A Human Like Incremental Decision Tree Algorithm

Combining Rule Learning, Pattern Induction, and Storing Examples

Christina Zeller and Ute Schmid

Cognitive Systems Group, University of Bamberg
An der Weberei 5, 96045 Bamberg, Germany
{Christina.Zeller,Ute.Schmid}@uni-bamberg.de

Abstract. Early machine learning research was strongly interrelated with research on human category learning while later on the focus shifted to the development of algorithms with high performance. Only recently, there is a renewed interest in cognitive aspects of learning. Machine learning approaches might be able to model and explain human category learning while cognitive models might inspire new, more human like, approaches to machine learning. In cognitive science research there exist different theories of category learning, especially, rule-based approaches, prototypes, and exemplar-based theories. To take account of the flexibility of human learning and categorization we propose a human like learning algorithm. In the algorithm we combine incremental decision tree learning, least general generalization, and storing examples for similarity-based categorization. In this paper we present first ideas of this algorithm.

Keywords: (human) supervised category learning, cognitive modeling, incremental decision trees, least general generalizations

1 Introduction

Early machine learning research strongly related to human category learning (cf. Michalski, Carbonell, & Mitchell, 1983). For example, decision tree algorithms were inspired by research on human category learning by Bruner, Goodnow, and Austin (1956). They investigated how humans learned conjunctive or disjunctive categorization rules for a fixed set of artificial stimuli. It could be shown that most humans learned such rules in an incremental way using for example the wholist strategy, that is, they generate an initial rule which is sequentially modified for new examples which do not confirm to the current rule. Based on these findings Hunt, Marin, and Stone (1966) developed the first decision tree algorithms. Subsequently, Unger and Wysotzki (1981) introduced the incremental decision tree algorithm Cal 2 as an ideal model for human category learning and Quinlan (1986) developed the well-known ID3.

In: M. Leyer (Ed.): Proceedings of the LWDA 2017 Workshops: KDML, FGWM, IR, and FGDB.
Rostock, Germany, 11.-13. September 2017, published at http://ceur-ws.org

Starting in the 1990s machine learning research was focused on the development of new algorithms with high performance and not on cognitive plausibility. Therefore, machine learning and cognitive modeling research separated (Langley, 2016). For example, a main characteristic of human learning is life-long learning, that is, incremental and cumulative learning, while in the field of machine learning many batch learning algorithms were developed (e.g., ID3, neuronal network approaches, Bayes classifier; Mitchell, 1997). The lack of research in the field of incremental and cumulative and therefore life-long machine learning was addressed and motivated by human learning recently but it did not focus on cognitive plausibility of the algorithms (Thrun, 1998).

Another main characteristic of human learning is the flexible use of different strategies while learning and while using the learned knowledge in categorization tasks. That is, in human category learning there is evidence for rule-based learning, prototype learning, and exemplar-based learning which are combined in hybrid theories of human category learning (Kruschke, 2008). In machine learning multistrategy learning has been focused in the 1980s and 1990s (Michalski, 1993; Langley, 2016) and Langley (2016) has suggested that the research should continue in this line. Multistrategy learning shares characteristics with ensemble learning. However while in ensemble learning focus is on combining different models to improve classification performance, in multistrategy learning, focus is on achieving human-level flexibility in application of different knowledge structures in different contexts (Dietterich, 2002).

Besides, Langley (2016) has recommended that researcher should work interdisciplinary because cognitive science and machine learning can mutually profit from each other. Machine learning approaches might be able to model and explain human category learning while cognitive models might inspire new, more human like, approaches to machine learning. Following these thoughts, we currently are developing a human-inspired learning approach based on Cal 2. Consequently, we focus on learning in supervised classification settings, or in other words in categorization learning from labeled examples.

Therefore, in the following section we describe Cal 2 as the incremental rule-based basis of our algorithm and ID3 which offers an idea for exemplar-based human category learning. Then we introduce least general generalizations (Mitchell, 1977) which inspire a prototypical view of learning. The algorithm combining these ideas is presented and applied to a small example in Section 3. In the last section we discuss further steps like possible extensions of our learning algorithm and the evaluation of the proposed approach.

2 Basic Symbolic Approaches to Category Learning

In this section we first describe relevant decision tree learning algorithms and how they connect to human category learning theories. However, not all aspects of human category learning can be explained with rules as generated with decision trees. Therefore, in the second part of this section we take a closer look on least

general generalizations which could relate to the prototype theories of human category learning.

2.1 Decision Tree Learning, Rules and Exemplars

Algorithm 1 shows the decision tree learner Cal 2 (Unger & Wysotzki, 1981). The algorithm handles examples in an incremental way where the information of previous examples is stored implicitly in the tree structure, but there is no knowledge of previous or later examples given explicitly in each step. The algorithm terminates successfully if all examples are classified correctly. Successful termination is guaranteed for linear separable examples, that is, disjunctive categories. A typical reason for non-disjunctivness is that the given set of features is not sufficient to discriminate the training examples. If examples are not linearly separable, the algorithm fails in Line 9.

The algorithm does not provide a strategy for feature selection. However, the *next feature* (cf. Line 9) can make use of a predefined selection strategy. In the simplest case the algorithm could choose a feature randomly or use a predefined feature order. The well-known information gain as proposed in ID3 (Quinlan, 1986) is no feasible feature selection criterion for Cal 2 because to calculate the information gain of a specific feature all examples have to be known in advance and therefore the learning is not incremental anymore. However, we currently investigate incremental variants of information gain, for example, the *igain* measure (Zeller & Schmid, 2016). Calculating an incremental variant of the information gain implies that the already used examples are stored explicitly and not only implicitly in the tree structure. Storing examples explicitly refers to the exemplar-based theories in cognitive psychology (Nosofksy & Palmeri, 1997; Jäkel, Schölkopf, & Wichmann, 2008).

Algorithm 1 Incremental Decision Tree Algorithm Cal 2 (Unger & Wysotzki, 1981)

```
 1: procedure CAL 2(examples)
 2:     Start with a tree containing only * as node
 3:     while at least one of the examples changes the tree structure do
 4:         if current example is classified correct then
 5:             do nothing
 6:         if current example is classified as * then
 7:             replace * with the class of current example
 8:         if current example is classified wrong then
 9:             add the next feature in the tree
10:             set for the branch with the feature value of current example
                the class of current example and set * for all other branches
```

The combination of Cal 2 and *igain* was used to model human categorization behavior obtained in an experiment with stimuli in form of lamps (Lafond, Lacouture, & Cohen, 2009). The lamps were described by four features with two

discrete feature values each. Features are defined for the base, upright, shade, and top of each lamp and are given as $F1, \ldots, F4$. The categorization task in the experiment is to decide whether a specific feature combination belongs to Category A or B. The knowledge structure for categorization has to be induced by the humans from nine training examples. Category distribution follows the in psychology often used 5-4 structure (see Table 1; Medin & Schaffer, 1978). The *igain* of the features helped to explain differences in decision trees when the presentation order of input examples where different for the learner (Zeller & Schmid, 2016). While ordering effects in incremental learning were addressed in early machine learning (cf. Fisher, 1993; Langley, 1995) it only recently comes into focus in the field of cognitive modeling of human category learning (cf. Carvalho & Goldstone, 2015; Mathy & Feldman, 2016).

Table 1. The 5-4 category structure (cf. Medin & Schaffer, 1978) and the input example order for the example in Figure 1.

Example	$F1$	$F2$	$F3$	$F4$	class
$e1$	1	1	1	0	A
$e2$	1	0	1	0	A
$e3$	1	0	1	1	A
$e4$	1	1	0	1	A
$e5$	0	1	1	1	A
$e6$	1	1	0	0	B
$e7$	0	1	1	0	B
$e8$	0	0	0	1	B
$e9$	0	0	0	0	B
Order:	$e8$, $e2$, $e5$, $e6$, $e7$, $e1$, $e4$, $e9$, $e3$				

2.2 Least General Generalizations and Prototypes

In human category learning there is strong evidence that, especially for natural categories, humans do not learn feature combinations (rules) but prototypes (Rosch & Mervis, 1975). Categories over entities with discrete features are formed by family resemblance, that is, a prototype is given by the most frequent feature value for each feature. It could be shown that humans often search for similarity by (implicit) counting of the occurrence of feature values within a category and compare it with the occurrence of feature values in the other categories.

A similar idea is described as least general generalizations in machine learning where a pattern for a set of examples of a class is generated (cf. Mitchell, 1977). This pattern includes for each feature either a concrete feature value that matches with all examples or a wild card if there exists more than one feature value for this feature in the set of examples. Least general generalizations can be formed for symbolic features, but also for structural features, terms and graphs

(cf. Schmid, Hofmann, Bader, Häberle, & Schneider, 2010; Siebers, Schmid, Seuß, Kunz, & Lautenbacher, 2016).

3 An Incremental Decision Tree Algorithm Including Most Specific Patterns and Examples

Algorithm 2 shows our human-inspired decision tree learning approach which realizes incremental learning of rules. Furthermore, to take account of the human flexibility of the categorization process, it integrates storage of generalized patterns and of examples. That is, dependent on the context of a categorization task the learned knowledge structure allows to categorize a new object by applying a rule, by pattern matching, or by similarity to known examples.

Core of our algorithm is Cal 2 which can be seen in the procedure INCLUDE (cf. Line 5) where a new example is included to a given tree. To take account of the flexibility of human categorization, the decision tree is enriched by least general generalizations at each decision node and the leafs (cf. most specific pattern, for example, in Line 23). Besides, the examples are stored at the leafs (cf., for example, Line 24). In difference to Cal 2 our algorithm considers every input example only once and stores it in the matching current leaf. For this first version of the algorithm which is restricted to disjunctive categories, termination conditions are inherited from Cal 2. We have realized this algorithm in Prolog and are currently investigating its ability to model reported findings of human category learning.

The result of the algorithm depends on the category structure, the order of the input examples, and the feature selection criterion. Figure 1 shows the generated tree, with the 5-4 category structure mentioned above and the order of the input examples given in Table 1 when using the *igain* for feature selection. Left branches show the branches with feature value 1, right branches show the branches with feature value 0. The least general patterns contain the feature values with the following structure: $< F1, F2, F3, F4 >$. The ? stands for the wildcard.

4 Future Work

The proposed algorithm is work in early progress and further aspects need to be considered: extension of the learning algorithm and evaluation of the proposed approach.

The current algorithm fails if a new example is categorized erroneously and there is no feature available to extend the tree such that correct categorization becomes possible. This category structure can, for example, be generated by non-disjunctive categories. Learning scenarios involving overlapping categories are often used in psychological studies to demonstrate that humans do not learn rules (Kruschke, 2008). We propose to keep the rule-based structure but to sample examples at the leaf nodes and postpone branching decisions until a certain

Algorithm 2 Incremental Decision Tree Algorithm Including Most Specific Patterns and Examples

```
 1: procedure incrDTmsp(examples)
 2:     tree ← empty tree
 3:     for each example e ∈ examples do
 4:         tree ← include(e, tree)
        return: tree

 5: procedure include(e, tree)
 6:     new tree ← empty tree
 7:     if tree is empty then
 8:         new tree ← (most specific pattern of e, label of e, e)
 9:     else if tree is leaf with the same class as e then
10:         all examples ← all examples of the leaf and e
11:         new tree ← same(all examples)
12:     else if tree is leaf with a different class as e then
13:         all examples ← all examples of the leaf and e
14:         new tree ← different(all examples)
15:     else
16:         branch ← the branch of tree matching with e
17:         node msp ← update msp of root node of branch with e
18:         new branch ← (node msp, include(e, branch))
19:         new tree ← substitute branch with new branch in tree
20:     return: new tree

21: procedure same(all examples)
22:     branch ← empty tree
23:     msp ← most specific pattern of all examples
24:     branch ← (msp, label of all examples, all examples)
25:     return: branch

26: procedure different(all examples)
27:     branch ← branch all examples by a feature not yet used
28:     for each attribute in branch do
29:         if attribute has no examples then
30:             node ← empty leaf
31:         else if all examples in attribute have the same class then
32:             branched examples ← examples of attribute
33:             node ← same(branched examples)
34:         else
35:             branched examples ← examples of attribute
36:             node msp ← most specific pattern of branched examples
37:             node ← (node msp, different(branched examples))
38:         branch ← add node for the attribute of the branch
39:     return: branch
```

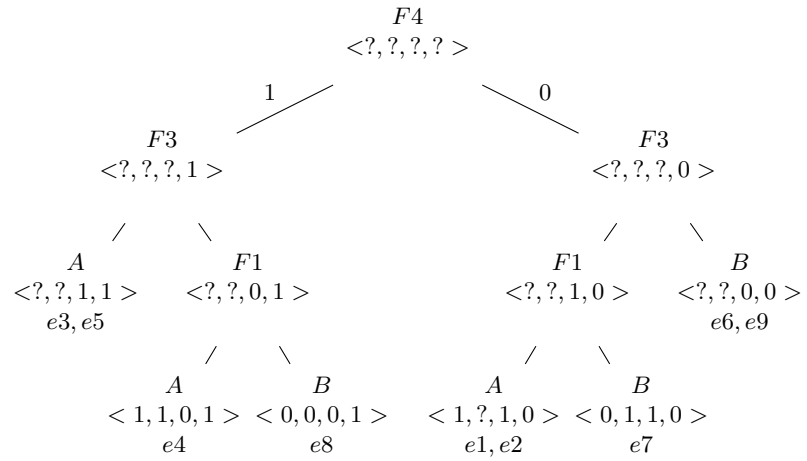

Fig. 1. Generated tree for the 5-4 category structure and the the input example order stated in Table 1. Left branches show the branches with feature value 1, right branches show the branches with feature value 0. The least general generalizations contain the feature values with the following structure: $< F1, F2, F3, F4 >$. The ? stands for the wildcard.

amount of evidence is reached. That is, we introduce "delayed branching" that can reduce but not solve the problem of failing.

Delayed branching typically leads to decisions based on stronger evidence, because a larger sample of examples offers a better basis for pattern generalization as well as for similarity-based categorization. Besides, delayed branching might help to reduce overfitting. While in the tree given in Figure 1 most leafs are associated with patterns representing feature vectors and single examples, trees constructed with delayed branching will less often specialize in such a way.

A simple approach to sampling is just to define a threshold for the number of examples needed before a new branch is introduced. This idea is realized in Cal 3 (Unger & Wysotzki, 1981), using a parameter for sample size. We plan to investigate more sophisticated criteria such as prior probabilities (Frermann & Lapata, 2016), or similarity-based coherence of examples (Michalski & Stepp, 1983).

Besides, the least general generalizations does not reflect the prototype approach in detail. That is, prototypes are build by the most frequent feature values, while the least general generalizations represents the feature value that are common among all examples. We plan a variant of the algorithm where at each node the feature values with the highest frequency instead of the least general generalizations is annotated.

Additionally, it might be interesting to make restructuring of the tree and forgetting of examples possible. In human category learning there is evidence that humans completely reject partially learned rules and start with a new rule (Unger

& Wysotzki, 1981). If we assume that wrong information of categories, that is noise, is seldom we could handle noise by forgetting seldom seen examples and restructure the tree. This procedure reflects representational shifts (Johansen & Palmeri, 2002).

Currently, categorization of a new example—while learning and while using the tree—is strictly guided by the branching in the decision tree. That is, patterns and exemplars have no influence on the learning and categorization. Hybrid theories in human category learning take into account that humans are flexible in using different strategies in different situations (Kruschke, 2008; Nosofsky, Palmeri, & McKinley, 1994; Rosch, 1983). The factors for using one or another information in the tree need to be investigated, and incorporated in the current algorithm.

Since our goal is to develop a human like machine learning algorithm as well as a machine learning inspired cognitive model of categorization learning, evaluation of the algorithm has to address performance characteristics as well as validity as a cognitive model. We plan to take a closer look on efficiency (time, number of training examples) and accuracy in comparison with other machine learning approaches, such as (other) decision tree learners, random forests, and inductive logic programming (Schmid, Zeller, Besold, Tamaddoni-Nezhad, & Muggleton, 2017).

Furthermore, we want to compare the learning steps as well as the resulting knowledge structure of our algorithm with human behavior and we plan to predict human behavior with our algorithm. For this aim we want to select several learning domains which have been introduced in cognitive science literature. We are especially interested in domains where it was shown that humans make use of different strategies for learning and categorization. Among these domains is the lamp domain introduced above but also artificial domains of letter strings, or natural categories such as fruit (Rosch & Mervis, 1975).

References

Bruner, J. S., Goodnow, J. J., & Austin, G. A. (1956). *A study of thinking: With an appendix on language by Roger W. Brown.* New York, NY: Wiley.

Carvalho, P. F., & Goldstone, R. L. (2015). What you learn is more than what you see: What can sequencing effects tell us about inductive category learning? *Frontiers in Psychology*, *6*.

Dietterich, T. G. (2002). Ensemble learning. *The handbook of brain theory and neural networks*, *2*, 110–125.

Fisher, D. (1993). Ordering effects in incremental learning. In *Training Issues in Incremental Learning: AAAI Spring Symposium at Stanford University* (pp. 35–42). Menlo Park, CA: AAAI Press.

Frermann, L., & Lapata, M. (2016). Incremental Bayesian category learning from natural language. *Cognitive Science*, *40*, 1333–1381.

Hunt, E. B., Marin, J., & Stone, P. J. (1966). *Experiments in induction.* New York, NY: Academic Press.

Jäkel, F., Schölkopf, B., & Wichmann, F. A. (2008). Generalization and similarity in exemplar models of categorization: Insights from machine learning. *Psychonomic Bulletin & Review*, *15*(2), 256–271.

Johansen, M. K., & Palmeri, T. J. (2002). Are there representational shifts during category learning? *Cognitive Psychology*, *45*, 482–553.

Kruschke, J. K. (2008). Models of categorization. In R. Sun (Ed.), *Computational psychology* (pp. 267–301). Cambridge University Press.

Lafond, D., Lacouture, Y., & Cohen, A. L. (2009). Decision-tree models of categorization response times, choice proportions, and typicality judgments. *Psychological Review*, *116*, 833–855.

Langley, P. W. (1995). Order effects in incremental learning. In P. Reimann & H. Spada (Eds.), *Learning in humans and machines: Towards and interdisciplinary learning science* (pp. 154–167). Oxford: Elsevier.

Langley, P. W. (2016). The central role of cognition in learning. *Advances in Cognitive Systems*, *4*, 3–12.

Mathy, F., & Feldman, J. (2016). The influence of presentation order on category transfer. *Experimental Psychology*, *63*(1), 59–69.

Medin, D. L., & Schaffer, M. M. (1978). Context theory of classification learning. *Psychological Review*, *85*, 207–238.

Michalski, R. S. (1993). *Multistrategy learning.* Boston: Kluwer Academic Publishers.

Michalski, R. S., Carbonell, J. G., & Mitchell, T. M. (1983). *Machine learning: An artificial intelligence approach.* Springer.

Michalski, R. S., & Stepp, R. (1983). Automated construction of classifications: Conceptual clustering versus numerical taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *5*, 219–243.

Mitchell, T. M. (1977). Version spaces: A candidate elimination approach to rule learning. In *Fifth International Joint Conference on AI* (pp. 305–310). Cambridge, MA: MIT Press.

Mitchell, T. M. (1997). *Machine learning.* McGraw-Hill.

Nosofksy, R. M., & Palmeri, T. J. (1997). An exemplar-based random walk model of speeded classification. *Psychological Review*, *104*(2), 266–300.

Nosofsky, R. M., Palmeri, T. J., & McKinley, S. C. (1994). Rule-plus-exception model of classification learning. *Psychological Review*, *101*(1), 53–79.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, *1*(1), 81–106.

Rosch, E. (1983). Prototype classification and logical classification: The two systems. In E. K. Scholnick (Ed.), *New trends in conceptual representation: Challenges to Piaget's theory?* (pp. 73–85). Hillsdale, NJ: Erlbaum.

Rosch, E., & Mervis, C. B. (1975). Family resemblances: Studies in the internal structure of categories. *Cognitive Psychology*, *7*, 573–605.

Schmid, U., Hofmann, M., Bader, F., Häberle, T., & Schneider, T. (2010). Incident mining using structural prototypes. In N. García-Pedrajas, F. Herrera, C. Fyfe, J. M. Benítez, & M. Ali (Eds.), *Trends in Applied Intelligent Systems: The Twenty Third International Conference on Industrial,*

Engineering & Other Applications of Applied Intelligent Systems, IEA-AIE 2010, Cordoba, Spain, 01.–04. June 2010 (Vol. 6097, pp. 327–336). Springer.

Schmid, U., Zeller, C., Besold, T., Tamaddoni-Nezhad, A., & Muggleton, S. (2017). How does predicate invention affect human comprehensibility? In J. Cussens & A. Russo (Eds.), *Inductive Logic Programming – 26th International Conference, ILP 2016, London, UK, 04.-06. September 2016, Revised Selected Papers* (Vol. 10326, pp. 52–67). Springer.

Siebers, M., Schmid, U., Seuß, D., Kunz, M., & Lautenbacher, S. (2016). Characterizing facial expression by grammars of action unit sequences: A first investigation using ABL. *Information Sciences*, *329*, 866–875.

Thrun, S. (1998). Lifelong learning algorithms. In S. Thrun & L. Pratt (Eds.), *Learning to learn* (pp. 181–209). Boston, MA: Springer US.

Unger, S., & Wysotzki, F. (1981). *Lernfähige Klassifizierungssyssteme (Classification Systems Being Able to Learn)*. Berlin, Germany: Akademie-Verlag.

Zeller, C., & Schmid, U. (2016). Rule learning from incremental presentation of training examples: Reanalysis of a categorization experiment. In *13th Biannual Conference of the German Cognitive Science Society; Bremen, Germany, 26.–30. September 2016* (pp. 39–42).