

Acquiring Knowledge by Efficient Query Learning

Achim G. Hoffmann

Technische Universität Berlin
Institut für Angewandte Informatik
Franklinstr.28/29
D-1000 Berlin 10, Germany

Sunil Thakar

Research Institute Berlin
Daimler-Benz AG
Alt-Moabit 91b
D-1000 Berlin 21, Germany

Abstract

Membership queries extended with the meta query concept is proposed as a method to acquire complex classification rules. Furthermore, relevant concept classes, where a small number of queries is sufficient, are characterized. In this paper we advocate and present the benefits of the use of queries in order to learn a target concept efficiently. Thus providing the foundations for automating the knowledge acquisition process. Based on these results, we developed a knowledge acquisition tool KAC-Z which uses queries about specific domain objects. The systems usefulness has been demonstrated by its application in the domain of manufacturing (cutting) industry.

1 Introduction

In building expert systems the knowledge acquisition is considered to be a bottleneck [3]. This is due to the fact that acquiring knowledge from an expert is generally time-consuming and error-prone. Thus, one is motivated to overcome this bottleneck by automating the knowledge acquisition, i.e. through building learning components for expert systems [4]. A successful learning component in an expert system will probably have to use queries directed towards the experts. For example, Sammut and Banerji's system [7] uses membership queries about specific examples as part of its strategy for efficiently learning a target concept. Let us consider the problem of using queries in order to learn an unknown concept. For this purpose Angluin [1] suggested different types of queries. Among these query types are membership queries which ask for the correct classification of a particular object. Furthermore, she proved that for many concept classes an exhaustive or nearly exhaustive search is necessary when using membership queries. However, in [6] concept classes have been introduced where efficient learning via various queries is possible. In the present paper these concept classes are significantly extended by introducing the notion of meta queries. Meta

queries ask for hints how to construct complex classification rules. By the use of meta queries combined with membership queries our system is in a position to learn concept classes for which otherwise an exhaustive search would be necessary. Based on this work we developed a knowledge acquisition system KAC-Z which exploits the theoretical results presented in this paper. To prove the applicability of our system in the real world we describe the application of KAC-Z in the domain of manufacturing industry. KAC-Z will be used to acquire the knowledge of an expert which selects an appropriate cutting metal for a specific cutting process. This paper is organized as follows. In section 2 our theoretical results concerning efficient query learning are presented. Section 3 describes the design of KAC-Z. The final section contains the conclusion.

2 Theoretical Foundations

We consider a set of objects X . The task of the learning system is to determine the class of each object in X , i.e. to determine for each object whether or not it belongs to the target concept. For this purpose, we assume a class of concepts $C \subseteq 2^X$ which underlies the learning system in the following sense. The learning system will classify the objects only according to one particular concept $c \in C$. The learning system L is allowed to address membership queries to an oracle. A **membership query** means that L provides an arbitrary object $O \in X$ to the oracle. The answer of the oracle will be 'yes' or 'no' depending on whether O belongs to the target concept or not. One simple but inefficient way of determining the correct class of all objects in X via membership queries is to present the description of each object in X to the oracle. Therefore, a trivial upper-bound for the required number of membership queries is $|X|$. Depending on the actual structure of the concept class C there may be clever query strategies which allow the learning system to determine the target concept with much less than $|X|$ membership queries. This may be possible because the classification of most objects will be logically implied by the classification of some *crucial* objects in X . In this paper we investigate the structure of concept classes which allow the learning system to use such

clever query strategies in order to learn the target concept efficiently, i.e. with a small number of queries. The following results are also applicable in the case of learning multiple classes instead of learning just one class as pointed out in [5]. In section 2.1 we define the property of a concept class being *independently monotonic*. This property allows to learn a concept from C with a small number of queries. As shown in Theorem 1 in section 2.2 by using the preceding Lemma which states an important consequence of a concept class being independently monotonic. In section 2.3 we extend the definition of being independently monotonic to the property of being *k-independently monotonic*. Concept classes of this kind allow to avoid an exhaustive search if initially k positive examples of the target concept are given to the learning system. A corresponding upper bound is given in Theorem 2 in section 2.3. In 2.4 the notion of meta queries is formalized.

2.1 Preliminaries

Let X be a finite set of objects, $C \subseteq 2^X$ be a set of concepts or a concept class. That means, C is a set of subsets of X with $|C| > 1$. Let $c \in C$ be an arbitrary concept of C . An object $x \in X$ is called a **positive example** of c iff $x \in c$ and a **negative example** of c iff $x \notin c$ respectively. A concept c' is **consistent with a positive (negative) example** x of a concept c iff $x \in c'$ ($x \notin c'$).

The **Vapnik-Chervonenkis dimension** [8] was introduced in the context of learning theory in [2]. A set $s \subseteq X$ is said to be **shattered by C** iff $\{s \cap c \mid c \in C\} = 2^s$. The **Vapnik-Chervonenkis dimension of C** , in short $VC\text{-Dim}(C)$, is the cardinality of the greatest set $s \subseteq X$ shattered by C . That means, the Vapnik-Chervonenkis dimension is given by

$$VC\text{-Dim}(C) = \max_{s \in \{s \mid s \subseteq X \wedge \{s \cap c \mid c \in C\} = 2^s\}} |s|$$

In the following, a class of concept classes is specified for which the *Vapnik-Chervonenkis dimension* can be used to give an upper-bound for the required number of membership queries.

Definition 1 Let $C \subseteq 2^X$ be a concept class and c an arbitrary concept in C . $s \in C$ is a **minimal superconcept** of c iff $c \subset s$ and there is no concept $c' \in C$ between c and s . I.e., there is no $c' \in C$ such that $c \subset c' \subset s$ holds.

Definition 2 A set of concepts $I = \{c_1, \dots, c_m\}$ is **independent in C** iff for all concepts $c_i \in I$ there exists a fixed object $x_i \in c_i$ as follows: For the union U of each subset $I_s \subseteq I$ of concepts in I there is a concept $c \in C$ such that $U \subseteq c$ and for all concepts $c_j \in (I \setminus I_s)$ c does not contain the corresponding object x_j . More formally: A set $I = \{c_1, \dots, c_m\}$ of concepts is **independent in C** iff $\forall (c_i \in I) \exists (x_i \in c_i) \forall (I_s \subseteq I) \exists (c_s \in C) ((\bigcup_{c_r \in I_s} c_r) \subseteq c_s \wedge \forall (c_j \in (I \setminus I_s)) (x_j \notin c_s))$

Let C be the set of linear decision functions. A particular linear decision function can be viewed as a hyperplane in the n -dimensional euclidean space E^n . In E^2 it is a line that partitions the plane into two parts. In the diagram, the positive examples lie in the lower part. The dotted lines indicate the only two minimal superconcepts of the concept represented by the heavy line. smallest superconcepts of c .

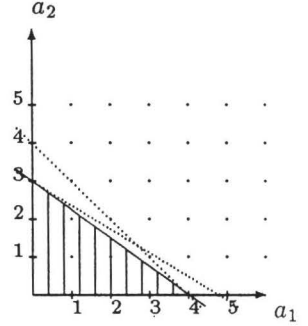


Figure 1: Linear threshold functions are independently monotonic

Note: This also means if I is independent in C then there exists a set $s \subseteq X$ of $|I|$ objects *shattered* by C . The set s contains one appropriate object from each concept in I respectively. In particular, $s = \{x_1, \dots, x_m\}$.

Example:

Let $C = \{\{\}, \{1\}, \{2\}, \{3\}, \{4\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 3\}, \{2, 3, 4\}, \{1, 2, 3, 4\}\}$. Let $I_0 = \{\{1\}, \{2\}, \{3\}\}$. Then I_0 is **independent in C** since for any union of sets in I_0 there exists a concept $c \in C$ that is disjoint with the remaining sets in I_0 . In contrast to that, the set $I_1 = \{\{2\}, \{3\}, \{4\}\}$ is **not independent in C** . That is due to the fact that any concept $c \in C$ covering both sets $\{4\}$ and $\{2\}$ also contains the element '3'. Thus, there is **no** concept $c \in C$ such that $(\{2\} \cup \{4\}) \subseteq c$ and the remaining set $\{3\} \in (I_1 \setminus (\{2\} \cup \{4\}))$ contains an element not contained in c .

Definition 3 A concept class C is **independently monotonic** iff for all concepts $c \in C$ the set $M(c)$ of minimal superconcepts of c in C is independent in C and the empty set is a concept in C as well.

Examples for relevant concept classes that are independently monotonic follow:

Let the set of objects X be the set of all attribute-value vectors $\langle v_1, v_2, \dots, v_{n-1}, v_n \rangle$ where $v_i \in \{1, \dots, j\} \forall i \in \{1, \dots, n\}$.

- **Onesided conjunctive threshold functions:** Let C be the set of all functions f of the following kind:

$$f(\langle v_1, \dots, v_n \rangle) = \begin{cases} 1 & \text{if } \forall i \in \{1, \dots, n\} t_i \geq v_i \\ & \text{where } t_i \in \{0, \dots, j\} \\ 0 & \text{otherwise} \end{cases}$$

For C representing the set of onesided-conjunctive threshold functions holds $VC\text{-Dim}(C)=n$.

Let C be the set of orthogonal rectangles. Then, by C being 1-independently monotonic is meant that for an arbitrary rectangle, given a point p within that rectangle we can expand or not expand it in all four directions *independently*.

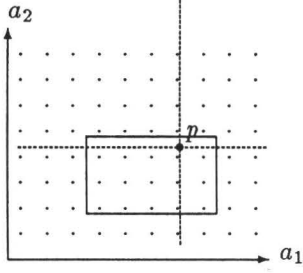


Figure 2: A geometrical example for a concept class being 1-independently monotonic

- Linear decision functions: Let C be the set of all functions f of the following kind:

$$f(\langle v_1, \dots, v_n \rangle) = \begin{cases} 1 & \text{if } \sum_{i=1}^n w_i v_i \leq t \text{ where} \\ & t, w_i \in \mathbb{N} \forall i \in \{1, \dots, n\} \\ 0 & \text{otherwise} \end{cases}$$

For C representing the set of linear decision functions holds $VC\text{-Dim}(C)=n$.

See Figure 1.

As the examples indicate, concept classes that do not require exhaustive searches have to be rather simple structured. That kind of being *rather simple structured* is reflected by the property of being independently monotonic.

For the proof of the following Lemma the next definition is still required.

Definition 4 An object $O \in X$ dominates an object $o \in X$ in C iff o is contained in all concepts of C in which O is contained. That is, iff the membership of O in the unknown target concept $c_t \in C$ implies the membership of o in c_t .

Example: Let $X = \{d, e, f\}$ and $C = \{c_1 = \{d, e, f\}, c_2 = \{d, e\}, c_3 = \{f\}, c_4 = \{e\}, c_5 = \{\}\}$. In this example the object 'd' dominates the object 'e' in C . The reason for this is that all concepts in C which contain 'd' contain the object 'e' as well. On the other hand, the object 'f' does not dominate the object 'e' because there is a concept $c_3 \in C$ which does not contain 'e' while it is containing 'f'.

2.2 The upper-bound

For the following Theorems the Lemma below is used, which exhibits an important property of *independently monotonic* concept classes.

Lemma Any independently monotonic concept class $C \subseteq 2^X$ has the following property: There are $VC\text{-Dim}(C) = d$ disjoint subsets $s_1, \dots, s_d$ such that each of

these subsets is linearly ordered by the dominance relation in the concept class C . Furthermore, determining the objects in $s_1, \dots, s_d$ being positive or negative examples of an unknown target concept $c_t \in C$ leaves only a single concept $c' \in C$ consistent with the classified objects in $s_1, \dots, s_d$. That means, $c' = c_t$.

Proof: We prove the Lemma by contradiction.

Assumption: There is a set $s = \{x_1, \dots, x_{d+1}\}$ of more than $VC\text{-Dim}(C)$ objects in X such that none of these objects is dominating another object of s in C . An object x not dominating another object y means that x and y belong to different concepts c_x and c_y . I.e. x and y belong to different superconcepts of the empty set such that neither c_x is a superconcept of c_y nor vice versa. Let be $c_1, \dots, c_{d+1}$ the corresponding concepts to which the objects in s belong respectively, such that none of these concepts is a superconcept of another one. For c_s being a *nonminimal* superconcept of a concept c means, there are concepts between c and c_s in the manner $c \subset c' \subset c'' \subset \dots \subset c^{**} \subset c_s$, i.e. there is a minimal superconcept $c^{s'}$ of c between c and c_s . Since the set $M(c)$ of minimal superconcepts of a concept $c \in C$ is independent by definition, the minimal superconcept $c^{s'}$ of c has in turn minimal superconcepts covering exactly one of the minimal superconcepts of $M(c) \setminus c^{s'}$. This property holds for all concepts between c and c_s , such that it also holds for the concept c^{**} of which c_s is a minimal superconcept. If the concepts $c_1, \dots, c_{d+1}$ do not dominate each other in C and there is a concept $c_m \in C$ such that $c_1, \dots, c_{d+1}$ are all minimal superconcepts of c_m . That is due to the fact that $c_1, \dots, c_{d+1}$ all are superconcepts of the empty set contained in C . However, c_m can be constructed by executing the following procedure using a variable 'current concept' c_c :

```

 $c_c \leftarrow \emptyset;$ 
for  $i=1$  to  $d+1$  do
  while  $c_i$  is not a minimal superconcept of  $c_c$  do
     $c_c \leftarrow c_i';$ 
  endwhile;
endfor;
```

After executing this procedure all superconcepts $c_1, \dots, c_{d+1}$ must be minimal superconcepts of c_c since $c_1, \dots, c_{d+1}$ are not dominating each other in C . But this is a contradiction to the definition of C being independently monotonic. $\square$

That situation can be illustrated in the geometrical example given in Figure 1 above. In Figure 1 for each dimension there exists a set O of objects along the axes a_1 and a_2 which are linearly ordered according to the dominance relation in C . That means, all points on the axes in the dashed area are positive examples of c while

	j=2	j=5	j=10	j=100	j=1000	j=65535
n=1	2	3.32	4.32	7.64	10.97	16.99
n=2	4	9.29	13.29	26.58	39.86	63.99
n=5	18.39	51.44	76.44	159.49	242.53	393.39
n=10	76.78	208.97	308.97	641.17	973.36	1576.78
n=30	782.8	1972.53	2872.53	5862.26	8851.99	14282.77
n=100	9435.6	22654.9	32654.9	65874.18	99093.46	159335.4

Table 1: Number of required queries

all remaining points in O are negative examples of c .
The upper-bound Theorem follows:

Theorem 1 *Let X be a set of objects, $C \subseteq 2^X$, $|C| > 1$ be an independently monotonic concept class and $d = VC\text{-Dim}(C)$. Then there is a learning algorithm which learns from membership queries only that requires at most*

$$d(1 + \log_2 \frac{|X|}{d})$$

membership queries in order to determine the correct target concept $c \in C$.

Proof: In the Lemma it has been shown that for any independently monotonic concept class C there are at most d subsets $s_1, \dots, s_d$ of X where each of those subsets can be linearly ordered by the dominance relation in C . That means, for an arbitrary concept $c \in C$ the first $k_i(c) \in \mathbb{N}$ elements of a subset s_i are positive while the remaining objects in s_i are negative examples of c . Furthermore, determining the objects of these subsets to be positive or negative examples of the unknown target concept c_t means, determining exactly one concept $c \in C$. In other words, there is only one concept in C consistent with the determined positive and negative examples. These subsets are linearly ordered by the dominance relation in C . Therefore, a binary search procedure is executable on each of these subsets $s_1, \dots, s_d$ to find out which objects are positive examples and which ones are negative examples of c_t . Hence, in each of the d linearly ordered subsets $s_1, \dots, s_d$ of X a binary search for the most objects in s_i dominating positive example p can be executed. For this binary search procedure there are at most $(\log_2 |s_i|) + 1$ queries necessary. Thus, the greatest number of queries will be required if all d subsets have the same cardinality. Therefore, an upper bound for the number of required queries is $d(1 + \log_2 \frac{|X|}{d})$. $\square$

For illustration, Theorem 1 can be applied to both of the given examples for *independently monotonic* concept classes: As already noted, the $VC\text{-Dim}(C)$ for C representing either the set of linear decision functions or the set of onesided conjunctive threshold functions, is given by the length n of the attribute-value vector of the set of objects X . Moreover, the size of the set X is given by j^n . Thus Theorem 1 states the following upper bounds for the number of required membership queries.

2.3 Providing hints

Suppose $X = \{1, \dots, m\}$ and the concept class C contains all singletons of X , i.e. all sets which cover only a single object in X . In this case, there is a worst case lower bound on the required number of membership queries of $m - 1$. I.e., a (nearly) exhaustive search is necessary. (See [1] for proofs of lower bounds for various types of queries.) Nevertheless, in certain cases the number of required membership queries can be reduced *dramatically* if initially the learning system is provided with one positive example of the target concept c_t . Think of X being the set of coordinates in a grid based plane. That is, $X = \{(0, 0), (0, 1), \dots, (m-1, m-1)\}$ and C being the set of all orthogonal rectangles in this grid based plane. In that case, C includes among other sets all singleton sets of X . Thus, a lower bound for required membership queries is $m^2 - 1$. For illustration, see Figure 2.

Assume, one gets an arbitrary positive example p of the target concept c_t and the concept class C is purged from all concepts not consistent with p . Then, the remaining concept class C' is independently monotonic. Thus, only $d(1 + \log_2 \frac{|X|}{d})$ with $d = VC\text{-Dim}(C)$ membership queries are necessary in order to determine the target concept c_t . This observation can be generalized by characterizing the appropriate concept classes through the following definition:

Definition 5 *A concept class C over a set of objects X is **k-independently monotonic** iff for all objects $x \in X$ there is a concept $c \in C$ covering x and $|c| \leq k$ and the set $M(c)$ of minimal superconcepts of c in C is independent in C .*

Now, the next Theorem can be formulated:

Theorem 2 *Let X be a set of objects, $C \subseteq 2^X$, $|C| > 1$ be a **k-independently monotonic** set of concepts and $d = VC\text{-Dim}(C)$. Then, after initially providing the learning system with k different positive examples of the target concept c_t there is a learning algorithm which needs at most $d(1 + \log_2 \frac{|X|}{d})$ membership queries in order to determine the correct target concept $c_t \in C$.*

Proof: Let $s = \{x_1, \dots, x_k\}$ be the set of objects given as positive examples of the concept c_t . By definition of C being **k-independently monotonic** there is a concept $c \in C$ such that $c \subseteq s$. Let C' be the remaining concept class after removing all concepts in C not covering c . Thus, there is a reduced concept class C' over a reduced set of objects $X' = (X \setminus c)$. Especially, C' contains an empty set, i.e. the original concept c after removing the objects of c from X . And by definition of C being **k-independently monotonic** C' over X' is independently monotonic and Theorem 1 can be applied to the reduced concept class C' . $\square$

Example: We follow Figure 2, where C is the set of all orthogonal rectangles in a grid based finite plane. In the generalized case of an n -dimensional euclidean space, Theorem 2 can be used to upper bound the number of membership queries required for this generalized concept class C : There is $\text{VC-Dim}(C)=2n$ and $|X|=j^n$ where j is the number of gridpoints along each dimension.

Then Theorem 2 says that at most $2n(1 + n \log_2 j - \log_2 2n)$ membership queries are necessary in order to determine the target concept $c \in C$ after initially getting one positive example of the target concept.

2.4 Meta queries

We give just a formal and general description of our notion of *meta queries*. The idea is illustrated in section 3.

Definition 6 Let $C \subseteq 2^X$ be a concept class on X . Then, a **meta query** is a specified subset $X_S \subseteq X$. The answer to the meta query is 'no' if the corresponding reduced concept class $C_S = \{c \cap X_S | c \in C\}$ on X_S is 1-independently monotonic. Otherwise, the answer is a set $X_M = \{X_1, \dots, X_n\}$ of mutually disjoint subsets of X_S . Each subset in X_M is required to contain at least one positive example of the target concept.

I.e. each supplied subset in X_M either has to be 1-independently monotonic or it has to have the potential to become splitted into multiple subsets - such that each subset is 1-independently monotonic. Here, the user or expert is asked to provide sets of subsets in a way that the number of subsequent meta queries by the system is minimized.

3 Knowledge Acquisition System KAC-Z

In this section we describe our Knowledge Acquisition System (KAC-Z) which incorporates the theoretical results described in the previous section. An application of KAC-Z in a real world (manufacturing industry) problem situation is being described as well.

3.1 System Design

The goal of this system is to acquire knowledge in order to perform a classification task. For this purpose KAC-Z makes use of goal-directed queries. In particular, membership queries are used. As pointed out in section 2.3 learning disjunctive concepts usually require a vast number of membership queries. This poses a difficult problem, because no expert is ready to answer a vast number of queries. In order to overcome this problem we use in KAC-Z a new query type called *meta queries*. Meta queries were used to tackle the problem of learning intervals within disjunctive classification. The problem is illustrated in Figure 3.

In the geometrical example (Figure 3) the task is to learn two rectangles r_1, r_2 . Assume, our positive example is

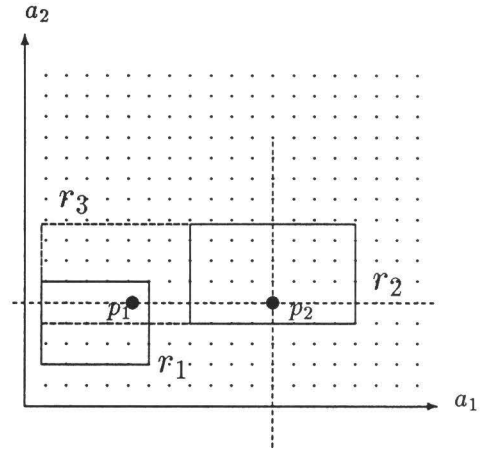


Figure 3: Learning two boxes as disjunctive concept

A	0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
B	0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1
C	0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1
D	0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
$f(A, B, C, D)$	1 1 0 0 0 0 0 0 1 1 0 0 0 1 1 1

Table 2: Equivalent disjunctive boolean functions f, f_1, f_2 of Figure 4.

point p_2 . Then the left boundary could easily be misjudged by the binary search. Reason being that the binary search may produce a membership query for p_1 .

The answer to that would indicate that p_1 is a positive example, which would mean that the left boundary of rectangle r_2 containing p_2 lies at the left of p_1 . In fact the binary search procedure would yield to the dashed rectangle r_3 as learning result. One generally cannot avoid this by using membership queries only, except one uses an exhaustive search. This leads us to the idea of using yet another type of query which we call meta query. Answers to these queries determine whether there are disjunctive terms in the classification rule and whether a binary search can be applied or not. The idea of meta queries is to provide the system KAC-Z with the branching structure of the classification rule tree (Figure 4). Consider an example with learning a disjunctive boolean function f as given in Table 2.

The boolean functions f_1 and f_2 reflect two different tree structures even though both functions are logically equivalent. A branch in the tree represents a disjunction in the corresponding boolean functions. The terminal nodes correspond to a conjunction (eventually intervals) of the remaining attributes. For each of the intermediate nodes the expert is requested to point out the disjunctions (branching) via a corresponding meta query. As can be seen in Figure 4, the tree structure depends upon

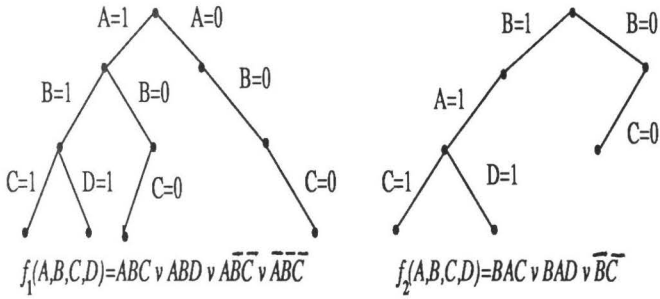


Figure 4: Different tree representations of $f(A, B, C, D) = f_1(A, B, C, D) = f_2(A, B, C, D)$

the order of disjunctions given by the expert. The system starts with a meta query for the root node. The expert's answer has to indicate whether disjunctive expressions are involved in the function to learn. If yes, the expert has to provide an attribute along with at least two disjoint attribute value sets. The given sets are assumed to constitute a relevant and meaningful dichotomy (polychotomy). In Figure 4 the attribute for the root node would be A respectively B . The attribute value sets would be '0' and '1' respectively. In going on, for each emerging node the system uses a further meta query. If the answer indicates that there is no disjunctive structure within the remaining attribute-values, then the node is a terminal node.

We use a frame like description for the objects whose classification should be learnt as shown in section 2. I.e. slot value intervals are efficiently learnt via membership queries. In order to do that, for each interval an upper and a lower bound with the help of a binary search has to be found. For each attribute an initial value which lies within the interval is required. Thus, all required initial values are supplied by an arbitrary single positive example.

Note: The case of irrelevant attributes is included simply if the whole of its value range is recognized as valid.

3.2 Application

KAC-Z is intended to be used in the manufacturing industry. Its goal is to acquire the knowledge for determining appropriate cutting metals for specific cutting operations. Currently this job is performed by a long time experienced expert from the fixture design department. An example of a classification rule tree for our application, where the expert defines a hierarchy of the materials being used in the specific cutting process is shown in Figure 5. In the example above there are non-boolean attributes. In this case similar to boolean attributes multiple-value sets as well as single-value sets can be assigned to a particular branch.

4 Conclusion

The introduced notion of meta queries turned out to be

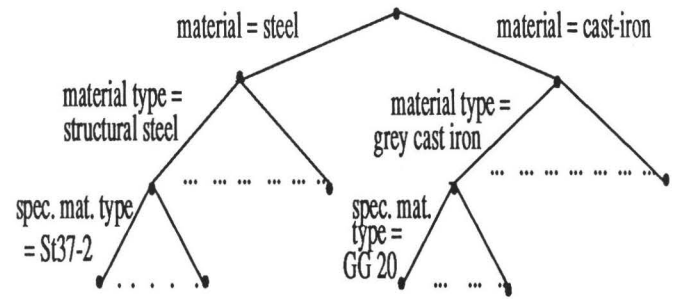


Figure 5: Material hierarchy tree

very valuable for acquiring complex knowledge structures. Our meta queries appear to have overcome the problem of exhaustive search for learning complex concept classes via membership queries. Based on our theoretical results KAC-Z acquires knowledge in the manufacturing domain. However, in future further types of queries will be considered in order to examine their usefulness for supporting the knowledge acquisition.

References

- [1] D. Angluin. Queries and concept learning. *Machine Learning*, 2:319–342, 1988.
- [2] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth. Classifying learnable geometric concepts with the Vapnik-Chervonenkis dimension. In *Proc. 18th STOC*, pp. 273–282, 1986.
- [3] E. Feigenbaum. The art of artificial intelligence I. themes and case studies of knowledge engineering. In *Proc. 5th IJCAI*, pp. 1014–1029, 1977.
- [4] B. R. Gaines. Knowledge acquisition: The continuum linking machine learning and expertise transfer. In *Proc. European Knowledge Acquisition Workshop*, 1989.
- [5] A. G. Hoffmann. Unifying several learning situations. In *Proc. Workshop on Computational Learning Theory*, pp. 415–416, 1988.
- [6] A. G. Hoffmann. Types of efficient query learning. In *Proc. 14th German Workshop on Artificial Intelligence*, pp. 257–266, 1990.
- [7] C. Sammut and R. Banerji. Learning concepts by asking questions. In R. S. Michalski, J. G. Carbonell, and Mitchell, T. M., ed., *Machine Learning: An Artificial Intelligence Approach*, vol. 2, pp. 167–191. Morgan Kaufmann Publ., 1986.
- [8] V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971.