

# IDDD: An Inductive, Domain Dependent Decision Algorithm

L. Gaga<sup>1,2</sup>, V. Moustakis<sup>1,3</sup>, G. Charissis<sup>4</sup> and S. Orphanoudakis<sup>1,2</sup>

<sup>1</sup>Institute of Computer Science, Foundation of Research and Technology - Hellas (FORTH), Heraklion, Crete, Greece

<sup>2</sup>Department of Computer Science, University of Crete, Heraklion, Crete, Greece

<sup>3</sup>Department of Production and Management Engineering, Technical University of Crete, Chania, Crete, Greece

<sup>4</sup>Director, Pediatric Surgery Clinic, University Hospital, Heraklion, Crete, Greece

**Abstract.** Decision tree induction, as supported by ID3, is a well known approach of heuristic classification. In this paper we introduce *mother-child* relationships to model dependencies between attributes which are used to represent training examples. Such relationships are implemented via IDDD which extends the original ID3 algorithm. The application of IDDD is demonstrated via a series of concept acquisition experiments using a ‘real-world’ medical domain. Results demonstrate that the application of IDDD contributes to the acquisition of more domain relevant knowledge as compared to knowledge induced by ID3 itself.

## 1 Introduction

The ability to classify objects or events as members of known classes is a very common task for learning systems. A well known approach to heuristic classification is decision tree induction. An alternative to decision tree induction is given by ID3, an algorithm described in [1]. ID3 uses a heuristic search process to find a set of discriminant descriptions between classes, given: (1) A set of observational statements each of which is assigned to a certain class and (2) a universe of classes.

Working in a recursive manner, the algorithm selects the most discriminant attribute maximizing an information gain function at each step. The result is a tree in which nodes represent tests on attributes, while branches are possible values of the corresponding attributes.

In this paper we present IDDD, which extends ID3 using dependency relationships, between attributes, as domain knowledge agents. IDDD is based on *NewId*, an enhanced implementation of ID3, developed by the Turing Institute, [2]. In section 2 we present IDDD and the definition of the dependency relationships. In section 3, we present experimental results from a ‘real-world’ medical application. These results show the value of domain knowledge in decision tree induction, which is further discussed in section 4.

## 2 IDDD

The basic premise of IDDD lies in the deployment of domain knowledge in the induction process. We introduce dependency relationships between attributes that provide some structure over the rather ‘flat’ data representation used by *NewId*. A *simple* dependency, *B depends\_on A*, states that information represented by *B* is useful only when it is combined with information represented by *A*. *A* is identified as a *mother* and *B* as a *child* attribute.

Another form of the relation defined above is the exclusion of an attribute when a specific value has already been assigned to another attribute. We call this dependency *exclusive* and its definition is the following: *B depends\_on A : a<sub>i</sub>, . . . , a<sub>j</sub>*, where *a<sub>i</sub>, . . . , a<sub>j</sub>* represent distinctive values if *mother* is nominal and the interval [*a<sub>i</sub>, a<sub>j</sub>*] when *mother* is a linear attribute. We should note here that any *child* attribute may have only one *mother*, while this restriction does not apply to a *mother* attribute.

*NewId* is able to ‘handle’ relationships such as the ones defined above, yet this is not based on explicit modeling of domain knowledge. However, when the value of a *mother* is unknown, *NewId* may drift into irrelevant attribute selection. Furthermore, when both *A* and *B* score equally in information gain, *NewId* may select *B* instead of *A*. The outcome is a set of rules which may be accurate, from the point of view of classification accuracy but meaningless from the expert’s point of view.

One approach to addressing this problem is suggested by van Someren [3]. A preprocessing system for reconstructing the attribute set is elaborated. For instance, attribute *A* valued in  $V_A$  and *B* valued in  $V_B$  are replaced by an attribute *F* valued in  $V_A \times V_B$ . It can be readily seen that such an approach can lead to an enormous increase in the number of attribute values, which may ‘bias’ tree induction in favor of *F* [1]. Also, this approach does not address linear attributes. In IDDD, instead of reformulating the attribute set, we modify the node selection process. This work is also related to that of Manago [4], who proposed a frame-based language to represent training examples.

### 2.1 Simple Dependency

This dependency can also be defined in *NewId* using its *ordering* mechanism. To express the above dependency with *NewId* we must define the ordering relationship: *A BEFORE B*. The imposition of order among attributes affects the procedure via which the most discriminant attribute is selected. If *A* has not been selected splitting on *B* is prevented. The approach taken in IDDD is different. We distinguish between two cases:

*Case I:* *A* emerges as the most discriminant attribute. The dependency mechanism is not triggered but the information that *A* has been used is recorded to allow further selection of attribute *B* along the current branch.

*Case II:* *B* emerges as the most discriminant attribute. Selection of *B* is prevented but the algorithm is forced to establish a node for *A*.

## 2.2 Exclusive Dependency

This dependency is not handled by *NewId*. Yet, exclusive dependencies are able to represent complex attribute relationships. By stating that, *B DEPENDS\_ON A: a<sub>i</sub>, ... a<sub>k</sub>* we can prevent selection of *B* when *A* is valued in the *a<sub>i</sub> ... a<sub>k</sub>* subset.

When *A* is valued in a nominal set, handling of the exclusive dependency is similar with the case of the simple dependency.

When *A* is linear, the selection of a split point is determined by the knowledge embodied in the dependency relationship. In dealing with linear attributes, *NewId* uses a statistical procedure to determine a point to split. In IDDD, instead of relying on 'domain independent' statistical metrics, we assume that the presence of a specific value in a dependency relationship means that this value is important to the domain and use this value to split. Thus, when *A* is linear, we distinguish among three types of the exclusive dependency, namely:

*Case III: B DEPENDS\_ON A: ≤ a<sub>i</sub>.*

*Case IV: B DEPENDS\_ON A: > a<sub>i</sub>*

*Case V: B DEPENDS\_ON A: a<sub>i</sub> ... a<sub>j</sub>.*

Each of the expressions defined above implies a different approach both to the determination of a split value and to decision tree construction. In cases III and IV splitting is done at the *a<sub>i</sub>* point and selection of attribute *B* is restricted to branches where the value of *A* is either 'less than or equal to', or 'greater than', *a<sub>i</sub>* respectively. In case V, we first split on *a<sub>i</sub>* and then again on *a<sub>j</sub>*. The path defined over the  $a > a_i$  &  $a \leq a_j$  values restricts *B* selection.

## 3 Experimental Results

In the framework of the MLT (ESPRIT P2154) project, we developed a real-world medical application, involving the therapeutic decision procedure of Maldescensus Testis. Maldescensus Testis refers to a pathological state where one, or both testicles are not properly descending from their inner-body positions after birth. If untreated, this leads to non-functional (atrophied) testicles that may even become malignant (cancerous) in adulthood [5]. The classification task is to assign a case to one of the following therapeutic *classes*: Follow up, Hormonal Treatment, Surgical.

Attributes used to represent case data are (attribute names are enclosed in parentheses): visit (visit), age (age), etiology (etiology), status of one testicle (one-t), status of the other testicle (two-t), size of one testicle (size-of-one), size of the other testicle (size-of-two). To capture the fact that treatment occurs over several visits of the patient, the following attributes were added to link the previous visit and the current one: previous status of one testicle (pre-one-t), previous status of the other testicle (pre-two-t), previous treatment (treatment).

*NewId* provides a 'don't care' value for attributes that are not applicable under certain circumstances. For example, when visit = 'first', attributes pre-one-t, pre-two-t and treatment are not applicable. However, *NewId* induces rules like:

IF (pre-two-t=normal) & (two-t=normal) & (visit=first) THEN follow up.

In another instance, *NewId* misses the fact that a ‘pre’ status is not self contained. It has meaning only when combined with a ‘current’ status.

IF (one-t=inguinal) & (visit=third) & (pre-two-t=normal) THEN surgical

Finally, when a testicle’s pathology is ‘not palpable’, size cannot be measured. Although in such cases size was valued as ‘don’t care’, rules like the following one were induced by *NewId*:

IF (size-of-two>0.60) & (two-t=not palpable) THEN hormonal

Correction and/or elimination of such rules should lead to a more domain meaningful set.

To evaluate possible improvements in the rules induced by IDDD, three rule acquisition experiments were carried out. In the first experiment, *NewId* is used to induce rules without any domain knowledge. We name the set of induced rules as ‘original’. The three rules cited above belong to this set.

A different order in attribute selection might have prevented *NewId* from inducing rules like those included in the ‘original’ set. Thus, we placed some structure on domain attributes by using the *NewId*’s “ordering” definition facility [2]. Accordingly, the following attribute ordering scheme was defined:

- (1) visit BEFORE pre-one-t, visit BEFORE pre-two-t
- (2) one-t BEFORE pre-one-t, two-t BEFORE pre-two-t
- (3) one-t BEFORE size-of-one, two-t BEFORE size-of-two

This constitutes the second experiment; we shall refer to it as ‘ordering’. Although, ‘ordering’ improved the ‘original’ set of rules, induction of meaningless rules continued. One such rule is listed below:

IF (one-t=not palpable) & (size-of-one  $\leq$  0.60) THEN surgical

Finally, a third experiment was also conducted in which full *mother/child* dependencies were implemented; a ‘pre’ status attribute (pre-one-t, pre-two-t) depends on both ‘current’ status (one-t,two-t) and visit attributes. Since in IDDD a *child* may only possess one *mother*, the relationship “visit BEFORE ‘pre’ status” was maintained. Thus, the third experiment (‘dependency’) was conducted using the following attribute dependency and order relationships:

- (1) size-of-one DEPENDS\_ON one-t:not palpable
- (2) size-of-two DEPENDS\_ON two-t:not palpable
- (3) pre-one-t DEPENDS\_ON one-t, pre-two-t DEPENDS\_ON two-t
- (4) visit BEFORE pre-one-t, visit BEFORE pre-two-t

### 3.1 Results

Each experiment constitutes a specific domain description. For each domain description, the average number of rules induced and average classification accuracy was estimated by applying case data to a ( $V = 8$ ) cross validation process [6]. The results listed in Table 1 clearly indicate that use of domain knowledge decreases classification accuracy but contributes to the induction of considerably reduced rule sets. *NewId* tries to cover as many examples as possible. However, in the specific application domain, the class of an example may depend on the

occurrence of extraneous factors which are not recorded. In these cases, the assignment of examples to classes is not compatible to the pure medical knowledge. Thus, *NewId* induces rules which are meaningless from the expert's point of view. Since IDDD tries to produce a more domain meaningful set of rules, it is expected that it will not cover cases like the ones mentioned above, resulting in a decreased classification accuracy. Furthermore, 'ordering' is also responsible for the decrease in accuracy. Use of 'ordering' affects classification accuracy because it forces *NewId* to discard attributes even when these emerge as the most discriminant ones. This fact is inherited by the 'dependency' domain description.

Average	original	ordering	dependency
Number of Rules	146	84	84.5
Classification Accuracy	76.9	67.4	68.4

Table 1: ( $V = 8$ ) fold evaluation results

Our goal was to produce a more domain meaningful set of rules. We attempted the validation of results with the participation of four medical experts. Each one assigned one of the following labels to each rule:

*Valuable.* A rule that exactly matches with clinical decision procedures.

*Over-General.* A rule that fails to consider specific clinical evidence.

*Over-Specific.* A rule which represents more knowledge than necessary.

*Useless.* A rule perceived as meaningless by the expert.

'Ordering' alone does not improve rule quality. Table 2 indicates that the percentage of valuable rules in both domain descriptions is the same. Yet, the percentage of useless rules was improved and the number of rules covering negative examples was decreased. A rule covers a negative example when it is capable of classifying a case which belongs to a class different from the one implied by the rule. Rule sets are identified as distinctive. This means that we focused on rules belonging to one domain and not to the other. In Table 2 we also present the results of a cross 'dependency', 'original' assessment. Notice the improvement in all rule quality aspects; the percentage of useless rules decreases significantly while more than 50% of the rules were assessed as valuable.

	orig.	order.	orig.	depend.
Useless	117 (66%)	31 (49%)	110 (59%)	6 (8%)
Over-General	34 (19%)	15 (24%)	42 (22%)	23 (31%)
Over-Specific	16 (9%)	14 (22%)	18 (10%)	6 (8%)
Valuable	9 (5%)	3 (5%)	9 (5%)	38 (52%)
Distinctive Rules	176	63	188	73
Rules Covering Negative Ex.	28 (16%)	8 (13%)	28 (15%)	7 (10%)

Table 2: 'original' vs 'ordering' and 'original' vs 'dependency'

## 4 Conclusions

Attributes used to describe a real world application are often interrelated. These relationships are characteristics of the application domain. Their formal definition can facilitate the rule learning process leading to the acquisition of more qualitatively and domain relevant rules. IDDD proved effective in introducing to the tree induction process domain dependent knowledge via explicit definition of attribute dependency relationships. However, a shortcoming of the IDDD dependencies is the fact that definition is restricted to 'single mother multiple children' cases. That is, a child is not allowed to possess multiple mothers. Thus, multiple dependencies of the same attribute cannot be expressed.

This paper constitutes a contribution towards the formal use of domain dependent knowledge in induction. Use of *NewId* was only indicative and/or demonstrative. Dependency relationships can be readily represented and used in conjunction with other inductive learning algorithms, for instance with CN2 [7]. The effectiveness of dependency relationships defined in IDDD indicates that further work is necessary towards the investigation, formal definition, and handling of domain knowledge in the inductive process.

## Acknowledgements

Work reported in this paper was partially supported via CEC contracts MLT (ES-PRIT, P2154) and GAMES-II (AIM, A2034). The authors would like to thank Drs. Yannopoulos and Vlachakis of the Pediatric Surgery Clinic, University of Crete, and Ms. Natassa Papageorgiou for evaluating rule sets. The comments of the three anonymous reviewers helped considerably in improving an earlier draft; we gratefully acknowledge their work.

## References

- [1] J.R. Quinlan. Induction of Decision Trees. *Machine Learning*, 1.1:81-106, 1986.
- [2] R.A. Boswell. Manual for Newid version 2.0. *Technical Report*, January 1990.
- [3] M.W van Someren. Using Attribute Dependencies for Rule Learning. In K. Morik, editor, *Knowledge Representation and Organization in Machine Learning*, pages 192-210. Springer-Verlag, Berlin, 1989.
- [4] Michel Manago. Knowledge Intensive Induction. In *Proceedings of the Sixth International Workshop on Machine Learning*, pages 151-155, Cornell University, 1989. Morgan Kaufmann.
- [5] W.E Grupe. Abnormalities of the Genital Tract. In M.E Avery and H.W Taesch Jr., editors, *Schaffer's Diseases of the Newborn*, pages 401-411. W.B Saunders Company, Philadelphia, 1984.
- [6] L. Breiman, J.H Friedman, R.A Olshen, and C.J Stone. *Classification and Regression Trees*. Wadsworth International, 1984.
- [7] Peter Clark and Tim Niblett. The CN2 Induction Algorithm. *Machine Learning*, 3:261-283, 1989.