainty pervading conclusions. This paper describes different types of first-order fuzzy rules and a method for learning each type. Finally, we discuss the interest of each type of rules on a benchmark example.

Keywords: Inductive Logic Programming, relational learning, fuzzy rule, confidence degree

1 Introduction

Inductive Logic Programming (ILP) [9] provides a general framework for learning classical first-order logic rules, for which reasonably efficient algorithms have been developed (Progol [6], FOIL [13],...). Relational learning can be presented as a subfield of ILP that concerns the induction process on relational databases compiled in first-order logic. In this scope, we have only to consider function-free Horn clauses. But first-order logic cannot directly handle rules with exceptions, which are common in practice. This has been a motivation for introducing probabilities in ILP [7]. In fact, probabilities, implicitly appear in the FOIL control procedure. Indeed, during the gain computation, the value associated to a rule can be viewed as a confidence degree expressed in terms of “domain probabilities”. Such probabilities, together with “world probabilities”, are the basic notions of Halpern’s first-order probabilistic logic [4]. Domain probabilities are used to capture statistical information for a fixed first-order logic interpretation. These probabilities are obtained by applying a probability measure to the set of valuations making rules true in the interpretation. So, there is no longer any genuine quantifier in a rule when the probability to encounter exceptions is non-zero.

One of the difficulties of the induction of rules from examples is to manage real numbers and imprecision when attributes are non-binary. Classical method for handling real-valued attributes is to turn them into (symbolic) qualitative labels by discretization. Fuzzy sets are known to provide a gradual interface

with numerical data, by escaping the problem of sharp transitions between categories. In the propositional framework, confidence degrees have been integrated in learning methods, together with the handling of fuzzy properties. At least three main trends of works can be distinguished w.r.t. this latter concern. First, neuro-fuzzy learning techniques have been developed for tuning fuzzy membership functions in fuzzy rules; see [8] for a survey. The fuzzy rules, which are produced in that way, are used for functions approximation in automatic control problems. Another research line has been investigated with a greater concern for the descriptive power of the fuzzy rules from the user's point of view, by extending Quinlan's [12] ID3 algorithm to fuzzy decision trees, involving a fuzzy descriptions of classes and making use of entropy measures (extended to fuzzy sets) for building the fuzzy rules; see [1] for a survey. More recently, the use of fuzzy membership functions has been advocated by several researcher for providing association rules in data mining with a better representation power, e.g. [5].

Presently, the majority of the methods for learning fuzzy rules are propositional. A version of FOIL that handles membership degrees has already been developed [15] but the rules induced still keep a classical meaning. In this paper, we propose a method for inducing first-order rules that may include fuzzy predicates. We first explain how a classical database is read in terms of fuzzy predicates, and we further discussed different types of fuzzy rules recently introduced in a learning perspective [11]. For each type of rules, the FOIL algorithm is adapted by defining the corresponding confidence degree. The paper is organized as follows. Section 2 provides a brief background on ILP. Section 3 presents different types of fuzzy rules and the fuzzy database. Section 4 describes our algorithm and section 5 illustrates the approach on an toy example and a benchmark.

2 Background

We first briefly recall the standard definitions and notations. Given a first-order language \mathcal{L} with a set of variables Var , we build the set of terms $Term$, atoms $Atom$ and formulas as usual. The set of ground terms is the Herbrand universe \mathcal{H} and the set of ground atoms or facts is the Herbrand base $\mathcal{B} \subset Atom$. A *literal* l is just an atom a (positive literal) or its negation $\neg a$ (negative literal). A (resp. ground) substitution σ is an application from Var to (resp. \mathcal{H}) $Term$ with inductive extension to $Atom$. We denote $Subst$ the set of ground substitutions. A *clause* is a finite disjunction of literals $l_1 \vee \dots \vee l_n$ also denoted $\{l_1, \dots, l_n\}$. A Horn clause is a clause with at most one positive literal. A Herbrand interpretation I is just a subset of \mathcal{B} : I is the set of true ground atomic formulas and its complementary denotes the set of false ground atomic formulas. Let us denote $\mathcal{I} = 2^{\mathcal{B}}$, the power set of \mathcal{B} i.e. the set of all Herbrand interpretations. We can now proceed with the notion of logical consequence.

Definition 1. *Given A an atomic formula, $I, \sigma \models A$ means that $\sigma(A) \in I$. As usual, the extension to general formulas F uses compositionality.*

$I \models F$ means $\forall \sigma, I, \sigma \models F$ (we say I is a model of F).
 $\models F$ means $\forall I \in \mathcal{I}, I \models F$.
 $F \models G$ means that all models of F are models of G .

Stated in the general context of first-order logic, the task of *induction* is to find a set of formulas H such that:

$$B \cup H \models E \quad (1)$$

given a background theory B and a set of observations E (training set), where E, B and H here denote sets of clauses. A set of formulas is here, as usual, considered as the conjunction of its elements.

Of course, one may add two natural restrictions:

- $B \not\models E$ since, in such a case, H would not be necessary to explain E .
- $B \cup H \not\models \perp$: this means $B \cup H$ is a consistent theory.

In ILP, there are two ways for describing examples. The first describes the set of positives examples E^+ and the set of negative examples E^- . The other describes only positive examples in E and make the closed world assumption. It is this hypothesis that we will uses along this paper. Each element of E is called an example and we call a counter-example a fact on the target concept which is not in E . In the setting of relational databases, inductive logic programming is often restricted to Horn clauses and function-free formulas, E is just a set of ground facts. Moreover, the set E itself satisfies the previous requirement but it is generally not considered as an acceptable solution since it has no predictive ability. Usually, rules extraction fits with the idea of providing a compression of the information content of E .

There are two general types of algorithms, *top down* and *bottom up* algorithms. *Top down* ones start from the most general clause and specialize it step by step. *Bottom up* procedures start from a fact and generalize it. In our case, we will use the FOIL algorithm [13] which is a *top down* process. The goal of FOIL is to produce rules until all the examples are covered. Rules with conclusion part C , the target predicate, are found in the following way:

1. take $A \rightarrow C$ as the most general clause with $A = \top$
2. choose the literal l such as the clause $l \wedge A \rightarrow C$ maximizes the gain function
3. $A = l \wedge A$
4. if $\text{confidence}(A \rightarrow C) < \text{threshold}$ goto 2
5. return $A \rightarrow C$

The gain function is computed by the formula:

$$\text{gain}(l \wedge A \rightarrow C, A \rightarrow C) = n * (\log_2(\text{cf}(l \wedge A \rightarrow C)) - \log_2(\text{cf}(A \rightarrow C)))$$

where n is the number of distinct examples covered by $l \wedge A \rightarrow C$. Given a Horn clause $A \rightarrow C$, the confidence $\text{cf}(A \rightarrow C) = \frac{P(A \wedge C)}{P(A)}$. Confidence degrees are computed according to the definition of domain probabilities [4]. ILP data are

supposed to describe one interpretation under Closed World Assumption. We call I_{ILP} this interpretation. So, given a fact f :

$$I_{ILP} \models f \text{ iff } B \wedge E \models f.$$

The domain \mathcal{H} is the Herbrand domain described by B and E . We take P as a uniform probability on \mathcal{H} . So we deduce that the confidence in a clause $A \rightarrow C$, with \vec{t} as vector on the n free variables, is:

$$cf(A(\vec{t}) \rightarrow C(\vec{t}))_{I_{ILP}} = \frac{|\{\vec{x} \in \mathcal{H}^n \mid I_{ILP} \models \sigma[\vec{t}/\vec{x}](A(\vec{t}) \wedge C(\vec{t}))\}|}{|\{\vec{x} \in \mathcal{H}^n \mid I_{ILP} \models \sigma[\vec{t}/\vec{x}](A(\vec{t}))\}|} \quad (2)$$

where $||$ denotes cardinality. Another possible definition of a confidence degree might be taken here as the proportion of the number of positive examples covered by the rule w.r.t. the number of total examples (positive and negative) covered by the rule. This confidence degree would represent the probability that a fact deduced from the rule is true. But this definition would not take into account the number of situations covered in the condition part of the rule, which is not always the total number of examples covered since we are in a first-order setting.

In ILP, the goal is to learn a concept represented by a predicate. E is the set of all facts pertaining to the target predicate. B is the set of facts pertaining to predicates other than the target one. So the learned rules are (in the non-recursive case) composed by predicates that appear in B for the condition part and by the target predicate in the consequence part.

3 Induction in Fuzzy Database

3.1 Fuzzy Databases and Fuzzy Rules

We consider a first-order logic database K with fuzzy predicates (e.g., heavy, old ...) as a set of positive facts labeled by real numbers in $[0, 1]$. For instance in Section 5 we shall deal with a database containing facts such as $(weight(a, heavy), 0.9)$ which means that the car a is very representative of heavy cars. Thus, K is made of pairs of the form $(A(\vec{x}), \mu(A(\vec{x})))$ for $\vec{x} \in \mathcal{H}^n$, where $A(\vec{x})$ is a fact, and $\mu(A(\vec{x}))$ is the satisfaction degree associated with the fuzzy property A for \vec{x} .

There exist at least two reasons for introducing fuzzy predicates in universally quantified rules. This may be for making them either more flexible or more expressive. Indeed, a fuzzy predicate can be viewed as a family of ordinary predicates whose characteristic functions are the level cut functions μ_{F_α} associated to the fuzzy set membership function μ_F , namely $\mu_{F_\alpha}(\vec{x}) = 1$ iff $\mu(F(\vec{x})) \geq \alpha$ and $\mu_{F_\alpha}(\vec{x}) = 0$ otherwise. Thus a rule “ $A(\vec{t}) \rightarrow C(\vec{t})$ ” is naturally associated with the crisp rules “ $A_\alpha(\vec{t}) \rightarrow C_\beta(\vec{t})$ ”. Note that, if $A_\beta(\vec{t})$ holds then $A_\alpha(\vec{t})$ also holds for $\alpha \leq \beta$. So we may only consider the crisp approximations “ $A_\alpha(\vec{t}) \rightarrow C_\alpha(\vec{t})$ ”.

Then, if we are concerned with *flexibility*, a possible understanding of the fuzzy rule “ $A(\vec{t}) \rightarrow C(\vec{t})$ ” can be

$$\forall \vec{x}, \exists \alpha \ A_{\alpha}(\vec{x}) \rightarrow C_{\alpha}(\vec{x}), \tag{3}$$

i.e. there exists a crisp understanding of the fuzzy rule which covers each example (but it is not necessary the same for each example since α depends on \vec{x}). This is a kind of rule yet considered in [10]. By flexible rules, we mean here rules which are robust since their predicates can be adapted to borderline situations.

If we are concerned with *expressivity*, we may look for fuzzy rules such that the rule holds for *each* of its level cut counterpart. This means that we have

$$\forall \vec{x}, \forall \alpha \ A_{\alpha}(\vec{x}) \rightarrow C_{\alpha}(\vec{x}). \tag{4}$$

This is clearly more restrictive than (3) since the fuzzy rule is equivalent to a set of ordinary rules with nested predicates and summarizes it into a unique fuzzy rule. In fact (4) is nothing but a gradual rule [3] expressing “The more \vec{x} satisfies A, the more \vec{x} satisfies C” (since they are modeled by a constraint of the form $\mu(A(\vec{x})) \geq \mu(C(\vec{x}))$).

Gradual rules are one of the four basic kinds of fuzzy rules [3]. Two of them, namely gradual rules and certainty rules, are based on implication connectives and express constraints on the possible models of the world. The two other types, named possibility rules and antigradual rules, rather express that some values are guaranteed to be possible (i.e. that they exist in the base of examples). For instance, let us take possibility rules of the form “The more \vec{x} is A, the more all the interpretations which makes C true (truth becomes a matter of degree when C is fuzzy) are guaranteed to be possible”. This means in practice that “The more \vec{x} is A, the more there are examples for any possible interpretation of C”. Note that this rule cannot have any “classical” counter-example since we are interested in the distribution of the membership degrees in the database.

In the following, we only consider gradual and certainty rules. Certainty rules contrast with possibility rules, and express that “the more \vec{x} is A, the more certain \vec{x} is C”. Let us first consider the case where “A” is a fuzzy predicate and “C” is an ordinary predicate. This expresses that “the more \vec{x} is A, i.e. the greater α such that $A(\vec{x}) \geq \alpha$, the smaller the number of exceptions of the rule $A_{\alpha}(\vec{t}) \rightarrow C(\vec{t})$ ”. Indeed when α decreases, the number of exceptions cannot but increase since the scope of A_{α} is then enlarged. When C is also a fuzzy predicate, in order to preserve this understanding of the rule, we are led to look for rules of the form

$$\forall \vec{x}, \forall \alpha \ A_{\alpha}(\vec{x}) \rightarrow C_{1-\alpha}(\vec{x}), \tag{5}$$

since when α increases $C_{1-\alpha}$ cover more cases.

3.2 Application to ILP

It is well known that algorithms for learning rules have difficulties for handling real-valued attributes. In fact, numerical values may lead to an infinite hypothesis

space. In relational learning, this problem is deeper since the hypothesis space is already large. The difficulties grow up when real numbers appears in the concept we want to learn. Real numbers are essentially treated in two way: either by introducing constraints or by discretization.

Introducing constraints in first order logic consists in the use of several operators such as inequalities or mathematical functions (average, ...) [14]. Rules induced by this method may suffer of a lack of expressivity and generality. Furthermore, algorithms dealing with constraints go out of the scope of the standard resolution process in first-order logic.

The second way is to use discretization and clusterization for transforming continuous information into qualitative information. Then, information can be directly treated in the classical logic setting. This method is the most currently used since it allows to cope with numerical values and to improves the readability. Since the clusters are usually defined in an arbitrary way before the induction process, the rules which are produced depend on the quality of the clusters. These clusters are often represented by predicates having an imprecise meaning. For example in the auto-mpg data in UCI, the mpg (city fuel consumption in miles per gallons) value can be represented in terms of the predicates “low consumption”, “medium consumption” and “ high consumption”. In this case, fuzzy labels, represented by fuzzy sets, are more appropriate for describing the mpg values since they avoid arbitrary thresholds between low and medium (see Fig. 1 for description).

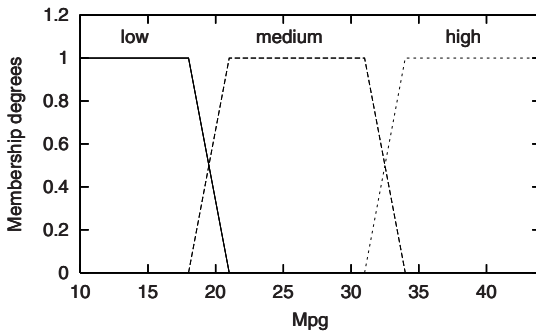


Fig. 1. Fuzzy cluster of mpg

Finally, using fuzzy predicates allows to relax the rigidity of crisp clustering and keeps the readability of the induced rules. Moreover, the different types of fuzzy rules we have described allow a better description and provide new types of summarization of the data. *Flexible rules* can be viewed as a fuzzy adaptation of a crisp rules in the sense that there exists a reading of the fuzzy predicates, corresponding to a high level-cut of its fuzzy representation, which leads to a meaningful rule w.r.t. data. *Gradual rules* and *Certainty rules* are new type of rules, which describe new implicative properties of the data. In the case of

ILP, the goal is to find an hypothesis which is sound and complete with respect to the examples. The hypothesis is sound if it does not cover facts on the target concept which are false in the interpretation defined by the background and the examples. It is complete if it covers all facts on the target concept which are true in the latter interpretation. In the case of fuzzy ILP, the definition of an example covered by a rule will depend on the type of the fuzzy rule and of the membership degrees of facts validating the rules.

4 Algorithm

In the FOIL algorithm, the guidelines for the process are: the confidence degree, the halting condition and the number of distinct examples covered by the rule. We consider that an example is covered by a fuzzy rule if it is itself covered by the classical counter-part of the rule. So we describe these guidelines for each kind of rules (see [10] [11] for details).

Flexible Rules. This first type of meaning for a fuzzy rule “ $A(\vec{t}) \rightarrow C(\vec{t})$ ” is close to the one of a classical rule. Of course, we are now expecting that the satisfaction degrees of $A(\vec{x})$ and $C(\vec{x})$ are as high as possible. So we can introduce classical interpretations associated with each α -cut.

Definition 2. An α -interpretation I_α , given a fact f , is defined by:

$$I_\alpha \models f \text{ iff } B \wedge E \models f \text{ and } \mu(f) \geq \alpha$$

In this type of interpretations, only facts having a satisfaction degree greater than α are true. Now we have to compute the confidence degree of the rule in the classical way (using (2)) for each α -interpretation. According to the intended meaning of the fuzzy rule, we must favor the confidence degrees of the rule computed in high α -interpretations. Indeed, we prefer the examples be covered with a high degree of satisfaction. The following definition, which is an adaptation in term of first-order logic of the one proposed by [2], takes this into account:

$$cf_{flex}(A(\vec{t}) \rightarrow C(\vec{t})) = \sum_{\alpha_i} (\alpha_i - \alpha_{i+1}) * cf(A(\vec{t}) \rightarrow C(\vec{t}))_{I_{\alpha_i}}$$

where $\alpha_1 = 1, \dots, \alpha_t = 0$ is the decreasing list of the satisfaction degrees that appear in the database. This confidence degree corresponds to the discretization of a Choquet integral of the confidence degrees on α -interpretations. We deduce the number of distinct examples covered:

$$n_{flex}(A(\vec{t}) \rightarrow C(\vec{t})) = \sum_{\alpha_i} (\alpha_i - \alpha_{i+1}) * |\{\vec{x}_1 \in \mathcal{H}^q, \exists \vec{x}_2 \in \mathcal{H}^r \mid I_{\alpha_i} \models \sigma[\vec{t}_1, \vec{t}_2 / \vec{x}_1, \vec{x}_2](A \wedge C)\}|$$

Gradual Rules. In this case, the values of the satisfaction degrees are only useful for comparing satisfaction degrees in condition and conclusion parts. So, we do not privilege the confidence degree in high α -interpretation as previously.

$$cf_{grad}(A(\vec{t}) \rightarrow C(\vec{t})) = \frac{|\{\vec{x} \in \mathcal{H}^n \mid I_{ILP} \models \sigma[\vec{t} / \vec{x}](A \wedge C), \mu(\sigma[\vec{t} / \vec{x}]C) \geq \mu(\sigma[\vec{t} / \vec{x}]A)\}|}{|\{\vec{x} \in \mathcal{H}^n \mid I_{ILP} \models \sigma[\vec{t} / \vec{x}](A)\}|}$$

When the valuation of the condition part of the rule is a conjunction of grounds literals, the satisfaction degree of this conjunction is the minimum of the degree of each literal. We deduce the number of distinct examples covered:

$$n_{grad}(A(\vec{t}) \rightarrow C(\vec{t})) = |\{\vec{x}_1 \in \mathcal{H}^q, \exists \vec{x}_2 \in \mathcal{H}^r \mid I_{ILP} \models \sigma[\vec{t}_1, \vec{t}_2/\vec{x}_1, \vec{x}_2](A \wedge C, \mu(\sigma[\vec{t}_1/\vec{x}_1]C) \geq \mu(\sigma[\vec{t}_1, \vec{t}_2/\vec{x}_1, \vec{x}_2]A))\}|$$

Type 1 Certainty Rules. The meaning of the fuzzy rule “ $A(\vec{t}) \rightarrow C(\vec{t})$ ” is then “the more \vec{x} is A, the more certain \vec{x} is C”. For these rules we are not interested in the satisfaction degrees of the consequence parts. This type of rule will be referred to as type 1 certainty rules in the following. The α -cut for these rules correspond to the following type of classical interpretation:

Definition 3. An α -certainty interpretation, given a fact f , is defined by:

$$I_{\alpha-cert} \models f \text{ iff } (B \models f \text{ and } \mu(f) \geq \alpha) \text{ or } E \models f$$

With this kind of rules, confidence degrees are expected to be high for high α -certainty interpretation. The idea is that we can be more permissive with respect to exceptions for the classical counterparts of the rule “ $A(\vec{t}) \rightarrow C(\vec{t})$ ” corresponding small values of α . So, we are led to use the following Choquet integral.

$$cf_{cert1}(A(\vec{t}) \rightarrow C(\vec{t})) = \sum_{\alpha_i}^t (\alpha_i - \alpha_{i+1}) * cf(A(\vec{t}) \rightarrow C(\vec{t}))_{I_{\alpha_i-cert}}$$

We deduce the number of distinct examples covered:

$$n_{cert1}(A(\vec{t}) \rightarrow C(\vec{t})) = \sum_{\alpha_i} (\alpha_i - \alpha_{i+1}) * |\{\vec{x}_1 \in \mathcal{H}^q, \exists \vec{x}_2 \in \mathcal{H}^r \mid I_{\alpha-cert} \models \sigma[\vec{t}_1, \vec{t}_2/\vec{x}_1, \vec{x}_2](A \wedge C)\}|$$

Type 2 Certainty Rules. The above definition is modified in the following way for taking care of the satisfaction degree of the consequence of the rules. This type of rule will be referred to as type 2 certainty rules in the following

$$cf_{cert2}(A(\vec{t}) \rightarrow C(\vec{t})) = \frac{|\{\vec{x} \in \mathcal{H}^n \mid I_{ILP} \models \sigma[\vec{t}/\vec{x}](A \wedge C), \mu(\sigma[\vec{t}/\vec{x}]C) > 1 - \mu(\sigma[\vec{t}/\vec{x}]A)\}|}{|\{\vec{x} \in \mathcal{H}^n \mid I_{ILP} \models \sigma[\vec{t}/\vec{x}](A)\}|}$$

We deduce the number of distinct examples covered:

$$n_{cert2}(A(\vec{t}) \rightarrow C(\vec{t})) = |\{\vec{x}_1 \in \mathcal{H}^q, \exists \vec{x}_2 \in \mathcal{H}^r \mid I_{ILP} \models \sigma[\vec{t}_1, \vec{t}_2/\vec{x}_1, \vec{x}_2](A \wedge C), \mu(\sigma[\vec{t}_1/\vec{x}_1]C) > 1 - \mu(\sigma[\vec{t}_1, \vec{t}_2/\vec{x}_1, \vec{x}_2]A)\}|$$

Thus, we can use the FOIL algorithm for inducing various kinds of first-order fuzzy rules by adapting confidence degree and cardinality with the type of rules we want to learn.

5 Results

5.1 Illustrative Example

Let us consider a database that describes 21 houses in a town. First we have some fuzzy relational predicates such as $(close(x, y), \alpha)$ which means that the

house x is close to the house y with a membership degree α , or ($know(x, y), \alpha'$) which means that the owner of house x knows the owner of the house y at a degree α' (from 0 for unknown to 1 for friends). The houses are also described with some nearly propositional fuzzy predicates such as $price(x, expensive)$ or $size(x, small)$.

So, in this context, we can find fuzzy rules of each type with a good confidence degree. For example, we find the *flexible* rule

$$close(x, y), price(y, expensive) \rightarrow price(x, expensive),$$

with 0.81 of confidence degree, because we can reasonably expect that a house which is close to an expensive one, is expensive as well (since expensive houses are often located in the same area). A typical *gradual* rule is

$$size(x, large) \rightarrow price(x, expensive),$$

i.e. “the larger the house, the more expensive”, which describes the fact that price grows up with size. Its confidence degree is 0.80. A good example of *certainty rules* is

$$close(x, y) \rightarrow know(x, y)$$

with 0.95 of confidence degree if the rule is viewed as type 1 certainty rule and 0.88 of confidence degree if the rule is viewed as type 2 certainty rule. This rule means “the closer the houses, the more we are sure that the owners know together”. The fact that owners of very close houses have a high probability to know each other is realistic. This probability can decrease when the distance between the houses grow up. As ending remark, we may observe that all these rules are obviously subject to exceptions and, despite their interest, they cannot be obtained in any way by a classical ILP machine.

5.2 Benchmark

As a benchmark, we use the “auto-mpg” database from UCI¹. This database is constituted with informations about cars and the concept we want to learn is the city-cycle fuel consumption in miles per gallon. There are 398 instances of cars described by 9 attributes of which 5 are continuous, including the concept to be learn. This database can be represented in propositional logic but is sufficient to illustrate the interest of the approach. First, the database has been “discretized” with fuzzy sets. Moreover, we also built three crisp discretizations corresponding to i) the crisp partition of the attribute domain which is the closest to the fuzzy partition, ii) the support of the fuzzy sets, iii) the core of the fuzzy sets. Then, we learn the class of city-cycle fuel consumption according to the crisp and fuzzy set for all the types of fuzzy rules.

¹ <http://www.ics.uci.edu/mllearn/MLRepository.html>

Types of rules	nbr of rules	coverage	avg cf
classical rules	8	0.77	0.84
classical rules with the core	11	0.59	0.87
classical rules with the support	10	0.85	0.80
flexible rules	3	0.51	0.84
gradual rules	4	0.47	0.91
type 1 certainty rules	2	0.62	0.75
type 2 certainty rules	2	0.59	0.76

Here are examples of rule induced by the algorithm for each type of them (classical rules are one induced on the discretisation corresponding to the crisp partition of the attribute domain which is the closest to the fuzzy partition).

Classical rules

$$cylinders(A, 8) \rightarrow mpg(A, low)$$

flexible rules

$$displacement(A, low), weight(A, medium) \rightarrow mpg(A, medium)$$

gradual rules

$$weight(A, high) \rightarrow mpg(A, low)$$

type 1 certainty rules

$$cylinders(A, 6), weight(A, high) \rightarrow mpg(A, low)$$

type 2 certainty rules

$$cylinders(A, 6), weight(A, high), origin(A, 1), acceleration(A, low), horsepower(A, low) \rightarrow mpg(A, low)$$

As expected, the coverage score of classical rules is between the score of classical rules with the core of fuzzy sets and classical rules with the support of fuzzy sets. It is due to the fact that, with the core of the fuzzy rules, the examples that are in the boundary of the crisp classes are not treated. On the contrary, with the support of fuzzy sets, the example of that are in the boundary of the crisp classes can belong to two classes. The smaller score of fuzzy rules w.r.t. coverage is due to the fact that fuzzy rules are harder to find than classical ones. This result is expected because fuzzy rules are more constrained since they take into account the membership degree of the valuations of each predicate. In fact, confidence degrees of classical rules do not rely on the distance of the data to the boundaries of the discretized sets. For example, let us consider a classical rule F with a good confidence degree, and its fuzzy flexible counterpart F' . If many example of F are borderline w.r.t. fuzzy sets, the confidence degree of F' will be lower than the one of F . On the contrary, if many counter-examples of F are borderline, the confidence degree of F' will be greater than the one of F . So, the confidence degree of fuzzy flexible counterpart of a rule is a good indicator of the robustness of the classical rule w.r.t. small variation of the boundaries of the sets.

Type 1 certainty rules focus on membership degrees of the conditional parts of the rules. Gradual and certainty rules show how conditions and conclusions parts evolve together. These rules have a meaning far from the classical one and the rules that we find have not necessarily a crisp counterpart or approximation. The fuzzy rules that handle certainty tend to favor the non-fuzzy predicates in

condition part because they leave more freedom with respect to the satisfaction degree of covered examples. Note that some rules could be described in propositional logic, but here the instantiations are automatically generated by the algorithm. As shown in some rules, the algorithm can mix fuzzy predicates and non-fuzzy predicates.

6 Conclusion

In this paper, we have provided a formal framework and a procedure for dealing with fuzzy predicates and learning fuzzy first-order rules of different kinds in the case of relational databases. Since the confidence degree computation is a weighted version of FOIL's one, it is easy to deduce that the complexity of our algorithm is the same as the FOIL's one. The definition of confidence degrees for each kind of rules allows us to take into account the fuzzy predicates in the algorithms that use confidence degrees for guiding the learning process. It is obvious that using fuzzy predicates for managing real-valued data instead of using crisp discretization or constraint-based induction is a good compromise between the readability of the rules and the flexibility of the discretization. Moreover, fuzzy predicates allow to extract new kinds of relations.

Through the example, we see that fuzzy rules are often too constrained for covering all the examples of the target concept, but they convey information on the robustness of the rules w.r.t. borderline examples. So, it can be useful to learn fuzzy rules together with classical ones.

In this paper, we focus on the search of different kinds of fuzzy rules and the definition of confidence degrees associated to each of them. In further works, it will be interesting to show how much fuzzy discretization is efficient in a learning point of view. More generally, a formal definition of ILP that handles all the types of rules must be defined. In this context, automatic deduction mechanisms may be developed for testing the efficiency of fuzzy rules in terms of classification.

References

1. B. Bouchon-Meunier and C. Marsala. Learning fuzzy decision rules, in. *Fuzzy Sets in Approximate Reasoning and Information Systems*,(J.C. Bezdek, D. Dubois, H. Prade, eds.), The Handbooks of Fuzzy Sets Series. Kluwer Academic Publishers, 1999, 279-304.
2. M. Delgado, D. Sanchez, and M.A. Vila. Fuzzy cardinality based evaluation of quantified sentences. *Inter. J. of Approximate Reasoning*, pages 23:23–66, 2000.
3. D. Dubois and H. Prade. What are fuzzy rules and how to use them. *Fuzzy Sets and Systems*, 84(2):169–189, 1996.
4. J. Halpern. An analysis of first-order logics of probability. *Artificial Intelligence*, 46:310–355, 1990.
5. E. Hüllermeier. Implication-based fuzzy association rules. In L. De Raedt and A. Siebes, editors, *Proc. PKDD-01, 5th Conf. on Principles and Practice of Knowledge Discovery in Databases*, number 2168 in LNAI, pages 241–252, 2001.

6. S.H. Muggleton. Inverse entailment and Progol. *New Generation Computing*, 13:245–286, 1995.
7. S.H. Muggleton. Learning stochastic logic programs. *Electronic Transactions in Artificial Intelligence*, 5(041), 2000.
8. D. Nauck and R. Kruse. Neuro-fuzzy methods in fuzzy rule generation, in. *Fuzzy Sets in Approximate Reasoning and Information Systems*, (J.C. Bezdek, D. Dubois, H. Prade, eds.), The Handbooks of Fuzzy Sets Series. Kluwer Acad. Pub., 1999, 305-334.
9. S-H Nienhuys-Cheng and R. de Wolf. *Foundations of Inductive Logic Programming*. Number 1228 in LNAI series. Springer, 1997.
10. H. Prade, G. Richard, and M. Serrurier. Learning first order fuzzy rules. In *Proc. of 10 th Int. Fuzzy Systems Association (IFSA-03)*, Istanbul, 2003.
11. H. Prade, G. Richard, and M. Serrurier. On the induction of different kinds of first-order fuzzy rules. In *Proc 7 th European Conference on Symbolic and Qualitative Approaches to Reasoning with Uncertainty (ECSQARU-03)*, Aalborg, 2003.
12. J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.
13. J. R. Quinlan. Learning logical definitions from relations. *Machine Learning*, 5:239–266, 1990.
14. J. R. Quinlan. Learning with continuous classes. In *Proc. Artificial Intelligence (AI'92)*, 343–348, Singapore, 1992.
15. D. Shibata, N. Inuzuka, S. Kato, T. Matsui and H. Itoh In Ning Zhong and Lizhu Zhou, editors, *Proceedings of The Third Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD-99): Methodologies for Knowledge Discovery and Data Mining*, 268–273, LNAI 1574, Beijing, China, April 1999.