Learning Abductive Theories

Yannis Dimopoulos and Antonis Kakas[1]

Abstract. In this paper we study the problem of learning abductive theories with particular interest in learning theories for the problem of attribute-based classification as studied in the area of machine learning. The paper proposes a new alternative formulation of this class of learning problems where abduction takes an integral part in the formulation of the appropriate theories and more importantly in the definition of the learning problem itself. We present a general algorithm that learns abductive theories for classification and examine its main features. We show how within our abductive approach it is possible to formulate and handle in a natural way cases of the problem with incomplete information. We also study the relation of our approach to other existing approaches for these learning problems, notably that of decision trees, and argue that our approach could provide a useful link between abduction and machine learning.

1 INTRODUCTION

Over the last decade many authors have pointed out the importance of abductive reasoning in problems of Artificial Intelligence and other areas of Computer Science (see the surveys [5], [9] and references therein). Abduction has been shown to be a useful form of reasoning for many applications ranging from diagnosis and planning to database updates and user modeling. Although recently there has been some work trying to relate abduction and learning (or induction) there has been very little work in studying the problem of learning abductive theories i.e. theories whose main form of reasoning is abduction suited for the kind of applications mentioned above.

In this paper we will address this problem of learning abductive theories by studying the special, but characteristic case, of learning abductive theories for classification (or diagnosis). In particular, we will investigate a simplified case of this problem that is essentially the same problem as that of attribute-based learning (or discrimination) studied in Machine Learning. From the point of view of learning, this paper aims to propose a new approach for learning classification theories based on an alternative abductive perspective in which the learning problem itself is formalized in terms of abduction rather than deduction as it is usually done in conventional learning theories. We will also provide a basic computational model for such learning problems based on computing suitable abductive theories and compare this to the well-known method of decision trees.

Intuitively, within this abductive perspective the classification of an object is understood as a task of best explaining the (given) features of the object by assuming that the object belongs to a certain class (or concept). Therefore in the learning phase we need to construct (abductive) theories which can provide abductive explanations of the features in terms of concepts. Instead of learning rules of the form $concept \leftarrow features$ that can be used deductively to infer (classify) concepts from their features (attributes) we will be learning rules of the form $feature \leftarrow concept$ that can be used abductively to explain the features.

These abductive theories will be triples $T =< T_A, A, IC >$, where T_A is a set of rules of the form $feature \leftarrow concept$, A -the set of abducible predicates- will contain the concept predicates that we are learning to discriminate among and IC is a set of integrity constraints. Hence the abducible predicates, which usually capture the incompleteness of the problem, are the concepts that we are trying to learn and an abductive theory learns these concepts when it is able to explain correctly (typical) sets of features in terms of their corresponding concepts. A formula ϕ is *abductively explained or entailed* by an abductive theory T iff there exists H such that $T_A \cup H \models \phi$ with $H \subseteq A$, provided that H does not violate any of the constraints in IC. Coverage of a training example e for the concept c is defined accordingly by saying that a theory T covers "e" if the hypothesis $H = \{c(e)\}$ is an abductive explanation of the given features of "e".

Let us illustrate these ideas by means of an example. Consider the simple situation where we want to learn (to discriminate) the concepts "$bird$" and "$plane$" from their features. Our intention is to construct a theory like $T_1 =< T_A, CP, IC >$, where T_A:

$flies(X) \leftarrow bird(X)$	$flies(X) \leftarrow plane(X)$
$wings(X) \leftarrow bird(X)$	$wings(X) \leftarrow plane(X)$
$feathers(X) \leftarrow bird(X)$	$metal(X) \leftarrow plane(X)$

and the set of abducibles is $CP = \{bird, plane\}$. Existing learning approaches could construct a theory T_2 composed of the rules

$bird(X) \leftarrow flies(X), wings(X), feathers(X)$

$plane(X) \leftarrow flies(X), wings(X), metal(X)$

A new object "Tweety" described by the set of observations (features) $O = \{flies(Tweety), feathers(Tweety)\}$ will be classified as a bird by T_1 since $bird(Tweety)$ abductively explains O. On the other hand the object "Eagle" with the features $O' = \{flies(Eagle), wings(Eagle)\}$ will be classified either as a bird or as a plane.

A main characteristic of abductive theories like T_1 is that we can reason under *incomplete information*. In the previous example even though we have no information about attribute $wings$ we can still classify Tweety as a bird. Similarly, for the example of Eagle but here this results in a disjunctive classification.

Note also that classification within the abductive theories framework is a *goal-directed* reasoning task. In the above example the classification of $Tweety$ as a $bird$ is obtained by the goal-directed process of explaining its given features. Deductive theories like T_2 that define the concepts by $concept \leftarrow features$ rules, in general require forward reasoning with all of these rules to see which one is satisfied in order to arrive at a classification.

The rest of the paper is organized as follows. In section 2 we present the language and basic notions of abductive theories needed to formulate classification problems. Then in section 3 we formalize the

[1] Department of Computer Science, University of Cyprus, CY-1678, Nicosia, Cyprus. Email: {yannis, antonis}@turing.cs.ucy.ac.cy

ECAI 96. *12th European Conference on Artificial Intelligence*
Edited by W. Wahlster
Published in 1996 by John Wiley & Sons, Ltd.

learning problem in terms of abduction and present a basic algorithm for learning abductive theories for classification problems. In section 4 we compare our framework with one of the main approaches to such learning problems namely that of decision trees. Finally, in section 5 we discuss some related work and possible directions for further development of the approach introduced in this paper.

2 ABDUCTIVE THEORIES

In this section we show how the problem of attribute-based classification can be formulated through abductive theories and set up the basic background notions needed for the learning problem. We assume a language that contains predicates of the form $attr_i(X, Y)$ that are used to denote that the value of the attribute A_i for object X is Y. Moreover, the language contains a set of concept predicates $C_j(X)$ denoting that an object X belongs to concept C_j. From this set of symbols we can construct *abductive rules* of the form $attr_i(X, Y) \leftarrow C_j(X)$. Intuitevely, such a rule states that for an object X that belongs to the concept C_j the value Y for A_i is valid. In the following we reserve the term *abductive theory* to refer to a triple of the form $< T_A, A, IC >$, where T_A is a set of abductive rules, A is the set of abducibles which coincides with the set of concept predicates C_j, and IC is a set of integrity constraints.

We will also assume that the values of the attributes may define a hierarchical structure. For instance, the values $black$ and $green$, for the attribute $color$ are instances of the more general value $dark\text{-}color$, while the values $white$ and $yellow$ are instances of the more general value $bright\text{-}color$. This hierarchy defines a *subsumption relation* between the values of the attributes. We say that the value $dark\text{-}color$ *subsumes* the value $green$, for the attribute $color$. We assume that the subsumption relation is reflexive and transitive, and that the distinguished value, *any*, for an attribute subsumes any other value for the same attribute. We use the notation $v < v'$ to denote that v' subsumes v. We say that a rule $attr_i(X, v) \leftarrow C_j(X)$ subsumes a value v' if $v' < v$ holds.

Abductive theories provide a framework for classification of attribute-value tuples as instances of one (or more) of the concepts defined in the theory. An attribute-value tuple is a configuration of the form $< obj, (A_1, v_1), \ldots, (A_n, v_n) >$ where v_i denotes the value of the attribute A_i for the object obj to which the tuple refers. A classification for an attribute-value tuple is an association of the corresponding object to a concept that provides an explanation for the values of all the attributes in the tuple. A particular notion of explanation that is appropriate for this task is that of the *generalized explanation*.

Definition 1 *(*Generalized explanation*)*
We say that $C(obj)$ is a generalized explanation for the tuple $< obj, (A_1, v_1), \ldots, (A_n, v_n) >$ in the abductive theory $T = < T_A, A, IC >$, iff for every attribute A_i, $T_A \cup C(obj) \models attr_i(obj, val_i)$ with $v_i < val_i$, and $T_A \cup C(obj)$ satisfies the constraints in IC.

The subsumption relation between the values of the attributes provides a natural notion of *preferred explanations*. Informally, we prefer concepts that can explain a given tuple with values that are lower in the value hierarchy, i.e. closer to the given values of the tuple. The notion of preferred explanation is defined formally as follows.

Definition 2 *(*Preferred generalized explanation*)*
We say that $C(obj)$ is a preferred generalized explanation for the tuple $< obj, (A_1, v_1), \ldots, (A_n, v_n) >$ in the abductive theory $T = < T_A, A, IC >$, iff it is a generalized explanation for the tuple and there is no other generalized explanation $C'(obj)$ such that for every attribute A_i with $T_A \cup C(obj) \models attr_i(obj, v)$ and $T_A \cup C'(obj) \models attr_i(obj, v')$, $v' < v$ holds.

The following example illustrates the classification problem within the abductive theories framework.

Example 3 *Consider the following set of abductive rules T_A, that contains the definitions of four concepts C_1, C_2, C_3 and C_4.*

$color(X, black) \leftarrow C_1(X)$	$color(X, dark) \leftarrow C_2(X)$
$size(X, big) \leftarrow C_1(X)$	$size(X, big) \leftarrow C_2(X)$
$shape(X, triangle) \leftarrow C_1(X)$	$shape(X, square) \leftarrow C_2(X)$
$color(X, yellow) \leftarrow C_3(X)$	$color(X, bright) \leftarrow C_4(X)$
$size(X, small) \leftarrow C_3(X)$	$size(X, small) \leftarrow C_4(X)$
$shape(X, polygon) \leftarrow C_3(X)$	$shape(X, square) \leftarrow C_4(X)$

Assume that the values for the attributes are related through the subsumption relation as follows: $black < dark$, $blue < dark$, $yellow < bright$, $triangle < polygon$, $square < polygon$.

The tuple $< obj_1, (color, blue), (size, big) >$ for object obj_1 has a unique generalized explanation in T, namely $C_2(obj_1)$. Hence, obj_1 is classified as an instance of concept C_2. Note again that if the definition of C_2 was represented by the rule $C_2(X) \leftarrow color(X, dark)$, $size(X, big)$, $shape(X, square)$ and we used deduction as the reasoning method, then $C_2(obj_1)$ would not follow since the value of attribute $shape$ is unknown. Assume now that we want to classify the tuple $< obj_2, (color, black), (size, big) >$. Notice that there are two generalized explanation for this tuple in T, namely $C_1(obj_2)$ and $C_2(obj_2)$. However, the explanation $C_1(obj_2)$ is preferred over $C_2(obj_2)$. Therefore, under the preferred explanation semantics, obj_2 will be classified as a C_1 object, since C_1 is the unique preferred explanation. However, it is not the case that a unique preferred generalized explanation always exists. Consider for example the tuple $< obj_3, (color, yellow), shape(square) >$. There are two preferred explanations for it, namely $C_3(obj_3)$ and $C_4(obj_3)$. This feature of "disjunctive answers" is an essential difference between the deductive and the abductive model for classification in cases of incomplete information. While in the deductive formulation classification is not possible the abductive reasoning method copes with this problem by computing a set of plausible classifications.

3 THE LEARNING PROBLEM

In this section we address the problem of learning abductive classification theories as defined in the previous section. We will first concentrate on learning a theory for a single concept and later show how this can be extended to cover the case of multiple concepts. Throughout this section we assume a fixed set of attributes $\mathcal{A} = \{A_1, \ldots, A_n\}$. The input of the learning process is a set of positive and negative examples which are pairs of the form $E_i = (C(obj_i), AV_i)$ where C is the concept to be learned, obj_i is the particular object to which the example refers, and AV_i is a tuple of attribute-value pairs of the form $< (A_1, v_{i1}), \ldots, (A_n, v_{in}) >$ with v_{ij} the value of the attribute A_j for the object obj_i. The output will be an abductive theory $T =< T_A, \{C\}, IC >$.
Formally, the abductive theory learning problem is defined as follows.

Definition 4 *(*abductive learning problem*)*
Given

- *A set of positive examples $E_1^+ = (C(pobj_1), PAV_1)), \ldots, E_n^+ = (C(pobj_n), PAV_n))$*
- *A set of negative examples $E_1^- = (C(nobj_1), NAV_1)), \ldots, E_m^- = (C(nobj_m), NAV_m))$*

Find
An abductive theory $T = < T_A, \{C\}, IC >$ such that for every positive example E_k^+, $C(pobj_k)$ is a generalized explanation for PAV_k (completeness) and for every negative example E_k^-, $C(nobj_k)$ is not a generalized explanation for NAV_k (consistency).

An algorithm that learns correctly abductive theories as required by definition 4 (i.e. the learned theory is complete and consistent wrt the training examples) is the following.

Algorithm Learn-Abductive-Theory
For every attribute A_i in $\mathcal{A}$ with values $v_{i1}, \ldots v_{in}$ in the n positive examples add to T the rule $attr_i(X, val) \leftarrow C(X)$, where val is the *minimal* value that subsumes each of the values $v_{i1}, \ldots v_{in}$;
While $C(obj)$ is a generalized explanation for a negative example $E^-(C(obj), NAV)$ **do**

1. Choose an attribute A_i and find the set R containing rules $attr_i(X, val) \leftarrow C(X)$ for which the value v' of the attribute A_i in the negative example is subsumed by val.
2. Delete from T every rule of R and compute the set V of values for the attribute A_i in the positive examples that the rules of R subsume and are not subsumed by any other in T.
3. Add the minimal set of rules $attr_i(X, val_{i1}) \leftarrow C(X), \ldots, attr_i(X, val_{ik}) \leftarrow C(X)$ such that for every value in V there exists a value val_p that subsumes it, and for the value v' of the attribute A_i in any of the negative examples, there is no value val_q that subsumes v'.

Notice that the third step of the algorithm specializes an attribute by effectively replacing its overgeneral value by a subset of values that are lower in the subsumption hierarchy. This is an indication for the need of a disjunctive concept definition. In some cases though this splitting of step 3 alone is not sufficient to capture the disjunctive concept. For this reason we introduce an extra operation, called *grouping* which chooses certain combinations of the rules generated by the splitting of step 3 and explicitly groups them as different cases of the concept definition. Grouping can be applied at some cases of the specialization step 3 or after every such step. We then effectively replace step 3 of the algorithm by a new step that includes grouping:

$3'$. Compute the minimal set of rules $attr_i(X, val_{i1}) \leftarrow C(X), \ldots, attr_i(X, val_{ik}) \leftarrow C(X)$ as in step 3.
To each one of these rules add a copy of all the other rules that contain C in their body and replace C by a new symbol C^m, $(1 \leq m \leq k)$. These C^m denote the different cases of C.

The introduction of the different cases of a concept can be easily accommodated in the framework by extending the definitions of the previous section in the obvious way. For instance, $C(obj)$ is a generalized explanation for an object obj iff one of $C^1, \ldots, C^k$ (the cases of C) is a generalized explanation for obj. We also note here that with step $3'$ the algorithm can be closely related to the basic algorithm of decisions trees. This issue will be discussed in the next section.
To illustrate the basic learning algorithm and the effect of grouping consider the following example.

Example 5 *Let $E_1^+ = (C(obj_1), < (color, black), (size, big), (shape, pentagon) >)$ and $E_2^+ = (C(obj_2), < (color, blue), (size, big), (shape, square) >)$ be positive examples of a concept C with the following relations between the attribute values: $black < dark$, $blue < dark$, $green < dark$, $triangle < polygon$, $square < polygon$, $pentagon < polygon$.
Then, the algorithm will compute an abductive theory with T_A:*
$color(X, dark) \leftarrow C(X)$
$size(X, big) \leftarrow C(X)$
$shape(X, polygon) \leftarrow C(X)$
Assume now that we have the negative examples $E_1^- = (C(obj_3), < (color, green), (size, big), (shape, triangle) >)$ and $E_2^- = (C(obj_4), < (color, green), (size, big), (shape, pentagon) >)$. These are covered (abductively entailed) by C in T_A and hence specialization is needed. Suppose that the algorithm chooses the attribute $color$ for specialization. This will result in a new set of rules T_A^1:
$color(X, black) \leftarrow C(X)$ $\quad$ $color(X, blue) \leftarrow C(X)$
$size(X, big) \leftarrow C(X)$ $\quad$ $shape(X, poly) \leftarrow C(X)$
If we also use grouping we will get the set of rules T_A^2:
$color(X, black) \leftarrow C^1(X)$ $\quad$ $color(X, blue) \leftarrow C^2(X)$
$size(X, big) \leftarrow C^1(X)$ $\quad$ $size(X, big) \leftarrow C^2(X)$
$shape(X, poly) \leftarrow C^1(X)$ $\quad$ $shape(X, poly) \leftarrow C^2(X)$
Notice that up to this point the two theories T_A^1 and T_A^2 are equivalent. Now suppose that another negative example $E_2^- = (C(obj_4), < (color, blue), (size, big), (shape, triangle) >)$ is given and assume that the algorithm chooses the attribute $shape$ for specialization. With step 3 alone T_A^1 will transform to:
$color(X, black) \leftarrow C(X)$ $\quad$ $color(X, blue) \leftarrow C(X)$
$shape(X, square) \leftarrow C(X)$ $\quad$ $shape(X, pentagon) \leftarrow C(X)$
$size(X, big) \leftarrow C(X)$
On the other hand with step $3'$, T_A^2 will be replaced by:
$color(X, black) \leftarrow C^1(X)$ $\quad$ $color(X, blue) \leftarrow C^{21}(X)$
$size(X, big) \leftarrow C^1(X)$ $\quad$ $size(X, big) \leftarrow C^{21}(X)$
$shape(X, poly) \leftarrow C^1(X)$ $\quad$ $shape(X, square) \leftarrow C^{21}(X)$
$color(X, blue) \leftarrow C^{22}(X)$
$size(X, big) \leftarrow C^{22}(X)$
$shape(X, pentagon) \leftarrow C^{22}(X)$
Notice that the theory T_A^2 is more general than T_A^1. This is because in T_A^2 we have restricted the necessary specialization due to the third negative example to a part of the theory, while in T_A^1 specialization has a more global effect. As a consequence the object obj with attributes $< (color, black), (size, big), (shape, triangle) >$ will be classified as an instance of the concept C by T_A^2 but not by T_A^1.

Note that the nondeterministic choice of an attribute for specialization in the first step of the loop of the algorithm is important for the quality of the learned theory. The situation is analogous to decision trees which use "goodness" criteria to define the splits of the tree. Some of these could be adapted for use in our case.

Learning in an abductive framework offers the possibility of specializing a theory by inducing integrity constraints. If a negative example for a concept C has a generalized explanation in a theory T, keep the overgeneral rules in T, choose an attribute A_i and add the integrity constraint $\neg(C(X), attr_i(X, v))$ where v is the value of attribute A_i in the negative example. The effect of this constraint is to block the assumption $C(X)$ for the objects for which the value v is known for A_i. In this way we can express *exceptions* to the definition of the concept C. In the context of the previous example we could keep the set of abductive rules T_A unchanged and add in the set of integrity constraints IC the constraint $\neg(C(X), color(X, green))$ to handle the first two negative examples.

Let us elaborate a little further on the addition of integrity constraints to the theory when negative information is available. First, we note that we may want to add to the denial of the integrity constraint all the attributes which are known to be true for the individuals involved in the negative example. This will lead to the least specialization of the learned theory. For instance, in the previous example we may want to add the constraint $\neg(C(X), color(X, green), size(X, big), shape(X, polygon))$. This will then allow for the possibility of classifying $< obj, (color(X, green), size(X, big) >$ as an instance of C.

Assume now that several negative examples for a concept are given. We can either handle each negative example in isolation and create an integrity constraint for each of them, or synthesize and generalize the information in the negative examples like we do for positive examples. Consider the next example.

Example 6 *Assume the following hierarchy of values for the attribute place: US < $America$, $Canada$ < $America$, $Japan$ < $Asia$, $China$ < $Asia$, UK < $Europe$, $France$ < $Europe$, $Germany$ < $Europe$. Moreover let $E_1^+ = (C(obj_1), < (place, US) >)$ and $E_2^+ = (C(obj_2), < (place, China) >)$ be two positive examples for the concept C. The learning algorithm will induce the rule $place(X, any) \leftarrow C(X)$ which means that any value for the attribute place is valid. Assume that latter we receive the negative information $E_1^- = (C(obj_3), < (place, UK) >)$ and $E_2^- = (C(obj_4), < (place, France) >)$. In order to specialize the induced rule we can add the two integrity constraints $\leftarrow C(X), place(X, UK)$ and $\leftarrow C(X), place(X, France)$. Instead we can go a step further and generalize these two constraints into the the integrity constraint $\leftarrow C(X), place(X, Europe)$. Notice that this last constraint will also block the classification of an object with attribute $(place, Germany)$ in the class C.*

Up to now we have concentrated on the problem of learning a single concept. When we have multiple concepts we can apply the same algorithm to learn rules for each concept, but now using preferred generalized explanations wherever we had used generalized explanations in the case of single concept. Using the preferred explanation is effectively a way of taking into account the background knowledge implicit in the hierarchy. Depending on the specific domain of a particular problem we may have additional preference criteria for explanations which we can incorporate in the notion of preferred explanation. Despite the use of preferred explanations in the specification of the learning problem it is still possible to have multiple (preferred) explanations when classifying some of the examples. In general there are two reasons why this occurs. The first is that some of the different concepts overlap. In this case the abductive framework offers a natural way to accommodate such overlapping definitions. This property was used in [12] for learning to diagnose under multiple disorders. If the concepts are not overlapping then multiple explanations indicate that the theory is overgeneral. In this case the *minimization* of the number of explanations can be a useful criterion for choosing among different abductive theories that can be learned.

When we know that some of the concepts are disjoint, we can reduce the number of explanations by augmenting the set of negative examples of each of these concepts with the set of positive examples of the other concepts.

The framework as it is presented so far treats all rules in a uniform way, irrespective of the fact that some of them may be learned from incompletely specified training examples. Consequently, we may want to reflect in some way in the learned theory that some rules have been induced by incomplete training data. We can for instance assign a confidence factor to every rule. This factor could be the proportion of examples with known values covered by this rule. Classification takes account of this information by preferring the explanation(s) with the highest overall confidence factor. Another possibility could be to impose an ordering on the rules that assigns a rule $attr_i(X, val) \leftarrow C_1(X)$ higher than the rule $attr_i(X, val) \leftarrow C_2(X)$ if the proportion of the training examples with known values for the attribute $attr_i$ for C_1 is higher (or much higher) than the same proportion for C_2. This problem of abductive reasoning with theories which include some ordering structure for preferring explanations has been extensively studied and some of these methods can be used here.

A problem that arises when we are learning from incomplete information is that the learned theory can be too specific on the attributes with missing values. For instance assume that we have induced the rule $color(X, dark) \leftarrow C(X)$ using the (possibly few) examples that had a value for the attribute $color$. This rule will not allow for any object with a non-dark color to be classified as a C object. To allow for this possibility we can include the additional rule $color(X, any) \leftarrow C(X)$ in the theory. This will give rise to multiple explanations replacing the problem of incomplete information by that of multiple explanations, but note that one of these explanations will be the correct classification. Also the use of preferred generalized explanation (definition 2) arising from the subsumption hierarchy will result in picking out explanations that concur with the values that have been induced from the examples with known values. Finally notice that if we use an ordering structure on the learned rules as described above then these additional rules should be included in this structure e.g. they could be assigned a weight propositional to the number of examples with unknown values.

4 RELATION TO DECISION TREES

In this section we discuss the relation of our approach to decision trees (DT) with emphasis on the problem of coping with incomplete information (missing values) in the classification. One main difference between the two approaches is in the reasoning methods they support. Classification with DT is essentially a deductive process. Roughly speaking, a DT can be considered as a theory T of deductive rules corresponding to the different branches of the tree.

As mentioned earlier, this deductive view of discrimination and classification has difficulties in handling incomplete information. To overcome these problems several techniques have been proposed with the aim of enabling DT classifiers to cope with unknown attribute values (see [10]). Such techniques include "filling in" the missing values or including in the tree special nodes that take care of unknown attribute values. In comparison to these extra-logical techniques our approach provides an alternative way to accommodate problems with missing information within the logical framework of abductive reasoning. For example, it offers the weaker form of disjunctive classification sitable for situations with incomplete information.

The difference in the ability to handle incomplete information also manifests itself in the different theories constructed by the two approaches. Unlike the DT construction method, which is in general purely top-down, the abductive learning algorithm first induces an overgeneral abductive theory and then specializes this theory in order to avoid coverage of negative examples. This initial theory can be seen as the least general generalization that covers the positive examples. On the other hand in the DT construction we choose only some of the attributes (disregarding the others) and effectively start from the value "any" for the chosen attributes.

Example 7 *Consider the following examples:*
$E_1^+ = (C_1(obj_1), < (color, black), (size, big) >)$
$E_2^+ = (C_1(obj_2), < (color, blue), (size, big) >)$
$E_3^+ = (C_2(obj_3), < (color, white), (size, small) >)$

$E_4^+ = (C_2(obj_4), < (color, yellow), (size, small) >)$
The abductive learning algorithm will induce the theory

$size(X, big) \leftarrow C_1(X) \qquad color(X, dark) \leftarrow C_1(X)$
$size(X, small) \leftarrow C_2(X) \qquad color(X, bright) \leftarrow C_2(X)$

On the other hand, a DT induction algorithm would generate a tree with a single internal node using the $size$ (or $color$) feature in its attempt to produce a minimal tree.

The DT is more compact, as it chooses to disregard some of the attributes. This means that information pertaining to such attributes becomes irrelevant. In contrast, in our abductive learning approach we hold more information in the theory and this can alter the classification of new objects. For example, the unseen object obj_5 with attributes $< (color, white), (size, big) >$ will be classified as an instance of concept C_1 by the decision tree. The same object will not be classified either as C_1 or C_2 by the abductive theory, since none of these concept can explain both attribute values of the object. Although by attempting to hold more information we compromise the principle of simplicity (Occam's razor) of the learned theories, this enhances the ability of the our framework to handle incomplete information. Again in the above example, if we are given an object with the incomplete description $< (color, red) >$ (i.e. no information about its size) the abductive theory will still be able to classify this as C_1. It is important to note that we are not giving up (entirely) the simplicity principle but instead we try to find a balance between simplicity of theories and loss of information that could be useful for classification under incomplete information. The abductive algorithm would still attempt to specialize the overgeneral initial theory in the simplest possible way.

Despite the above mentioned differences, ideas and methods from DT remain meaningful in the context of abductive theories learning. For example, a common point in the algorithms is the choice of an attribute to specialize in the abductive algorithm or to grow the tree in the DT method. The same criteria developed in the DT methods can also be used here to guide specialization. More importantly, the learning algorithm of the previous section with grouping at every step (step $3'$) can simulate the basic DT algorithm. In fact, if the background knowledge of the value subsumption hierarchy is empty the two algorithms are essentially identical. Thus the abductive framework can be seen as an extension of DT that can take into account background knowledge.

5 RELATED WORK AND CONCLUSIONS

Abduction has already been incorporated in a number of machine learning and theory revision systems (e.g. [1], [2], [3], [7], [8], [11], [13]) that use analytical methods to utilize the background knowledge. Most of this work is concerned with the problem of how to integrate abductive and inductive reasoning methods to enhance the learning capabilities of their frameworks. For a general discussion of this issue of integrating abduction and learning see [1], [6] and [4] and references therein.

To our knowledge the only case where abduction in machine learning has been used as in this paper, as an integral part of the definition of the learning problem, is [12]. The authors argue for the usefulness of abduction for learning to diagnose in situations where there are *multiple disorders* for the same set of symptoms. While their work is driven by the specific application our work tries to give a general study of this abductive approach to learning. Learning rules with the learned concept in their bodies can also be found in [7]. However these rules are not to be used in the classification process itself but rather to reason about the ramifications of a classification.

In this paper we have presented a new approach to learning classification theories based on an abductive formulation of both the reasoning required and the definition of the corresponding learning problem. This approach provides a novel link between the areas of abduction and learning that brings together the two fields and enables the exchange of useful ideas and techniques between the two areas. From the point of view of abduction it provides a way to learn abductive theories in which we can exploit existing techniques from machine learning e.g. methods and heuristics from decision trees. From the point of view of learning it provides an alternative and natural way to formulate and handle problems with incomplete (or noisy) information. Also its links with decision trees offers the possibility of extending this method with some forms of background knowledge and integrity constraints.

The theoretical framework set up in this paper needs to be developed further especially with its application to specific problems in order to test and refine these ideas, find good practical heuristics and more generally evaluate the effect of the different parameters in the framework. It is also important to study the problem of extending our approach to learn more general forms of abductive rules. One way to do this would be to assume more complex forms of background knowledge (than that of a subsumption hierarchy) and integrate within our method existing techniques from knowledge-intensive machine learning approaches. In general, more work is needed in order to fully utilize the link between abduction and learning offered by the approach proposed in this paper.

REFERENCES

[1] H. Ade and M. Denecker. Abductive Inductive Logic Programming. *IJCAI-95*.

[2] W. W. Cohen. Abductive Explanation-Based Learning: A Solution to the Multiple Inconsistent Explanation Problem. *Machine Learning*, 8:167-219, 1992.

[3] L. De Raedt. *Interactive Theory Revision: An Inductive Logic Programming Approach*. Academic Press, 1992.

[4] Y. Dimopoulos and A. Kakas. Abduction and Learning. In *Advances in Inductive Logic Programming*, L. De Raedt (ed.), IOS Press, 1996.

[5] A. C. Kakas, R. A. Kowlaski and F. Toni. Abductive Logic Programming. *Journal of Logic and Computation*, 2(6):719-770, 1993.

[6] R. S. Michalski. Inferential Theory of Learning as a Conceptual Basis for Multistrategy Learning. *Machine Learning*, 11:111-151, 1993.

[7] K. Morik, S. Wrobel, J. Kietz and W. Emde. *Knowledge Acquisition and Machine Learning: Theory, Methods and Applications*, Academic Press, 1993.

[8] D. Ourston and R. J. Mooney. Theory Refinement Combining Analytical and Empirical Methods. *Artificial Intelligence*, 66:311–344, 1994.

[9] G. Paul. Approaches to abductive reasoning: an overview. *Artificial Intelligence Review*, 7:109-152, 1993.

[10] J. R. Quinlan. Unknown Attribute Values in Induction. *Proc. 6th Intern. Workshop on Machine Learning*, 164-168, Morgan Kaufmann, 1989.

[11] P. O'Rorke. Abduction and Explanation-Based Learning: Case Studies in Diverse Domains. *Computational Intelligence*, 10:295-330, 1994.

[12] C. Thompson and R. Mooney. Inductive Learning for Abductive Diagnosis. *Proc. of AAAI-94*.

[13] J. Wogulis. Revising Relational Domain Theories. *Proc. of the 8th Inter. Workshop on Machine Learning*, Evaston, IL, pp. 462-466, Morgan Kaufmann, 1991.