Learning with Abduction

A.C. Kakas[1], F. Riguzzi[2]

[1] Department of Computer Science, University of Cyprus
75 Kallipoleos str., CY-1678 Nicosia, Cyprus
antonis@turing.cs.ucy.ac.cy
[2] DEIS, Università di Bologna
Viale Risorgimento 2, 40136 Bologna, Italy
friguzzi@deis.unibo.it

Abstract. We investigate how abduction and induction can be integrated into a common learning framework through the notion of Abductive Concept Learning (ACL). ACL is an extension of Inductive Logic Programming (ILP) to the case in which both the background and the target theory are abductive logic programs and where an abductive notion of entailment is used as the coverage relation. In this framework, it is then possible to learn with incomplete information about the examples by exploiting the hypothetical reasoning of abduction.
The paper presents the basic framework of ACL with its main characteristics. An algorithm for an intermediate version of ACL is developed by suitably extending the top-down ILP method and integrating this with an abductive proof procedure for Abductive Logic Programming (ALP). A prototype system has been developed and applied to learning problems with incomplete information.

1 Introduction

The problem of integrating abduction and induction in Machine Learning systems has recently received renewed attention with several works on this topic [5, 2, 1]. In [5] the notion of Abductive Concept Learning (ACL) was proposed as a learning framework based on an integration of Inductive Logic Programming (ILP) and Abductive Logic Programming (ALP) [8].

Abductive Concept Learning is an extension of ILP that allows us to learn abductive logic programs of ALP with abduction playing a central role in the covering relation of the learning problem. The abductive logic programs learned in ACL contain both rules for the concept(s) to be learned as well as general clausal theories called integrity constraints. These two parts are synthesized together in a non-trivial way via the abductive reasoning of ALP which is then used as the basic covering relation for learning. In this way, learning in ACL synthesizes discriminant and characteristic induction to learn abductive logic programs.

This paper presents the basic framework of ACL with its main characteristics and demonstrates its potential for addressing the problem of learning from an incomplete background knowledge and of classifying new cases that again could be incompletely specified.

An algorithm is presented that performs a simpler version of ACL called Intermediate ACL (I-ACL): new rules are learned but not new integrity constraints and learned programs must satisfy a weaker condition on negative examples.

Integrity constraints can be learned externally to the I-ACL to solve the full ACL problem. The algorithm has been implemented in a new system also called I-ACL that constitutes the first building block of a system for performing full ACL. Integrity constraints are learned externally to this system using the Claudien [12] system together with additional data generated by the I-ACL system when this has finished. Several initial experiments are presented that demonstrate the ability of I-ACL and ACL to learn with incomplete information.

2 Abductive Logic Programming

In this section we very briefly present some of the elements of Abductive Logic Programming (ALP) needed for the formulation of the learning framework of Abductive Concept Learning (ACL). For more details the reader is referred to the survey [8] and the technical report [10].

Definition 1. An **abductive theory** T in ALP is a triple $\langle P, A, I \rangle$, where P is a normal logic program, A is the set of predicates called *abducible predicates*, and I is a set of first-order closed formulae called *integrity constraints*[3].

The semantics of an abductive theory is formalized through the notion of a Generalized Stable Model (for details see [8]). With this we define the notion of an abductive explanation and abductive entailment as follows. Given an abductive theory $T = \langle P, A, I \rangle$ and a formula G, the goal of abduction is to find a set of ground atoms Δ (*abductive explanation*) on the set of predicates in A which together with P entails G, i.e. $P \cup \Delta \models G$. It is also required that the program $P \cup \Delta$ is consistent with respect to I, i.e. $P \cup \Delta \models I$. When there exists an abductive explanation for e from T we say that T *abductively entails* e and we write $T \models_A e$.

These definitions can be extended (see [10]) to allow in Δ negative abducible assumptions for the falsity (or absence) of an abducible assumption. Such negative assumptions will be denoted by not_p where p is an abducible.

3 Learning with Abduction

Abductive Concept Learning can be defined as a case of discriminant concept learning but which contains in it also a characteristic learning problem. It thus combines discriminant and characteristic learning in a non-trivial way using the abductive entailment relation as the covers relation for the training examples.

The language of hypotheses is that of Abductive Logic programming as is also the language of background knowledge. The language of the examples is simply that of atomic ground facts.

[3] In practice, integrity constraints are restricted to be range-restricted clauses.

Definition 2. Abductive Concept Learning (ACL)
Given

- a set of positive examples E^+ of a concept C,
- a set of negative examples E^- of a concept C,
- an abductive theory $T = \langle P, A, I \rangle$ as background theory.

Find
A new abductive theory $T' = \langle P', A, I' \rangle$ with $P' \supseteq P$ and $I' \supseteq I$, such that

- $T' \models_A E^+$,
- $\forall e^- \in E^-, T' \not\models_A e^-$.

Therefore, ACL differs from ILP because both the background knowledge and the learned program are abductive logic programs. As a consequence, the notion of deductive entailment of ILP is substituted with the notion of abductive entailment ($\models_A$) in ALP.

Note that whereas for positive examples it is sufficient for the learned theory to provide an explanation of these, for the negative examples there should not exist a single abductive explanation in the learned theory that covers any of the negative examples.

It is often useful for practical reasons to initially relax this strong requirement on the negative examples and consider an **intermediate** problem, denoted by **I-ACL**, where the negative examples are also only required to be credulously uncovered i.e. that there exits in the learned theory at least one abductive explanation where the negative examples can not be covered (and of course in this extension it is also possible to cover the positive examples). Moreover, in this intermediate problem we are not interested in learning new integrity constraints. To this end we define the intermediate problem where the two conditions in the above definition are replaced by:
Find
A new abductive theory $T' = \langle P', A, I \rangle$ with $P' \supseteq P$, such that

- $T' \models_A E^+ \cup not\_E^-$, where $not\_E^- = \{not\_e^- | e^- \in E^-\}$.

where we use the notation $T' \models_A not\_e^-$ to mean that there exists at least one abductive sets of hypotheses such that, when added to P', then e^- would fail. We can then solve first this simpler problem and then use the explanations for the examples as input for further learning in order to satisfy the full ACL problem. The following example illustrate how it is possible to solve an ACL problem by first solving an I-ACL problem and then learning the integrity constraints.

Example 1. Suppose we want to learn the concept $father$. Let the background theory be $T = \langle P, A, \emptyset \rangle$ where:
$P = \{parent(john, mary), male(john),$
$parent(david, steve),$
$parent(kathy, ellen), female(kathy)\}$

$A = \{male/1, female/1\}$
and let the training data be:
$E^+ = \{father(john, mary), father(david, steve)\}$
$E^- = \{father(kathy, ellen), father(john, steve)\}$
In this case, a possible hypotheses $T' = \langle P', A, I \rangle$ learned by I-ACL would contain in P' the rule
$father(X, Y) \leftarrow parent(X, Y), male(X).$
under the abductive assumptions
$\Delta = \{male(david), not\_male(kathy)\}.$
The positive assumption comes from the requirement to cover the positive examples while the negative assumption from the requirement to cover the default negation of negative examples. In addition, by considering the background knowledge together with the positive assumption $male(david)$ from Δ, we could learn (using characteristic induction) the integrity constraint:
$\leftarrow male(X), female(X).$
Note that this integrity constraint provides independent support for the negative assumption $not\_male(kathy)$ in Δ thus ensuring that the corresponding negative example can not be abductively covered by the learned hypothesis. Therefore, the final theory containing the rule and the constraint satisfies the full ACL problem definition.

4 Algorithm

In this section we present an algorithm for I-ACL. This algorithm is based on the generic top-down algorithm (see e.g. [11]) suitably adapted to deal with the incompleteness of the abducible predicates and to take into account the integrity constraints in the background theory. It incorporates (and adapts) algorithms for abductive reasoning from ALP [9], extending the algorithm in [6]. We will not consider here fully the problem of integrating the learning of integrity constraints in this algorithm but rather we will assume that these exist (or have been learned) in the background theory. For lack of space, we outline a version of I-ACL for the case of single predicate learning, however I-ACL is able to perform effectively multiple predicate learning (see the technical report [10] for this extension).

The I-ACL algorithm differs from the basic top-down algorithm in the following respects. In the specialization loop, instead of a greedy search in the space of possible clauses, a best-first search is performed using an heuristic evaluation function that will be defined below.

The evaluation of a clause is done by starting and abductive derivation for each e^+ and each not_e^-, using the procedure defined in [9]. This procedure takes as input the goal to be derived, the abductive theory, the set of previous assumptions and, if it succeeds, produces as output the set of assumptions extended with the atoms abduced during the derivation. The set of assumptions abduced for earlier examples is also considered as input to ensure that the assumptions made during the derivation of the current example are consistent with the ones made before. Thus, we can test each positive and negative example separately and be sure that the clause will abductively derive $e_1^+ \wedge \ldots \wedge e_n^+ \wedge not\_e_1^- \wedge \ldots \wedge not\_e_m^-$.

The heuristic function of a clause is an *expected classification accuracy* [11] in which we need to take into account the relative strength of covering an example with abduction (i.e. some assumptions are needed and the strength of those assumptions) or without assumption (i.e. no assumption is needed) and similarly for the failure to cover negative examples with or without assumptions. The heuristic function used is

$$A = \frac{n^{\oplus} + k^{\oplus} \times n_A^{\oplus}}{n^{\oplus} + n^{\ominus} + k^{\oplus} \times n_A^{\oplus} + k^{\ominus} \times n_A^{\ominus}}$$

where $n^{\oplus}, n_A^{\oplus}, n^{\ominus}, n_A^{\ominus}$ are:

$n^{\oplus}$ number of pos. ex. covered by the clause without abduction,
$n_A^{\oplus}$ number of pos. ex. covered by the clause with abduction,
$n^{\ominus}$ number of neg. ex. covered by the clause (not_e^- has failed),
$n_A^{\ominus}$ number of neg. ex. uncovered by the clause with abduction.

The coefficients $k^{\oplus}$ and $k^{\ominus}$ are introduced in order to take into account the uncertainty in the coverage of the positive examples and in the failure to cover the negative examples when abduction is necessary for this to occur. Further details on these coefficients and the heuristic function can be found in [10].

The algorithm described above is sound but not complete for the intermediate problem I-ACL. Its soundness is ensured by the soundness of the abductive proof procedure and its extensions developed for the learning algorithm.

The algorithm is not complete because the search space is not completely explored. In particular, there are two choice points which are not considered in order to reduce the computational complexity of the algorithm. The first choice point is related to the greedy search in the space of possible programs (clauses are never retracted). The second choice point concerns the abductions that are made in order to cover the examples. In fact, an example can be abductively derived with different sets of assumptions. Depending on which set of assumptions we choose, this can change the outcome of the algorithm later on.

5 Experiments

We have performed several experiments to test the ability of I-ACL and ACL to learn from incomplete data. One set of experiments centered around the problem of learning family relations under different degree of incompleteness in the background knowledge. The I-ACL system was able to learn correctly from data which was only 80 % complete (20 % of the data was randomly removed) and, when integrity constraints were introduced, the system learned correctly with the missing data approaching 40 % of the initially complete data. Also multiple predicate learning experiments were carried out. The I-ACL system was able to handle this problem using the abductive data generated for one predicate while learning another predicate.

More details on these experiments can be found in [10]. Here we describe in some detail a series of experiments on the multiplexer problem [13] and compare the results of I-ACL with those of ICL-Sat.

5.1 Multiplexer

The multiplexer problem consists in learning the definition of a 6 bits multiplexer, starting from a training set where each example is composed by 6 bits, where the first two bits are interpreted as the address of one of the other four bits. If the bit at the specified address is 1 (regardless of the values of the other three bits), the example is considered positive, otherwise it is considered negative. For example, consider the example 10 0110. The first two bits specify that the third bit should be at 1, so this example is positive. We represent it by adding the positive example $mul(e1)$ to the training set and by including the following facts in the background knowledge

$bit1at1(e1)$. $bit2at0(e1)$. $bit3at0(e1)$.

$bit4at1(e1)$. $bit5at1(e1)$. $bit6at0(e1)$.

For the 6-bit multiplexer problem we have $2^6 = 64$ examples, 32 positive and 32 negative. We performed three experiments using the same datasets as in [13]: the first on the complete dataset, the second on an incomplete dataset and the third on the incomplete dataset plus some integrity constraints. The incomplete dataset was obtained by considering 12 examples out of 64 and by specifying for them only three bits where both the examples and bits were selected at random. E.g. the above example 10 0110 could have been replaced by 1? 0?1?. The incomplete example is still in the set of positive examples and its description in the background knowledge would be

$bit1at1(e1)$. $bit3at0(e1)$. $bit5at1(e1)$.

Now, all the predicates $bitNatB$ are abducibles and integrity constraints of the form below are added to the background theory

$\leftarrow bitNat0(X), bitNat1(X)$.

The dataset of the third experiment is obtained by including additional integrity constraints to the incomplete dataset. For each of the incomplete examples, an attempt was made to add constraints so that 1) the value of unknown attributes was still unknown (could still be 1 or 0); 2) some combination of values incompatible with the known class was now made impossible.

I-ACL was run on all the three datasets. The measure of performance that was adopted is accuracy, defined as the number of examples correctly classified over the total number of examples in the testing set, i.e. the number of positive examples covered plus the number of negative examples not covered over 64. The theory was tested on complete examples. The result are summarized in the following table:

Experiments	I-ACL	ICL- Sat
Complete background	100 %	100 %
Incomplete background	98.4 %	82.8 %
Incomplete background plus constraints	96.9 %	92.2 %

The theories learned by I-ACL were also tested on the incomplete examples and other randomly generated incomplete testing data to see how these could classify incomplete examples. The accuracy of this classification remained at approximately the same level as with the complete testing examples shown above.

Finally, we performed an experiment where more information was taken out. In this case we have taken out completely from the training data some of the negative examples. The I-ACL system learned rules that are incorrect (i.e. cover negative examples) as they are too general, e.g. the rule:

$mul(X) \leftarrow bit1at1(X), bit2at1(X).$

Extending the I-ACL algorithm, we run Claudien to learn clauses (integrity constraints) that hold on the positive examples. In this way we generate several constraints amongst which is

$\leftarrow mul(X), bit1at1(X), bit2at1(X), bit6at0(X)$

which effectively corrects the rule above by preventing the negative examples to be abductively classified as positive even when the rule succeeds. This experiment illustrates the potential of ACL to integrate characteristic with discriminant induction.

6 Conclusions and Related Work

We have studied the new learning framework of Abductive Concept Learning developing a first system for an intermediate version of it, I-ACL. Initial experiments with this system have demonstrated I-ACL as a suitable framework for learning with incomplete information.

Abduction has been incorporated in many learning systems (see e.g. [5, 2, 1]) but in most cases this is seen as a useful mechanism that can support some of the activities of the learning systems. For example, in multistrategy learning (or theory revision) abduction is identified as one of the basic computational mechanisms (revision operators) for the overall learning process. A notable exception is that of [15] where a simple form of abduction is used as the covering relation in the context of a particular application of learning theories for diagnosis. Also recently, the deeper relationship between abduction and induction has been the topic of study of an ECAI96 workshop [3]. Our work builds on earlier work in [5] and [6] for learning simpler forms of abductive theories. Finally, the issue of integrating discriminant (or explanatory) and characterizing ILP systems has also been put forward in [4].

Recently, there have been several other proposals for learning with incomplete information. The FOIL-I system [7] learns from incomplete information in the training examples set but not in the background knowledge. In [16] the authors propose several frameworks for learning from partial interpretations. A framework that can learn form incomplete information and is closely related to ACL is that of learning from satisfiability [13].

Further development of the I-ACL system is needed to integrate in it the characteristic induction process of learning integrity constraints and thus solving directly the full ACL problem. An interesting approach for this is to use existing

ILP systems for characteristic learning such as Claudien and Claudien-Sat, but also discriminant systems like ICL [14] and ICL-Sat for learning integrity constraints that discriminate between positive and negative abductive assumptions generated by ACL. We have also began to investigate the application of I-ACL to real-life problems in the area of analyzing market research questionnaires where incompleteness of information occurs naturally by unanswered questions or do not know and undecided answers.

References

1. H. Adé and M. Denecker. RUTH: An ILP theory revision system. In *Proceedings of the 8th International Symposium on Methodologies for Intelligent Systems*, 1994.
2. H. Adé and M. Denecker. AILP: Abductive inductive logic programming. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, 1995.
3. M. Denecker, L. De Raedt, P. Flach, and A. Kakas, editors. *Proceedings of ECAI96 Workshop on Abductive and Inductive Reasoning.* Catholic University of Leuven, 1996.
4. Y. Dimopoulos, S. Džeroski, and A.C. Kakas. Integrating explanatory and descriptive learning in ILP. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence*, 1997.
5. Y. Dimopoulos and A. Kakas. Abduction and inductive learning. In L. De Raedt, editor, *Advances in Inductive Logic Programming*, pages 144–171. IOS Press, 1996.
6. F. Esposito, E. Lamma, D. Malerba, P. Mello, M. Milano, F. Riguzzi, and G. Semeraro. Learning abductive logic programs. In Denecker et al. [3].
7. N. Inuzuka, M. Kamo, N. Ishii, H. Seki, and H. Itoh. Top-down induction of logic programs from incomplete samples. In S. Muggleton, editor, *Proceedings of the 6th International Workshop on Inductive Logic Programming*, pages 119–136. Stockholm University, Royal Institute of Technology, 1996.
8. A.C. Kakas, R.A. Kowalski, and F. Toni. Abductive logic programming. *Journal of Logic and Computation*, 2:719–770, 1993.
9. A.C. Kakas and P. Mancarella. On the relation between truth maintenance and abduction. In *Proceedings of the 2nd Pacific Rim International Conference on Artificial Intelligence*, 1990.
10. A.C. Kakas and F. Riguzzi. Learning with abduction. Technical Report TR-96-15, University of Cyprus, Computer Science Department, 1996.
11. N. Lavrač and S. Džeroski. *Inductive Logic Programming: Techniques and Applications.* Ellis Horwood, 1994.
12. L. De Raedt and M. Bruynooghe. A theory of clausal discovery. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, 1993.
13. L. De Raedt and L. Dehaspe. Learning from satisfiability. Technical report, Katholieke Universiteit Leuven, 1996.
14. L. De Raedt and W. Van Lear. Inductive constraint logic. In *Proceedings of the 5th International Workshop on Algorithmic Learning Theory*, 1995.
15. C. Thompson and R. Mooney. Inductive learning for abductive diagnosis. In *Proceedings of the 12th National Conference on Artificial Intelligence*, 1994.
16. S. Wroble and S. Džeroski. The ILP description learning problem: Towards a genearl model-leve definition of data mining in ILP. In *Proceedings of the Fachgruppentreffen Maschinelles Lernen*, 1995.