Apprentissage de Concept à partir d'Exemples (très) Ambigus

Dominique Bouthinon[1], Henry Soldano[1], Véronique Ventos[2]

[1] L.I.P.N, UMR-CNRS 7030, Université Paris-Nord,
93430 Villetaneuse, France
[2] LRI, UMR-CNRS 8623, Université Paris-Sud,
91405 Orsay, France
{dominique.bouthinon, henry.soldano}@lipn.univ-paris13.fr,
ventos@lri.fr

Résumé : Dans cet article nous explorons l'incomplétude des données dans le cadre de l'apprentissage de concepts propositionnels. Nous suivons l'idée de H. Hirsh qui étend le paradigme de l'espace des versions : dans cette extension une hypothèse doit être compatible (dans un sens à définir au cas par cas) avec toutes les informations relatives aux exemples. Nous proposons une représentation de ces informations qui rend non seulement compte de situations où les données sont manquntes mais aussi de situations plus générales d'ambiguité dans lesquelles l'exemple est caché au sein d'un ensemble d'instances virtuelles. Nous présentons un nouvel algorithme, LEa, qui apprend un concept DNF (monotone) existentiel à partir d'un ensemble d'exemples ambigus. Nous comparons LEa à *J48* et *Naive Bayes* sur des problèmes usuels rendus incomplets à divers degrés.

Mots-clés : Apprentissage de concepts propositionnels, Ambiguité, Données incomplètes.

1 Introduction

We investigate here the effect of incompleteness in propositional concept learning from examples and in its first order extension : the learning from interpretations setting introduced by DeRaedt (1997). Concept learning from examples relies on a membership relation between hypotheses and examples denoted as *cover* and such that to be a solution an hypothesis has to *cover* positive examples and should *not cover* negative examples of the target concept. This set of solutions, inheriting its partial order from the Hypothesis language, is called the Version Space (Mitchell, 1982) of the learning problem. This definition of concept learning relies on a complete description of the examples. In Hirsh (1994), the author informally proposes to extend the notion of solution in order to use any piece of information concerning the current example set. The definition of concept learning problems has then to be modified : a hypothesis has now to be in some sense *compatible* with such pieces of information. We consider the general case, where an example is *ambiguous* in the following sense : the example is represented as a set of *possible* complete examples further denoted as *possibilities*. The idea here is that the true example, corresponding to an observation, is exactly one of these possibilities which is so *hidden* within the ambiguous example. To take into account this ambiguity we use two relations *compatible*$^+$ and *compatible*$^-$: a hypothesis h is then *compatible*$^+$ with a positive ambiguous example e if h covers at least one possibility of e, while h is *compatible*$^-$ with a negative ambiguous example e whether there is at least one possibility of e which is not covered by h.

As an illustration, consider a world of birds from which we want to learn the concept *fly*. Any bird is described with the atoms $P = \{$*red*, *green*, *migratory*, *not_migratory*, *light*, *not_light*$\}$ and a bird is either *red* or *green*, either *migratory* or *not_migratory*, and either *light* or *not_light*. Now suppose the only thing we know about a given bird is that it is *red*. Then it is extensionnaly represented as the ambiguous example $e = \{\{$*red, migratory, light*$\}$, $\{$*red, migratory, not_light*$\}$, $\{$*red, not_migratory, light*$\}$,$\{$*red, not_migratory, not_light*$\}\}$ containing 4 valid possibilities. Here a hypothesis h covers a possibility p if h is included in p. First assume that e is a positive ambiguous example, then $h = \{$*migratory*$\}$ is *compatible*$^+$ with e since h covers $\{$*red, migratory, light*$\}$. Assume now that e is a negative ambiguous example then h is *compatible*$^-$ with e since h does not covers $\{$*red, not_migratory, light*$\}$.

An ambiguous example can also be intentionally described as a clausal theory that defines constraints on the universe of instances, together with a set of facts. This is the approach of *abductive concept learning* (Kakas & Riguzzi, 2000) in which hypothesis are clauses and the coverage relation is replaced by a procedure of *abductive entailment* playing the same role as our *compatibility* relation. Unfortunately the cost of the abductive entailment test applied to each example may become prohibitive whenever we face strong uncertainty. In contrast, the extensional approach presented here uses a simple subsumption test, but strong ambiguity can result in a huge set of possibilities and thus in a prohibitive cost. Our proposal is a rule learning algorithm that returns one of the simplest elements of the Version Space. It uses a compact *multi-table representation* (Alphonse, 2004) of ambiguous examples that can lead to an exponential gain in the representation size. Furthermore, we will see that only *maximal* possibilities (following the inclusion order on interpretations) have to be considered when considering a positive example, whereas only *minimal* ones have to be considered given a negative example.

2 Compatibility of DNF formulas with ambiguous positive and negative examples

In *learning from interpretations* De Raedt considers an example as a Herbrand interpretation that is the assignment of truth-values to a set of grounded atoms built from a first order language. In concept learning from interpretations an hypothesis either is a CNF formula, i.e. a conjunction of clauses as in LOGAN_H Khardon (2000) and ICL VanLaer *et al.* (1997) in its CNF mode, or a DNF formula, i.e. a disjunction of partial concept definitions as in ICL in its DNF mode. Our general purpose is to learn such a DNF formula representing a target concept, using both positive and negative ambiguous examples, however we only consider here the propositional case.

Let P be a set of atoms, we will note $a_1 \vee \ldots \vee a_m \leftarrow b_1 \wedge \ldots \wedge b_m$ a clause both containing positive and negative literals of P, $a_1 \vee \ldots \vee a_m$ a clause only containing positive literals, and $\neg(b_1 \wedge \ldots \wedge b_m)$ a clause only containing negative literals (a_is and b_js are atoms). A clausal theory $c_1 \wedge \ldots \wedge c_n$, that is a conjunction of clauses, is represented as the set of clauses $\{c_1, \ldots, c_m\}$. Note that an interpretation i can be represented as a clausal theory $B(i)$ having i as its single model. For example consider the set of atoms $P = \{a, b, c\}$ and the interpretation $i = \{a, b\}$ (meaning that a and b are *true* while c is *false*). Then i can be represented as the clausal theory $\{a, b, \neg c\}$. In our framework 1) a *hypothesis* is a monotone DNF (or DNF$^+$ for short) $h_1 \vee \ldots \vee h_n$ where each h_k is a conjunction of positive litterals, and 2) an ambiguous example is a set of interpretations $e = \{i_1, \ldots, i_n\}$, that also has an intentional representation as a clausal theory $B(e)$ having e as its set of models. The compatibility relation defined hereunder extends the coverage relation used in learning from interpretations and in propositional learning.

Definition 1 (compatibility relations with DNF)
Let H be a DNF and let e be an ambiguous example, then H is compatible$^+$ *with e if and only if there exists an interpretation i in e such that i is a model of H, and H is* compatible$^-$ *with e if and only if there exists an interpretation i in e such that i is not a model of H.*

Let i_1 and i_2 be two interpretations each represented as the set of ground atoms assigned to *True*, then i_1 is *smaller* than i_2 iff $i_1 \subset i_2$. The following property says that when learning a monotone DNF, we only need to keep maximal interpretations when *e* is a positive ambiguous example, and minimal interpretations when *e* is a negative one :

Proposition 1
Let H be a DNF+ hypothesis, then H is compatible$^+$ with a positive ambiguous example e iff there exists a maximal interpretation in e which is a model of H, and H is compatible$^-$ with a negative ambiguous example e iff there exists a minimal interpretation in e which is not a model of H.

Proof[1]

3 Learning DNF from ambiguous examples

LE$_a$ is a standard top-down greedy set covering algorithm whose search space for each partial concept definition is restricted, as in PROGOL Muggleton (1995), to parts of a particular positive example denoted as a *seed*. LE$_a$ learns DNF+ from ambiguous examples and differs from other top-down learners as 1) it has to maintain the coherence of assumptions made on negative examples, 2) it has to handle ambiguous seeds, and 3) it uses *compatibilty* rather than *coverage* in order to deal with ambiguous examples.

[1] All proofs are available at the site `http://www-lipn.univ-paris13.fr/~soldano/Annexe.pdf`

LE$_a$ as described in algorithm 1 works as follows : a first conjunction h_1 *compatible* with at least one positive example (the seed) and no negative examples is selected, then the positive examples compatible with h_1 are discarded. Another seed is selected and a new conjunction h_2 is searched for. The process continues building conjunctions h_i until there is no more positive examples to consider. As each h_i must be *compatible*$^-$ with all negative examples, in our uncertainty setting we have to ensure that the h_is relies on valid assumptions about the negative examples. Suppose for instance that our current DNF is $h_1 = a$ that is *compatible*$^-$ with the negative ambiguous example $e = \{\{a\}, \{b\}\}$ through the second possibility. Thus h_1 makes the assumption that the negative example hidden in e is $\{b\}$. Now if we check the new term $h_2 = b$, we will find that it is *compatible*$^-$ with e through the first possibility, so assuming that the negative example hidden in e is $\{a\}$. As h_1 and h_2 rely on contradictory assumptions about e, the DNF $h_1 \vee h_2$ is not *compatible*$^-$ with e. To avoid this situation, we have to discard the possibilities of e that do not match the assumptions made by any h_i added to the current DNF. This process is achieved for all negative examples.

The core of LE$_a$ is the procedure $bestRule_a$ described in algorithm 2 and whose goal is to find the conjunctive term that will be added to the current DNF. $bestRule_a$ uses a beam search that retains, at each step the W best conjunctions (i.e. the beam) according to the evaluation function. At each step the beam search applies a refinement operator. As in our framework the seed is an ambiguous positive example $\{i_1, \ldots, i_n\}$, our refinement operator $\rho_a(h, seed)$ returns the maximally general specializations of h that are *compatible*$^+$ with $seed$. Let $\rho(h, x)$ be the ususal refinement operator that returns the maximally general specializations of h that covers the positive example x, then $\rho_a(h, \{i_1, \ldots, i_n\}) = \rho(h, i_1) \cup \ldots \cup \rho(h, i_n)$. The refinement operator ρ_a is used in the procedure *maximallyGeneralSpecializations*.

In algorithm $bestRule_a$ we associate to each candidate conjunction h its *ambiguous accuracy*. Here *accuracy(h)* is simply the proportion of examples compatible with h : $accuracy(h) = \frac{n+p}{N+P}$ where N is the number of negative examples, P the number of positive examples still not compatible$^+$ with the current DNF, n the number of negative examples compatible$^-$ with h, and p the number of positive examples compatible$^+$ with h. We also introduce the function *quality(h)* such that *quality(h) = p* if h is *compatible*$^-$ with all the negative examples and else *quality(h)*= 0. Finally our evaluation function is *evaluation(h) = max(quality(h), accuracy(h))*.

We now describe our *multi-table* representation. The key idea is to divide the ambiguous examples in parts called *tables* so that the compatibility can be checked table by table. A table is associated to a set of *connected* atoms, that is atoms that depend on each others. More precisely two atoms a and b are *directly connected* when either $a = b$ or a and b both appear in some clause of the background knowledge B. a and b are *connected* when (a, b) belongs to the transitive closure of the relation *directly connected*. Let us get back to the example of bird given in section 1. From the background knowledge $B = \{red \vee green, \neg(red \wedge green), migratory \vee not\_migratory, \neg(migratory \wedge not\_migratory), light \vee not\_light, \neg(light \wedge not\_light)\}$, we can exhibit 3 sets of connected atoms : $P_1 = \{red, green\}$, $P_2 = \{migratory, not\_migratory\}$ and $P_3 = \{light, not\_light\}$. We use this partition to divide the previous ambiguous example e in 3 tables which cross

Algorithm 1 LE_a

```
input  $E^+$, $E^-$, $W$                                    /* Width of the beam. */
output  $DNF$                  /* a DNF compatible with each example of $E^+$ and $E^-$ */
begin
  $DNF \leftarrow \emptyset$ ;              /* Empty disjunction (compatible with no example). */
  while $E^+ \neq \emptyset$ do
      $h \leftarrow$ bestRule$_a$($E^+$, $E^-$, W) ; $DNF \leftarrow DNF \vee h$ ;
      $E^+ \leftarrow E^+ \setminus$ {examples of $E^+$ compatible$^+$ with $h$} ;
                                        /* Update possibilities of negative examples. */
      for each example $e$ in $E^-$ do
          discard each possibility in $e$ that is a model of $h$ ;
      end for ;        /* Now $h$ is compatible$^-$ with each possibility of each negative example. */
  end while ;
  return $DNF$ ;
end.
```

Algorithm 2 : $bestRule_a$

```
input  $E^+$, $E^-$, $W$                                    /* Width of the beam. */
output  $best$      /* A conjunction compatible with some examples of $E^+$ and with all examples of $E^-$. */
begin  $seed \leftarrow$ any example of $E^+$ ; variabilize($seed$) ; $best \leftarrow \emptyset$ ;
              /* Empty conjunction that is compatible$^+$ with all examples and compatible$^-$ with no example. */
  N $\leftarrow |E^-|$ ; P $\leftarrow |E^+|$ ; quality($best$) $\leftarrow 0$ ;
  accuracy($best$) $\leftarrow \frac{P}{N+P}$ ; evaluation($best$) $\leftarrow$ accuracy($best$) ; $C \leftarrow \{best\}$ ;
  while evaluation($best$) $< P$ and $C \neq \emptyset$ do
      $S \leftarrow$ maximallyGeneralSpecializations($C$, $seed$) ;
      for each conjunction $h$ in $S$ do
          $p \leftarrow$ number of examples of $E^+$ compatible$^+$ with $h$ ;
          $n \leftarrow$ number of examples of $E^-$ compatible$^-$ with $h$ ;
          if $n < N$ then quality($h$) = 0 ; else quality($h$) = $p$ ; endif ;
          accuracy($h$) = $\frac{n+p}{N+P}$ ; evaluation($h$) $\leftarrow$ max(quality($h$), accuracy($h$))
      end for ;
      $C \leftarrow$ the (at most) $W$ conjunctions of $S$ having the best evaluations ;
      if a conjunction $h$ among $C$ has a better evaluation than $best$ then
      evaluation($best$) $\leftarrow$ evaluation($h$) ; $best \leftarrow h$ ; endif ;
      $C \leftarrow C \setminus \{h$ | quality($h$) $> 0$ $\}$ ;
  end while ;
  return $best$ ;
end.
```

product represent the four possibilities of e :

e_1	e_2	e_3

{*red*}	{*migratory*}
	{*not_migratory*}
		{*light*}
		{*not_ligth*}

Each table e_i is a set of possibilities described with atoms of P_i. Consider now the hypothesis h =$\{migratory,\ not\_light\}$, it can be divided in 3 parts with respect to P_1, P_2 and P_3 : h_1 = {}, h_2 = {*migratory*} and h_3 = {*not_light*}. To check that h is *compatible*$^+$ with e, we check that each h_i is *compatible*$^+$ with the corresponding e_i : here h_1 covers $\{red\}$ in e_1, h_2 covers $\{migratory\}$ in e_2 and h_3 covers $\{not\_light\}$ in e_3. As a consequence h covers the possibility {*red, migratory, not_light*} and so is *compatible*$^+$ with e. To check whether h is *compatible*$^-$ with e, now considered as a negative example, we check that at least one h_i does not cover the corresponding e_i : here h_2 does not cover {*not_migratory*} in e_2. As a consequence h does not cover the possibilities {*red, not_migratory, light*} and {*red, not_migratory, not_light*}, and so is *compatible*$^-$ with e.

More formally we will note $S = S_1 + \cdots + S_m$ a *partition* of S and $S = S_1 \oplus \ldots \oplus S_m$ a *weak partition* of S : S_is are subsets of S such that $S_j \cap S_k = \emptyset$ $(j \neq k)$ and $S = S_1 \cup \ldots \cup S_m$ but here some S_i may be empty. Let then $P_1 + \ldots + P_m$ be a partition of P and let S be a set of clauses or a conjunction of atoms. Then the *projection of S with respect to P* is obtained by selecting for each P_k the maximal subset $P_k(S) = S_k$ of S in which only appear atoms of P_k. We say that $P_1 + \ldots + P_m$ is a *valid partition* of P with respect to B if $P(B) = \{P_1(B), \ldots, P_m(B)\}$ is a weak partition of B. As an illustration let $P = \{a, b, d, e, f\}$ and $B = \{a \leftarrow b, b \leftarrow d, e \leftarrow f\}$. Then $P = \{a, b, d\} + \{e, f\}$ is a valid partition w.r.t. B while $P = \{a, d\} + \{b, e, f\}$ is not a valid one. We will note $\mathcal{M}(B)_P$ the models of the clausal theory B w.r.t. P.

Proposition 2
Let e be an ambiguous example expressed from atoms of P and let $B(e)$ be clausal theory having e as set of models. Let $P_1 + \ldots + P_m$ be a valid partition of P w.r.t. $B(e)$, then $e = \mathcal{M}(P_1(B(e)))_{P_1} \times \ldots \times \mathcal{M}(P_m(B(e)))_{P_m}$.

From now on $\mathcal{M}(P_k(B(e)))_{P_k}$ will be simply noted as $T_k(e)$ (the k^{th} table of e), so $e = T_1(e) \times \ldots \times T_m(e)$ is called the *m-table ambiguous example* e. Then there is a m-table representation w.r.t. E if and only if there exists a partition $P = P_1 + \ldots + P_m$ such that each ambiguous example e of E can be expressed as the cross product $T_1(e) \times \ldots \times T_m(e)$. Consider now the *intentional* case in which each ambiguous example e is represented as a set of literals represented as the clausal theory $F(e)$, together with a general background knowledge theory B, then $B(e) = B \cup F(e)$ and it follows that :

Proposition 3
Let $P_1 + \ldots + P_m$ be a valid partition of P with respect to B, and let $F(e)$ be a clausal theory representing a set of ground atoms, then $P_1 + \ldots + P_m$ is a valid partition with respect to $B \cup F(e)$.

Now let us define $P_1(h) \oplus \ldots \oplus P_m(h)$ as the *m-table* conjunctive hypothesis equivalent to h, then :

Proposition 4
Let $T_1(e)\times \ldots \times T_m(e)$ be a m-table ambiguous example and let $P_1(h) \oplus \cdots \oplus P_m(h)$ be a m-table conjunctive hypothesis. Then h is compatible$^+$ e *if and only if each table $T_k(e)$ contains a model of $P_k(h)$ h is $compatible^- e$ if and only if a table $T_k(e)$ contains an interpretation that is not a model of $P_k(h)$*

Proposition 4 allows us to check the compatibility between conjunctive hypothesis and ambiguous examples table by table. Now let us call $min(I)$ (respectively $max(I)$) the set of smallest (respectively greatest) interpretations among the set of interpretations I, then :

Proposition 5
Let $T_1(e)\times\ldots\times T_m(e)$ be a m-table ambiguous example. Then : $min(e) = min(T_1(e)) \times \ldots \times min(T_m(e))$, and $max(e) = max(T_1(e)) \times \ldots \times max(T_m(e))$.

So if e is positive we will only keep the m-table example $max(T_1(e)) \times \ldots \times max(T_m(e))$, if e is negative we will keep $min(T_1(e))\times \ldots \times min(T_m(e))$.
To deal with a m-table representation LE$_a$ has to be modified in such a way that :

- Each ambiguous example e is represented by a set of tables such that $e = T_1(e) \times \ldots \times T_m(e)$ where each $T_k(e)$ is either a set of minimal interpretations if e is negative or of maximal interpretations if e is positive
- Each conjunctive hypothesis h is represented by a set of tables such that $h = P_1(h) \oplus \ldots \oplus P_m(h)$.

LE$_a$ is implemented in Swi-Prolog (Wielemaker (2003)) and is available on request to the first author.

4 Convergence

Hereunder we assume that the learning set is obtained by first drawing independent and identically distributed (i.i.d) positive and negative examples from a universe of instances built on $\{0,1\}^n$. The universe of instances here is the set of valid instances with respect to a possibly unknown background theory B. A *hiding* process, that hides the example within an ambiguous example, is applied to each drawn example. In the particular case of missing values, this hiding process corresponds to a *blocking* process as defined in Schuurmans & Greiner (1997) : the boolean value of each atom of the example can be turned into the value ' ?' with a probability p .

We suppose now that each k-length part of a valid instance x has a non zero probability to be known as *True* in an ambiguous e with the same label as x :

Proposition 6
If each k-uple $(a_1 = v_1 \ldots, a_n = v_n)$, part of some valid instance x, has a non zero probability to be known in an ambiguous example with the same label as x, then when learning a k-term-k-DNF in a i.i.d way, the Version Space converges to a set of hypothesis all equivalent on the universe of instances, for a finite number of ambiguous examples.

Now recall that LE$_a$ translates any DNF problem as a DNF$^+$ problem by adding negated atoms. In LE$_a$, all the possibilities of each ambiguous example are investigated and a hypothesis is stated as a solution by LE$_a$ if and only if it belongs to the version space. However the beam search in a bestRule$_a$ step is of course not exhaustive. Whenever the seed is not ambiguous, the hypothesis space is built on a subset of the atoms of the seed, and thus the seed[2]belongs to this space and does not cover any negative example. However in the case of an ambiguous seed $s = \{s_1, \ldots, s_n\}$, the whole hypothesis space H is the union of several hypothesis space H_i, each built on subsets of a possible complete seed s_i. The search in bestRule$_a$ can then reach a state where no hypothesis in the beam covers the correct s_i hidden in s. In that case bestRule$_a$ can end with no solution. In this case we check whether there exists a possibility in the seed that, as a hypothesis, covers no negative examples. If such a possibility exists, it is returned as a conjunctive term to add to h, otherwise the whole problem has no solution. Given this, the following proposition holds :

Proposition 7
Let c be a concept that can be represented as a DNF, then LE$_a$ always outputs a hypothesis h that belongs to the VS delimited by a set of ambiguous examples of c and so converges, when conditions of proposition 6 are satisfied, to an exact solution for a finite number of ambiguous examples.

5 Experimentation

Our experiments concern attribute-value learning. For each atom a_i, an atom *not*-a_i is added to the hypothesis language whenever learning unrestricted DNF. The background knowledge then always contains at least all the clauses of the form $(a_i \vee \textit{not-}a_i)$ and $\neg(a_i \wedge \textit{not-}a_i)$. In our experiments, we have compared LE$_a$, with a beam of size 3, to C4.5 and Naive Bayes, as implemented in Weka Witten & Frank (1999) and denoted as J48 and NBayes. J48 is used in its unpruned setting and with its default parameters. All our problems, but the last one, are artificial : there always exists a coherent and simple solution.

When splitting a node, J48 propagates a fraction of the example on each son of the node, according to estimated probabilities. In various experiments, this has been shown to be a very efficient, and still simple, way of dealing with missing values (Saar-Tsechansky & Provost, 2007). NBayes represents a simple, robust, and still often accurate probabilistic learner. In all the experiments each learning instance is made incomplete by replacing the truth value of a boolean variable by an unknown tag " ?" with a probability p. For each value of p, 100 trials are performed, and average accuracy and standard deviation are computed. Each trial is performed with a random sample of N_e examples as a learning set. The test set is the same for all the trials and contains only complete examples.

We have experimented LE$_a$ on a simple boolean problem, further denoted as M. We learn $(a1 \wedge a2 \wedge a3) \vee (a2 \wedge a4 \wedge a5) \vee (a5 \wedge a6 \wedge a7) \vee (a7 \wedge a8 \wedge a9)$ as an unrestricted

[2] or more precisely the most specific term which the seed is a model of.

DNF. The variable $a0$ is irrelevant here. An example is described by 20 atoms and negated atoms, and the instance space contains $2^{10} = 1024$ instances, $\tilde{4}0\%$ of which are positive. LE$_a$ generates for each example its multi-table representation, thus resulting in 10 tables of two lines, each corresponding to a pair $\{aj, not\_aj\}$.

We first consider $N_e = 630$ and p ranging from 0 to 0.6 and remark that NBayes is not sensitive to the missing values, whereas J48 and LE$_a$ have accuracies decreasing from 100% to the accuracy of NBayes. LE$_a$ first clearly outperforms J48, with a maximum gain of 9%, and then crashes at the level of NBayes at $p = 0.6$. We then experiment $N_e = 3000$ with p ranging from 0.6 to 0.9 and remark that LE$_a$ again outperforms J48 and then sharply decreases, and is outperformed by NBayes when $p = 0.9$. Here the bias of LE$_a$ and $J48$ outperforms NBayes when there is enough information provided by the incomplete examples :

Prog.	p=0	p=0.1	p=0.2	p=0.3	p=0.4	p=0.5	p=0.6
LE$_a$ (630)	100	99.99	99.99	99.86	98.89(2.57)	92.13(8.13)	78.14(8.21)
J48	99.16	97.40	94.85	92.38	89.63(2.82)	85.38(3.39)	79.67(4.39)
NBayes	79.70	79.62	79.49	79.46	79.35(1.10)	79.17(1.39)	79.00(1.35)

Prog.	p=0.6	p=0.7	p=0.8	p=0.9
LE$_a$ (3000)	98.77(2.63)	87.16(8.97)	70.26(5.65)	66.36(4.60)
J48	81.71(2.06)	71.83(1.90)	62.61(1.17)	59.98(0.0)
NBayes	79.81(0.79)	79.82(0.57)	79.72(0.75)	79.03(1.14)

Now we add constraints to the M problem, turning it to the MC problem. We consider that all the instances are models of $B = \{a0 \leftarrow a1, a2 \leftarrow a3, a4 \leftarrow a5, a6 \leftarrow a7, a8 \leftarrow a9\}$. LE$_a$ will only consider as possibilities for each ambiguous example e those that are models of B. The multi-table representation exhibits here only 5 tables of the form $\{ai, not\_ai, ai+1, not\_ai+1\}$ because now $a0$ is related to $a1$, $a2$ is related to $a3$ and so on. The results are as follows :

Prog.	p=0	p=0.1	p=0.2	p=0.3	p=0.4	p=0.5	p=0.6
LE$_a$ (630)	100	100	99.98	99.85	99.77(0.83)	98.59(2.27)	94.83(4.88)
J48	100	99.56	99.07	98.42	97.36(1.91)	94.67(2.72)	88.57(4.06)
NBayes	84.56	84.51	84.42	84.46	84.47(0.99)	84.36(0.94)	84.09(1.23)

Prog.	p=0.6	p=0.7	p=0.8	p=0.9
LE$_a$ (3000)	99.34(1.37)	97.54 (2.54)	90.86 (5.72)	82.40 (6.73)
J48	93.94(1.63)	80.53(2.17)	70.35(1.60)	69.82(0.0)
NBayes	86.29(0.75)	84.33(0.62)	84.25(0.87)	85.54(1.14)

The first comment here is that it's much easier to learn our DNF when the universe of instances is reduced through constraints. LEa, J48 and Bayes perform better in learning MC than in learning M. For instance, learning MC with 630 examples with $p = 0.6$ results in accuracies from $\approx 95\%$ to $\approx 84\%$ when learning M results in accuracies $\approx 79\%$. The second comment is that LE$_a$, again, seems much more resistant to ambiguity,

[2] Unexpectedly sometimes LEaNC is better than LEa, and sometimes LE$_a$ is better, but in much cases there is no significant differences between them.

and its accuracy decreases slower than those of J48 or other programs. For instance when $N_e = 3000, p = 0.9$ the accuracy of LE$_a$ is close to $\approx 80\%$ when that of J48 is about 70%. However at such a high level of uncertainty NBayes is still better than LE$_a$. In the next experiment, we investigate accuracies with $p = 0.9$ and increasing values of N_e ranging from 6000 to 24000 examples. The result clearly is that LE$_a$ then benefits from this additional information and outperforms NBayes :

Prog.	nb=6000	nb=12000	nb=24000
LE$_a$ (p=0.9)	85.28(5.50)	86.28(6.34)	89.26(5.97)
J48	67.48(0.00)	67.70(0.13)	66.41(0.00)
NBayes	84.80(1.09)	84.22(0.78)	85.84(0.61)

5.1 Problem Breast-w5

In this last experiment we address a problem of the UCI database (Breast cancer Wisconsin) whose accuracy, as reported in Lim *et al.* (2000) ranges from 91 to 97%. There are 9 numeric variables but we only consider the 5 first variables. We use a boolean description of each numeric value by defining atoms as $x \leq x_1, x > x_1, x \leq x_2, x > x_2, \ldots x \leq x_n, x > x_n$ and adding to the background knowledge all the clauses of the form $\neg(x \leq x_i \wedge x > x_i)$, $x \leq x_i \leftarrow x \leq x_{i+1}$, and $x > x_i \leftarrow x > x_{i-1}$. Here the thresholds are computed on all the data but ignoring the label of the instances, and using equal frequency intervals with a maximum of 9 thresholds *per* numeric variable. The test set contains the last 249 instances whereas the learning set is drawn within the 400 remaining complete examples to which we apply our blocking process with various values of p. Note that here, after the blocking process is applied, the numeric value of a variable x in an instance may still be constrained to an interval, possibly larger than its initial interval $]x_i, x_{i+1}]$. So, in some sense we address also the problem of *imprecise* values. In our experiment hereunder we consider 100 learning examples and p ranges from 0.5 to 0.95 :

Prog.	p=0.5	p=0.6	p=0.7	p=0.8	p=0.9	p=0.95
LE$_a$	94.56(3.2)	94.76(3.0)	95.01(3.1)	94.32(3.6)	92.25(7.3)	90.67(7.9)
J48	96.26(2.3)	95.60(3.0)	95.82(2.6)	94.07(5.4)	89.75(8.0)	78.40(7.2)
NBayes	98.26(0.2)	98.26(0.2)	98.28(0.2)	98.32(0.2)	98.40(0.2)	98.46(0.26)

Even with a very weak information (few examples with many missing values) the various programs perform well. NBayes has a high accuracy, LE$_a$ and J48 build very simple solutions but are outperformed by NBayes. J48 in this task first outperforms LE$_a$ but begins to decrease for lower values of p. LE$_a$ is better when p is greater than 0.9. Clearly problems with nominal, hierarchic and numeric attributes should be further investigated, but at least on this example, using LE_a results in interesting accuracies for high levels of incompleteness.

5.2 CPU-time

The benefits of the multi-table implementation are clear : we hardly observe any increase of CPU-time with increasing uncertainty probability p, when the average number of possibilities *per* example is 2^{n*p}. For instance, in the MC problem with 3000 examples and p ranging from 0.6 to 0.9, the CPU-time on a intel Dual core was about 1 hour per 100 trials for all values of p.

6 Related work

In the Multiple instance learning setting originally proposed by Dietterich Dietterich *et al.* (1997) each example *e* of the target concept is a set $\{inst_1, \ldots\ inst_n\}$ of descriptions called instances. A positive example e^+ works as an ambiguous example : at least one instance (possibly several ones) has to satisfy the target concept[3]. A negative example e^- works differently : it is required that none of its instances satisfy the target concept. The same setting occurs with multiple part problems, as defined in Zucker & Ganascia (1998), and in various attempts to propositionalize first order learning problems in order to use variants of efficient propositional or attribute-value learners (Alphonse & Rouveirol, 2000; Sebag & Rouveirol, 2000). Note that a slight modification of LE$_a$ allows to address Multiple-Instance problems : here a hypothesis h has to be defined as compatible$^-$ with a negative example e whenever h is not compatible$^+$ with e.

Uncertainty in propositional or attribute-value representations is addressed with basically two approaches : either predicting the complete description or taking into account the missing values when scoring the hypotheses. The former approach includes single or multiple imputation methods (Dick *et al.*, 2008) and methods that learn from the examples to predict the missing values (Liu *et al.*, 1997). In the latter approach, the scoring function to optimize when searching a preferred solution is weighted according to an estimation of the probability distribution of the possible values for uncertain attributes at each node of a decision tree, as in C4.5 Quinlan (1993).

7 Perspectives and Conclusion

In this paper, we have discussed learning from ambiguous examples from a pure logical point of view and shown that the proposed method was efficient, thanks to the multi-table representation. It is far more robust to very high level of uncertainty than popular approaches in Machine Learning, as long as enough examples, even extremely incomplete, are provided. However the experiments here are only preliminary, further ones have to be performed on various attribute-values problems. We are currently experimenting a variant of LE_a that address first order problems. Future research directions includes experiments on uncertainty models more realistic than the independent blocking process experimented here and ways to make the approach robust to various data incompleteness scenarii.

[3] More precisely a boolean function i is associated with each example *e* : if *e* is positive $\exists inst \in e$ such that *f(inst) = true*, and if *e* is negative $\forall inst \in e, f(inst) = false$.

Références

ALPHONSE É. (2004). Macro-operators revisited in inductive logic programming. In *ILP*, volume 3194 of *Lecture Notes in Computer Science*, p. 8–25 : Springer.

ALPHONSE É. & ROUVEIROL C. (2000). Lazy propositionalization for relational learning. In W. HORN, Ed., *Proc. of ECAI'2000*, p. 256–260 : IOS Press.

DERAEDT L. (1997). Logical settings for concept-learning. *Artif. Intell.*, **95**(1), 187–201.

DICK U., HAIDER P. & SCHEFFER T. (2008). Learning from incomplete data with infinite imputations. In *ICML '08*, p. 232–239, New York, NY, USA : ACM.

DIETTERICH T. G., LATHROP R. H. & LOZANO-PEREZ T. (1997). Solving the multiple instance problem with axis-parallel rectangles. *Artif. Intell.*, **89**(1-2), 31–71.

HIRSH H. (1994). Generalizing version spaces. *Mach. Learn.*, **17**(1), 5–46.

KAKAS A. C. & RIGUZZI F. (2000). Abductive concept learning. *New Generation Computing*, **18**(3), 243–294.

KHARDON R. (2000). Learning horn expressions with logan-h. In *ICML '00 : Proceedings of the Seventeenth International Conference on Machine Learning*, p. 471–478, San Francisco, CA, USA : Morgan Kaufmann Publishers Inc.

LIM T.-S., LOH W.-Y. & SHIH Y.-S. (2000). A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. *Machine Learning*, **40**(3), 203–228.

LIU W. Z., WHITE A. P., THOMPSON S. G. & BRAMER M. A. (1997). Techniques for dealing with missing values in classification. In *IDA '97 :*, p. 527–536, London, UK : Springer-Verlag.

MITCHELL T. M. (1982). Generalization as search. *Artif. Intell.*, **18**(2), 203–226.

MUGGLETON S. (1995). Inverse entailment and Progol. *New Generation Computing*, **13**(3-4), 245–286.

QUINLAN J. R. (1993). *C4.5 : programs for machine learning*. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc.

SAAR-TSECHANSKY M. & PROVOST F. (2007). Handling missing values when applying classification models. *Journal of machine learning research*, **8**, 1623–1657.

SCHUURMANS D. & GREINER R. (1997). Learning to classify incomplete examples. In *Computational Learning Theory and Natural Learning Systems : Addressing Real World Tasks*, p. 87–105 : MIT Press.

SEBAG M. & ROUVEIROL C. (2000). Resource-bounded relational reasoning : Induction and deduction through stochastic matching. *Machine Learning Journal*, **38**, 43–65.

VANLAER W., DERAEDT L. & DZEROSKI S. (1997). On multi-class problems and discretization in inductive logic programming. In *ISMIS'97*, p. 277–286.

WIELEMAKER J. (2003). An overview of the SWI-Prolog programming environment. In F. MESNARD & A. SEREBENIK, Eds., *Proceedings of the 13th International Workshop on Logic Programming Environments*, p. 1–16, Heverlee, Belgium : Katholieke Universiteit Leuven. CW 371.

WITTEN I. H. & FRANK E. (1999). *Data Mining : Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann.

ZUCKER J.-D. & GANASCIA J.-G. (1998). Learning structurally indeterminate clauses. In D. PAGE, Ed., *ILP*, volume 1446 of *LNCS*, p. 235–244 : Springer.