Learning family relationships exploiting multistrategy

Stefano Ferilli, Nicola Fanizzi, Teresa M.A. Basile and Nicola Di Mauro
Dipartimento di Informatica
Università di Bari
via E. Orabona, 4 - 70125 Bari - Italia
{ferilli, fanizzi, basile, nicodimauro}@di.uniba.it

Abstract. This work presents the application of INTHELEX, an incremental learning system enhanced by multistrategy capabilities, on a dataset concerning family relationships. The aim is to investigate if and how much abduction, abstraction and deduction can support pure induction. The reported experimental results, although preliminary, demonstrate that such a cooperation may improve efficiency and effectiveness of the learning process.

1 Introduction

Supervised Machine Learning aims, given positive and negative examples of some concept and possibly some background knowledge, at finding a theory that accounts for all positive examples and is consistent with the negative ones and the background knowledge. While the learning process is often run all at once on the whole evidence, sometimes this is not possible because the examples set is not entirely available since the beginning or the concept to be learnt has a changing nature and therefore has to be captured over time. In such cases, being able to refine a previously generated theory, by taking into account new evidence as long as it becomes available, is the only way to overcome the problem. In fact, in the Machine Learning (ML) literature there are systems that try to modify an existing incomplete or inconsistent theory to fit a new set of pre-classified examples. Another important aspect to be analyzed is the possibility of combining together different reasoning methods and learning strategies. Indeed, while at the beginning ML research focused on single-strategy methods that apply a primary type of inference and/or computational mechanism, more recently the limitations of these methods led to exploit/combine various, different and complementary learning strategies together. This mimes the typical ability of humans to apply a great variety of learning strategies depending on the particular situation and problem faced. A theoretical framework for integrating different learning strategies is the Inferential Learning Theory (ILT) [10].

All these considerations, plus the need of testing theoretical results on the Object Identity paradigm [6] in practice, led to the design and implementation of the learning system INTHELEX [5]. Its most characterizing features are in its incremental nature, in the reduced need of a deep background knowledge, in the exploitation of negative information, in the peculiar bias on the generalization model, which reduces the search space and does not limit the expressive power of the adopted representation language, and in the possibility to exploit multistrategical operators whenever necessary during theory revision.

This paper is organized as follows. In Section 2 a general description of the system and of the way in which it integrates multiple reasoning strategies is presented. Section 3 shows experiments concerning application of the multistrategy operators to see if and how they are able to enhance the system performance. Finally, Section 4 draws some conclusions and future work.

2 Multistrategical learning in INTHELEX

INTHELEX (INcremental THEory Learner from EXamples) is a learning system for the induction of logic theories from examples [5]. Among its characterizing features:

- it is based on the *Object Identity assumption* (terms, even variables, denoted by different names within a formula must refer to different objects)[1];
- it learns theories expressed as sets of Datalog^{OI} clauses [11] from positive and negative examples;
- it can learn simultaneously *multiple concepts*, possibly related to each other according to a given hierarchy (recursion is not allowed);
- it guarantees validity of the learned theories on all the processed examples;
- it is a *closed loop* learning system (i.e. a system in which feedback on performance is used to activate the theory revision phase [1]);
- it is *fully incremental*, i.e. in addition to the possibility of refining a previously generated version of the theory, learning can also start from an empty theory[2].

It exploits a (possibly empty) previous theory, a graph describing the dependence relationships among concepts, and a historical memory of all the past examples that led to the current theory. Whenever a new example is taken into account, it is stored in such a repository and the current theory is checked against it.

A novelty in INTHELEX is the integration of various multistrategy operators that may help in the solution of the theory revision problem by pre-processing the incoming information [6]. Deduction is exploited to fill observations with information that is not explicitly stated, but is implicit in their description, and hence refers to the possibility of better representing the examples and, consequently, the inferred theories. Abduction aims at completing possibly partial information in the examples (adding more details). Lastly, abstraction allows to the system to carry out language shift in the examples descriptions and hence in the theory rules. Even if with opposite perspectives, both abduction and abstraction aim at reducing the computational effort required to learn a correct theory with respect to the incoming examples. More details on the theoretical foundations of the cooperation of these strategies in our environment are given in [3], whereas this paper focuses on their performance.

[1] This often corresponds to human intuition, while allowing the search space to fulfill nice properties affecting efficiency and effectiveness of the learning process [11].

[2] As already noticed in the previous section, this can be an important characteristic for handling real-world situations.

2.1 Induction

Induction means inferring, from a certain number of significant observations, regularities and laws valid for the whole population.

INTHELEX incorporates two inductive refinement operators, one for generalizing hypotheses that reject positive examples, and the other for specializing hypotheses that explain negative examples.

In the former case, firstly one of the clauses defining the wrong concept in the theory is chosen. Then, the lgg_{OI} of this clause and the example is computed [11], by taking into account a number of parameters that restrict the search space according to the degree of generalization to be obtained and the computational budget allowed. If one of the lgg_{OI}'s is consistent with all the past negative examples, then it replaces the chosen clause in the theory, or else a new clause is chosen to compute the lgg_{OI}. If no clause can be generalized in a consistent way, the system checks if the example itself, with the constants properly turned into variables, is consistent with the past negative ones. If so, such a clause is added to the theory, or else the example is considered an exception.

In the specialization phase, the system chooses the clause to be refined among those occurring in the SLD-derivation of the example, starting from the lowest possible level and going upwards, and tries to add to it one (or more) positive literal(s), which characterize all the past positive examples and can discriminate them from the current negative one. Again, parameters that bound the search for the set of literals to be added are considered. In case of failure on all of the clauses in the derivation, the system tries to add the negation of a literal, that is able to discriminate the negative example from all the past positive ones, to the clause related to the concept the example is an instance of. If this fails too, the negative example is considered an exception.

New incoming observations are always checked against the exceptions (both positive and negative) before applying the general rules.

2.2 Deduction

INTHELEX requires the observations to be expressed only in terms of the set of predicates that make up the description language for the given learning problem. To ensure uniformity of the example descriptions, such predicates have no definition. Nevertheless, since the system is able to handle a hierarchy of concepts, combinations of these predicates might identify higher level concepts that is worth adding to the descriptions in order to raise their semantic level. For this reason, INTHELEX implements a saturation operator that exploits deduction to recognize such concepts and explicitly add them to the examples description. Indeed, the system can be provided with a background knowledge that is supposed to be correct, hence not modifiable, and contains (complete or partial) definitions in the same format as the theory rules. Any time a new example is considered, a preliminary saturation phase can be performed, that adds the higher level concepts whose presence can be deduced from theory and/or background knowledge rules by subsumption and/or resolution. In particular, the generalization model of implication under Object Identity is exploited [4].

Since all the specific information used by saturation is left in the example description, it is preserved in the learning process until other evidence reveals it is not significant for the concept definition, which is a more cautious behaviour. This is fundamental when the saturation phase involves concepts to be learnt (i.e., theory rules), since their definition could not be stable yet, and hence the preserved information might be needed to recover from deductions made because of wrong rules.

2.3 Abduction

Abduction was defined by Peirce as hypothesizing some facts that, together with a given theory, could explain a given observation. According to the framework proposed in [8], an *abductive logic theory* is made up by a normal logic program [9], a set of *abducibles*, i.e. the predicates about which assumptions (*abductions*) can be made, and a set of *integrity constraints*, that provide indirect information about them (each corresponds to a combination of literals that is not allowed to occur).

The proof procedure implemented in INTHELEX starts from a goal and a set of initial assumptions and results in a set of consistent hypotheses (abduced literals) by intertwining *abductive* and *consistency derivations*. Intuitively, an abductive derivation is the standard Logic Programming derivation suitably extended in order to consider abducibles. As soon as an abducible atom is encountered, it is added to the current set of hypotheses, provided that any integrity constraint containing it is satisfied, which happens when its components are not all true. This is checked by means of a consistency derivation, that in turn may start an abductive derivation to prove the falsity of the abducibles it encounters.

Abduction can be exploited at various moments in both the inductive refinement operators, according to a parameter introduced by the user. Specifically, during generalization the system can decide to use the abduction procedure in one of the following cases: before performing the lgg_{OI}, or before turning the constants of the example into variables, or before adding the example as an exception. On the other hand, in the specialization phase the abduction procedure can be started before trying to add one (or more) positive literal(s) to a theory clause, or before trying to append negative information to a definition, or before adding the example as an exception. The later the application of abduction, the less influence of uncertain information on the final theory and the more frequent theory changes.

2.4 Abstraction

Abstraction is a pervasive activity in human perception and reasoning. A possible exploitation of abstraction concerns the shift from the language in which the theory is described to a higher level one. According to the framework proposed in [13], concept representation deals with entities belonging to three different levels. Concrete objects reside in the *world*, but any observer's access to it is mediated by his *perception* of it. To be available over time, these stimuli must be memorized in an organized *structure*, i.e. an *extensional* representation of the perceived world. Finally, to reason about the perceived world and communicate with other agents, a *language* is needed, that describes it *intensionally*. If we assume that perception is the source of information, that is recorded into a structure and then described by a language, modifications to the latter two are just a consequence of differences in the former (due, e.g., to the medium used and/or the focus-of-attention). Thus, abstraction takes place at the world-perception level by means of a set of operators, and then propagates to higher levels, where it is possible to identify operators corresponding to the previous ones. An abstraction theory contains information for performing the shift specified by the abstraction operators.

In INTHELEX, it is assumed that the abstraction theory is already given (i.e., it has not to be learned by the system), and that the system automatically applies it to each incoming observations before processing it. The implemented abstraction operators allow the system to carry out language shift in a number of different ways: by eliminating superfluous details, by grouping specific component patterns into compound objects, by reducing the number of object attributes, by ignoring the number of instances of a given object or, lastly, by obtaining a coarser grain-size for attribute values.

3 Experimental results

This section presents some results showing the effectiveness and efficiency of the system when its multistrategy capabilities are exploited. In particular, the experiments concerned learning definitions for family relationships, and were carried out exploiting the same dataset as in [2], that includes a total of 778 facts describing people belonging to a given family. Specifically, 159 of these facts concerned basic observations (72 *parent* facts, 31 *male* facts, 24 *female* facts and 32 *married* facts), and were taken as the description for all examples. The other 619 concerned the relationships to be learnt, and hence were used as the set of positive examples. Other 619 negative examples for the same concepts were generated from positive ones, resulting in the following global distribution of the examples among the concepts to be learnt: *father* (36+, 58-), *mother* (36+, 58-), *brother* (44+, 50 -), *sister* (38+, 51-), *son* (46+, 56-), *daughter* (26+, 57-), *aunt* (50+, 47-), *uncle* (57+, 48-), *grandmother* (32+, 55-), *grandfather* (32+, 52-), *cousin* (126+, 42-) and *cousine* (96+, 45-).

A first experiment was performed on the whole dataset by assuming independence between the concepts to be learnt and without providing the system with any background knowledge. The learned theory, reported in Figure 1, was made up of just one definition for *father*, *mother*, *brother*, *sister*, *son*, *daughter*, *grandmother* and *grandfather*, two definitions for *aunt*, *cousin* and *cousine* and three for *uncle*. Figure 1, does not present one of the definitions for cousine since it was added to the theory when the last example for this concept was considered, and hence it is very specific due to the fact that it was never refined. It is worth noting that, starting from observations made up of 159 literals, the clauses in the resulting theory have a length ranging between 3 literals (for *son*) and 9 (for *brother*). These 17 definitions (clauses) were obtained performing 30 lgg's, for an average number of 1.8 lgg's per clause. As to the concepts for which more than one clause was needed, a possible explanation is the following: an uncle (respectively, an aunt) is the brother/brother-in-law (respectively, the sister/sister-in-law) of either the mother or the father, and INTHELEX distinguishes all the specific situations (some of the possible combinations are not represented because of lacking of corresponding examples). A similar interpretation can be provided for the other concepts, *cousin* and *cousine*.

To avoid multiple definitions for some concepts we tried to neglect the person's sex. Driven by the above observations, and with the aim of improving the results, a new experiment was run in which the system was provided with a background knowledge containing the definitions of *sibling* and *au* (corresponding to the brother/sister and uncle/aunt relationships, respectively, regardless of the person's sex). The resulting theory (see Figure 2) contained one clause for all of the concepts; moreover the new definitions also eliminated superfluous information that was present in the previous theory. Such a result is supported by the increased average number of lgg's performed, which is now 2.25 per clause (27 lgg's, 12 clauses).

All these hypotheses, however, are represented by means of basic predicates only. Hence,

```
father(A,B)      :- male(A),parent(C,B),female(C),parent(A,B).
mother(A,B)      :- female(A),parent(C,B), male(C),parent(A,B).
son(A,B)         :- male(A),parent(B,A),parent(C,A).
daughter(A,B)    :- female(A),married(C,B),married(B,C),
                    parent(B,A),parent(C,A).
grandmother(A,B):- female(A),parent(C,B),parent(D,C),male(D),
                    parent(A,C), married(A,D),married(D,A).
grandfather(A,B):- male(A),parent(C,B),parent(D,C),female(D),
                    parent(A,C),married(D,A),married(A,D).
brother(A,B)     :- male(A),parent(C,B),male(C),married(D,C),
                    female(D),married(C,D),parent(D,B),
                    parent(C,A),parent(D,A).
sister(A,B)      :- female(A),parent(C,B),male(C),parent(C,A).
uncle(A,B)       :- male(A),parent(C,B),male(C),parent(D,C),
                    male(D),parent(A,E),female(E).
uncle(A,B)       :- male(A),parent(C,B),female(C),parent(D,C),
                    male(D),parent(D,A).
uncle(A,B)       :- male(A),male(B),parent(C,B),male(C),
                    parent(D,C),female(D),parent(D,A).
aunt(A,B)        :- female(A),parent(C,B),male(C),parent(D,C),
                    male(D),parent(D,A).
aunt(A,B)        :- female(A),parent(C,B),parent(C,E),male(D),
                    female(C),parent(D,C),parent(D,A).
cousin(A,B)      :- male(A),parent(C,B),male(C),parent(D,C),
                    male(D),parent(D,E),parent(E,A).
cousin(A,B)      :- male(A),parent(C,B),female(C),parent(D,C),
                    male(D),parent(D,E),parent(E,A).
cousine(A,B)     :- female(A),parent(C,B),male(C),parent(D,A),
                    male(D),parent(E,D),male(E).
```

Figure 1: Definitions generated with the pure induction

in order to obtain definitions expressed in a higher level language, we tried to use abstraction. For this purpose, we focused on concepts *grandmother* and *grandfather*, since they seemed sufficiently complex to justify a higher level description. Knowing that “a grandmother (respectively, grandfather) is the mother (respectively, father) of either the father or the mother”, we aimed at checking if the system was able to learn a theory reflecting such information. Thus we provide it with an abstraction theory containing the definitions for *mother* and *father* that were learnt by the system itself in the previous experiments. The result, shown in Figure 3, was indeed as expected.

Lastly, an experiment to check the effect of applying the abductive operator was performed. In this case, in order for abduction to be meaningful, incomplete observations were needed, which was obtained by corrupting the available family description according to the modalities described in [7] (where the same dataset was exploited). Also according to the approach in the same paper, other changes were made to the problem setting. First of all, only the examples about *father* were taken into account: the training set included 36 positive examples and 200 negative ones that were randomly generated. Moreover, the examples description is more complex than before, in that it includes not only the basic observations but

```
father(A,B)       :- male(A),parent(A,B).
mother(A,B)       :- female(A),parent(A,B).
son(A,B)          :- male(A),parent(B,A).
daughter(A,B)     :- female(A),parent(B,A).
grandmother(A,B)  :- female(A),parent(C,B),parent(A,C).
grandfather(A,B)  :- male(A),parent(C,B),parent(A,C).
brother(A,B)      :- male(A),parent(C,B),sibling(B,A),
                     male(C),sibling(A,B),parent(C,A).
sister(A,B)       :- female(A),parent(C,B),parent(C,A),
                     male(C),sibling(A,B),sibling(B,A).
uncle(A,B)        :- male(A),parent(C,B),au(A,B),
                     sibling(A,C),sibling(C,A).
aunt(A,B)         :- female(A),parent(C,B),parent(D,C),
                     au(A,B),sibling(A,C),sibling(C,A),
                     male(D),parent(D,A).
cousin(A,B)       :- male(A),parent(C,B),male(C),au(D,B),
                     parent(E,D),male(E),au(C,A).
cousine(A,B)      :- female(A),parent(C,B),female(C),
                     male(D),married(C,D),married(D,C),
                     parent(D,B),au(C,A),au(D,A).
```

Figure 2: Definitions generated exploiting deduction

also all the known facts concerning the concepts other than *father* [3], for a total of 742 literals. Progressive corruption of such a complete description was obtained by randomly eliminating facts from it. Specifically, learning was run on the following percentages of preserved descriptions: 100% (no incompleteness), 90%, 80%, 70%, 60%, 50% and 40%. Hence, the description size varied as follows: 742 literals (100%), 668 literals (90%), and so on. For each percentage, 5 different corrupted observations were generated, and 5 corresponding learning problems were obtained whose performances were averaged.

The abductive theory (see Section 2.3) for this domain contained the following set of *abducibles* [4]: parent, male, female, married, mother, son, daughter, grandmother, grandfather, brother, sister, cousin, cousine, uncle, aunt. The *integrity constraints* are reported in Figure 4; their interpretation is: "one person cannot be both male and female"; "a son cannot be female, and vice versa"; "a daughter cannot be male, and vice versa". INTHELEX was allowed to exploit abduction to hypothesize facts, concerning the above descriptors, before performing lgg's.

If we compare the performance with and without abduction on the corrupted datasets, the benefit becomes very evident with respect to all the parameters taken into account in Table 1: number of clauses, number of lgg's, runtime and accuracy. It is possible to note that the number of lgg's per clause is higher with abduction than without it for less corrupted datasets, indicating that abduction is less useful when descriptions tend to be complete. On the contrary, in more corrupted cases, abduction shows its power since it is able to preserve

[3] This can be seen as a saturation of examples.

[4] By definition, *abducibles* cannot be concepts to be learnt, neither can they have definitions in the theory. The presence of concepts such as son, daughter etc. here is allowed merely because of the new experimental setting with the respect to the previous ones.

```
Abstraction theory

father(A,B) :- male(A),parent(A,B).
mother(A,B) :- female(A),parent(A,B).

Generated Definitions

grandfather(A,B) :-  father(A,C),mother(C,B).
grandfather(A,B) :-  father(A,C),father(C,B).
grandmother(A,B) :-  mother(A,C),mother(C,B).
grandmother(A,B) :-  mother(A,C),father(C,B).
```

Figure 3: Definitions generated exploiting abstraction

```
Integrity Constraints

ic([male(X),female(X)]).
ic([son(X,Y),female(X)]).
ic([daughter(X,Y),male(X)]).
```

Figure 4: Abductive Theory: integrity constraints

the theories from being refined (indeed, the number of lgg's per clause dramatically decreases and goes below the case without abduction). Moreover, lower runtimes prove that the abductive procedure is also efficient. Finally, note that the number of clauses is less when using abduction in all corrupted cases, nevertheless predictive accuracy is always comparable to, and in some cases significantly higher than, the case without abduction.

4 Conclusions and Future Work

Approaches to machine learning that combine different reasoning strategies can help in obtaining more efficiency and effectiveness and hence might turn out to be useful in a number of real-world situations. The incremental system INTHELEX works on first-order logic representations. Its multistrategy learning capabilities were obtained by augmenting pure induction and abduction with abstraction and deduction. This paper presents some sample results proving the benefits that the addition of each strategy can bring. Future work will concern a more extensive experimentation, aimed at finding tighter ways of cooperation among the learning strategies, and an analysis of the complexity of the presented techniques. Supported by previous successful experience in the application of symbolic learning techniques to paper documents [6, 12], benefits of applying multistrategy to real-world problems, such as learning rules for classification and interpretation of cultural heritage material, are currently under study. Specifically, INTHELEX is being exploited as a learning component in EU project COLLATE, dealing with historical filmographic documents concerning European production in the 20ies and 30ies.

Table 1: Abduction on family dataset

		Clause	Lgg	Lgg/Clause	Runtime	Accuracy
100%	noabd	1	1.6	1.6	34.65	0.994
	abd	1.2	1.6	1.3	39.22	0.994
90%	noabd	3.6	5.6	1.6	377.62	0.956
	abd	1.4	3.2	2.3	54.68	0.986
80%	noabd	4.4	9.2	2.1	362.67	0.948
	abd	1.2	3	2.5	49.6	1
70%	noabd	6.6	7.8	1.2	615.82	0.940
	abd	1.2	2.4	2	39.29	1
60%	noabd	6.8	10.4	1.5	323.11	0.924
	abd	2.2	3.2	1.4	55.34	1
50%	noabd	7.6	9.4	1.2	294.85	0.920
	abd	2.2	1.4	0.6	29.6	1
40%	noabd	10.6	11.8	1.1	446.05	0.896
	abd	1.8	1.2	0.7	19.36	1

Acknowledgements.

This work was partially funded by the EU project IST-1999-20882 project COLLATE (Collaboratory for Annotation, Indexing and Retrieval of Digitized Historical Archive Material) (URL:`http://www.collate.de`).

References

[1] J. M. Becker. Inductive learning of decision rules with exceptions: Methodology and experimentation. B.s. diss., Dept. of Computer Science, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA, 1985. UIUCDCS-F-85-945.

[2] H. Blockeel and L. De Raedt. Inductive database design. In *ISMIS96*, volume 1079 of *Lecture Notes in Artificial Intelligence*, pages 376–385. SV, 1996.

[3] F. Esposito, N. Fanizzi, S. Ferilli, and G. Semeraro. Abduction and abstraction in inductive learning. In *Proceedings of the 5th International Workshop on Multistrategy Learning*, Guimarães, Portugal, 2000.

[4] F. Esposito, N. Fanizzi, S. Ferilli, and G. Semeraro. Oi-implication: Soundness and refutation completeness. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, pages 847–852, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers.

[5] F. Esposito, G. Semeraro, N. Fanizzi, and S. Ferilli. Multistrategy Theory Revision: Induction and abduction in INTHELEX. *Machine Learning Journal*, 38(1/2):133–156, 2000.

[6] S. Ferilli. *A Framework for Incremental Synthesis of Logic Theories: An Application to Document Processing*. Ph.D. thesis, Dipartimento di Informatica, Università di Bari, Bari, Italy, November 2000.

[7] A. C. Kakas and F. Riguzzi. Learning with abduction. In S. Džeroski and N. Lavrač, editors, *Proceedings of the 7th International Workshop on Inductive Logic Programming*, volume 1297, pages 181–188. Springer-Verlag, 1997.

[8] E. Lamma, P. Mello, F. Riguzzi, F. Esposito, S. Ferilli, and G. Semeraro. Cooperation of abduction and induction in logic programming. In A. C. Kakas and P. Flach, editors, *Abductive and Inductive Reasoning: Essays on their Relation and Integration*. Kluwer, 2000.

[9] J. W. Lloyd. *Foundations of Logic Programming*. Springer-Verlag, Berlin, second edition, 1987.

[10] R.S. Michalski. Inferential theory of learning. developing foundations for multistrategy learning. In R.S. Michalski and G. Tecuci, editors, *Machine Learning. A Multistrategy Approach*, volume IV, pages 3–61. Morgan Kaufmann, San Mateo, CA, 1994.

[11] G. Semeraro, F. Esposito, D. Malerba, N. Fanizzi, and S. Ferilli. A logic framework for the incremental inductive synthesis of Datalog theories. In N.E. Fuchs, editor, *Proceedings of 7th International Workshop on Logic Program Synthesis and Transformation - LOPSTR97*, volume 1463 of *LNCS*, pages 300–321. Springer, 1998.

[12] G. Semeraro, N. Fanizzi, S. Ferilli, and F. Esposito. Document classification and interpretation through the inference of logic-based models. In P. Constantopoulos and I.T. Sølvberg, editors, *Research and Advanced Technology for Digital Libraries*, number 2163 in Lecture Notes in Computer Science, pages 59–70. Springer-Verlag, 2001.

[13] J.-D. Zucker. Semantic abstraction for concept representation and learning. In R. S. Michalski and L. Saitta, editors, *Proceedings of the 4th International Workshop on Multistrategy Learning*, Desenzano del Garda, Italy, 1998.