Queries Revisited

Dana Angluin

Computer Science Department
Yale University
P. O. Box 208285
New Haven, CT 06520-8285
angluin@cs.yale.edu

Abstract. We begin with a brief tutorial on the problem of learning a finite concept class over a finite domain using membership queries and/or equivalence queries. We then sketch general results on the number of queries needed to learn a class of concepts, focusing on the various notions of combinatorial dimension that have been employed, including the teaching dimension, the exclusion dimension, the extended teaching dimension, the fingerprint dimension, the sample exclusion dimension, the Vapnik-Chervonenkis dimension, the abstract identification dimension, and the general dimension.

1 Introduction

Formal models of learning reflect a variety of differences in tasks, sources of information, prior knowledge and capabilities of the learner, and criteria of successful performance. In the model of exact identification with queries [1], the task is to identify an unknown concept drawn from a known concept class using queries to gather information about the unknown concept. The two most studied types of queries are membership queries and equivalence queries. In a membership query, the learner asks if a particular domain element is included in the unknown concept or not. In an equivalence query, the learner proposes a particular concept, and is told either that the proposed concept is the same as the unknown concept, or is given a counterexample, that is, a domain element that is classified differently by the proposed concept and the unknown concept. If there are several possible counterexamples, the choice of which one to present is generally assumed to be made adversarially.

Researchers have invented a wonderful variety of ingenious and beautiful polynomial-time learning algorithms that use queries to achieve exact identification of different classes of concepts, as well as important modifications of the basic model to incorporate more realism, e.g., background knowledge and errors. However, this survey will focus on the question of how many queries are needed to learn different classes of concepts, ignoring other computational costs. The analogous question in the PAC model [19] is how many examples are needed to learn different classes of concepts. In the case of the PAC model, bounds in terms of a combinatorial property of the concept class called the Vapnik-Chervonenkis

N. Abe, R. Khardon, and T. Zeugmann (Eds.): ALT 2001, LNAI 2225, pp. 12–31, 2001.

dimension early provided a satisfying answer [7,8]. In the case of learning with queries, the development has been both more gradual and more variegated.

2 Preliminaries

The *domain* X is a nonempty finite set. A *concept* is any subset of X, and a *concept class* is any nonempty set of concepts. We ignore the issues of how concepts and domain elements are represented. We distinguish certain useful concept classes: the class 2^X of all subsets of X, and the class $S(X)$ of singleton subsets of X. We also define $S^+(X)$, the class $S(X)$ together with the empty set.

One way to visualize a domain X and a concept class C is as a binary matrix whose rows are indexed by the concepts, say $c_1, c_2, \ldots, c_M$, and whose columns are indexed by the elements of X, say, $x_1, x_2, \ldots, x_N$, and whose (i, j) entry is 1 if $x_j \in c_i$ and 0 otherwise. An example is given in Figure 1.

	x_1	x_2	x_3
c_1	1	0	1
c_2	0	0	1
c_3	1	1	0
c_4	1	0	0

Fig. 1. Matrix representation of the concept class $C_0 = \{c_1, c_2, c_3, c_4\}$, where $c_1 = \{x_1, x_3\}$, $c_2 = \{x_3\}$, $c_3 = \{x_1, x_2\}$, $c_4 = \{x_1\}$

The rows (representing concepts) are all distinct, though the columns need not be. For our purposes the columns (representing domain elements) may also be assumed to be distinct, because there is no point in distinguishing between elements x and x' that are contained in exactly the same set of concepts. Thus, a domain and concept class can be represented simply as a finite binary relation whose rows are distinct and whose columns are distinct. This makes clear the symmetry of the roles of the domain and the concept class.

For any concept $c \subseteq X$ we define two basic types of queries with respect to c. In a *membership query*, the input is an element $x \in X$, and the output is 1 if $x \in c$ and 0 if $x \notin c$. In an *equivalence query*, the input is a concept $c' \subseteq X$, and the output is either "yes," if $c' = c$, or an element x in the symmetric difference of c and c', if $c' \neq c$. Such an element x is a *counterexample*. The choice of a counterexample is nondeterministic.

A learning problem is specified by giving the domain X, the class of concepts C, and the permitted types of queries. The task of a learning algorithm is to identify an unknown concept c drawn from C using the permitted types of queries. Because we ignore computational resources other than the number of queries, we use decision trees to model learning algorithms.

A *learning algorithm over* X is a finite rooted tree that may have two types of internal nodes. A *membership query node* is labelled by an element $x \in X$ and has two outgoing edges, labelled by 0 and 1. An *equivalence query node* is labelled by a concept $c \subseteq X$ and has $|X| + 1$ outgoing edges, labelled by "yes" and the elements of X. The leaf nodes are unlabelled. An example of a learning algorithm T_0 that uses only membership queries is given in Figure 2.

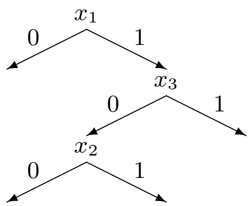

Fig. 2. MQ-algorithm T_0 over domain $X = \{x_1, x_2, x_3\}$

Given a learning algorithm T and a concept class C, we recursively define the *evaluation of* T *on* C as follows. Each node of T will be assigned the subset of C consistent with the answers to queries along the path from the root to that node.

The root node is assigned C itself. Suppose an internal node v has been assigned the subset C' of C. If v is a membership query labelled by x, then the 0-child of v is assigned the subset of C' consisting of concepts c such that $x \notin c$, and the 1-child of v is assigned the subset of C' consisting of concepts c such that $x \in c$. In this case, the set C' is partitioned between the two children of v. If v is an equivalence query labelled by c', then for each $x \in X$, the x-child of v is assigned the subset of C' consisting of concepts c such that x is in the symmetric difference of c' and c. The "yes"-child of v is assigned the singleton $\{c'\}$ if $c' \in C'$, otherwise it is assigned the empty set. In this case, we do not necessarily have a partition; a concept in C' may be assigned to several of the children of v. The assignment produced by evaluation of the tree T_0 on the concept class C_0 is shown in Figure 2.

A learning algorithm T is *successful* for a class of concepts C if in the evaluation of T on C, there is no leaf ℓ of T such that two distinct concepts $c, c' \in C$ are assigned to ℓ. This implies that T has at least $|C|$ leaves, because in the evaluation of T on C each element of C is assigned to at least one leaf, and no two elements of C are assigned to the same leaf of T. It also implies that the decision tree T may be used to identify an unknown concept $c \in C$ by asking queries starting with the root and following the edges corresponding to the answers,

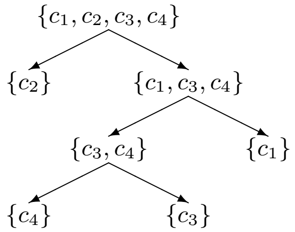

Fig. 3. Assignment produced by evaluation of T_0 from Figure 2 on C_0

until a leaf is reached, at which point exactly one concept $c \in C$ is consistent with the answers received.

Let T be a learning algorithm over X. The *depth of* T, denoted $d(T)$ is the maximum number of edges in any path from the root to a leaf of T. Let $c \subseteq X$ be any concept. The *depth of c in T*, denoted $d(c, T)$, is the maximum number of edges in a path from the root to any leaf assigned c in the evaluation of T on the class $\{c\}$. This is the worst-case number of queries used by the algorithm T in identifying c. Figure 2 shows that T_0 is successful for C_0, and $d(c_4, T_0) = 3$.

3 Membership Queries Only

A *MQ-algorithm* uses only membership queries. The partition property of membership queries implies that every concept is assigned to just one leaf of a MQ-algorithm. If a MQ-algorithm T is successful for a concept class C, then

$$\log |C| \leq d(T), \tag{1}$$

because T is a binary tree with at least $|C|$ leaves.

Let $T_{MQ}(C)$ denote the set of MQ-algorithms T that are successful for C, and have no leaf assigned $\emptyset$ in the evaluation of C. To see that $T_{MQ}(C)$ is nonempty, consider the *exhaustive MQ-algorithm* that systematically queries every element of X in turn. Certainly, no two concepts are assigned to the same leaf, although some leaves may be assigned $\emptyset$. If so, redundant queries may be pruned until every leaf is assigned exactly one concept from C. This MQ-algorithm is successful for every concept class over X. Its depth is at most $|X|$.

Define the *MQ-cost* of a class C of concepts over X, denoted $\#\mathrm{MQ}(C)$, as

$$\#\mathrm{MQ}(C) = \min_{T \in T_{MQ}(C)} \max_{c \in C} d(c, T). \tag{2}$$

Then

$$\log |C| \leq \#\mathrm{MQ}(C) \leq |X|, \tag{3}$$

because any MQ-algorithm successful on C has depth at least $\log |C|$, and the exhaustive MQ-algorithm has depth $|X|$. For the class 2^X, the upper and lower bounds are equal. MQ-algorithms are equivalent to the mistake trees of Littlestone [15].

4 Equivalence Queries Only

Consider the learning algorithm T_1, which uses only equivalence queries over X, presented in Figure 4. The evaluation of T_1 on the concept class C_0 is presented in Figure 5.

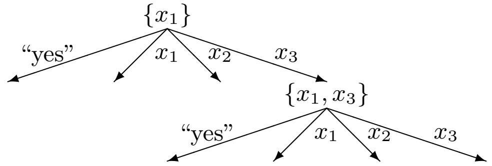

Fig. 4. Equivalence query algorithm T_1

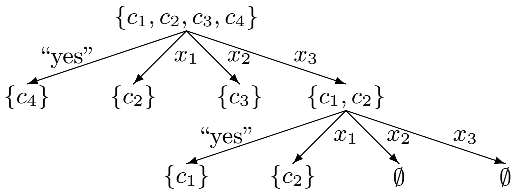

Fig. 5. Evaluation of T_1 on C_0

The algorithm T_1 is successful for C_0, because no two concepts from C_0 are assigned to the same leaf of T_1. The concept c_2 is assigned to two different leaves of T_1, illustrating the non-partition property of equivalence queries.

Given a concept class C, a *proper equivalence query* with respect to C is an equivalence query that uses an element $c \in C$. We use the notation EQ for equivalence queries proper with respect to a class C, and XEQ for *extended equivalence queries*, which are unrestricted. A useful generalization allows equivalence queries from a hypothesis class H containing C, but for simplicity we do not pursue that option. The equivalence queries in T_1 involve only concepts that are elements of C_0, namely $c_4 = \{x_3\}$ and $c_1 = \{x_1, x_3\}$. Consequently, we say that T_1 is an *EQ-algorithm for* C_0.

Given a concept class C, let $T_{EQ}(C)$ denote the set of EQ-algorithms successful for C, and let $T_{XEQ}(C)$ denote the set of XEQ-algorithms successful for C. Clearly, $T_{EQ}(C) \subseteq T_{XEQ}(C)$. To see that $T_{EQ}(C)$ is nonempty, consider the *exhaustive EQ-algorithm for* C, which consists of making an equivalence query with every element of C, except one, in some order. This gives an EQ-algorithm of depth $|C|-1$ that is successful for C.

Define

$$\#\mathrm{EQ}(C) = \min_{T \in T_{EQ}(C)} \max_{c \in C} d(c,T), \tag{4}$$

and

$$\#\mathrm{XEQ}(C) = \min_{T \in T_{XEQ}(C)} \max_{c \in C} d(c,T). \tag{5}$$

For every concept class C,

$$\#\mathrm{XEQ}(C) \leq \#\mathrm{EQ}(C) \leq |C|-1. \tag{6}$$

For the class of singletons, $S(X)$, a simple adversary argument shows that $\#\mathrm{EQ}(S(X)) = |X|-1$, attaining the upper bound above. For the same class, a single XEQ with the empty set discloses the identity of the target concept, therefore, $\#\mathrm{XEQ}(S(X)) = 1$.

5 Membership and Equivalence Queries

Algorithms may involve both membership and equivalence queries. We distinguish MQ&EQ-algorithms, in which all the equivalence queries are proper for the concept class under consideration, from MQ&XEQ-algorithms, in which there is no restriction on the equivalence queries. Figure 6 shows T_2, a MQ&EQ-algorithm that is successful for the class C_0.

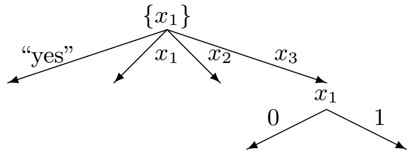

Fig. 6. MQ&EQ-algorithm T_2 for the concept class C_0

Let $T_{MQ\&EQ}(C)$ denote the set of MQ&EQ-algorithms successful for the concept class C. Define

$$\#\mathrm{MQ\&EQ}(C) = \min_{T \in T_{MQ\&EQ}(C)} \max_{c \in C} d(c,T). \tag{7}$$

For any concept class C, because MQ&EQ-algorithms have both types of queries available, we have the following inequalities.

$$\#\mathrm{MQ\&EQ}(C) \leq \#\mathrm{MQ}(C), \tag{8}$$

and

$$\#\text{MQ\&EQ}(C) \leq \#\text{EQ}(C). \tag{9}$$

For the concept class C, let $T_{MQ\&XEQ}(C)$ denote the set of MQ&XEQ-algorithms successful for C. Define

$$\#\text{MQ\&XEQ}(C) = \min_{T \in T_{MQ\&XEQ}(C)} \max_{c \in C} d(c, T). \tag{10}$$

Clearly,

$$\#\text{MQ\&XEQ}(C) \leq \#\text{MQ\&EQ}(C). \tag{11}$$

6 XEQ's, Majority Vote, and Halving

The first query in the algorithm T_1 is very productive, in the sense that no child of the root is assigned more than half the concepts in C_0. The existence of such a productive query is fortuitous in the case of EQ's, but is guaranteed in the case of XEQ's. In particular, for any class C of concepts over X, we define the *majority vote of* C, denoted $c_m(C)$, as follows.

$$(\forall x \in X)[x \in c_m(C) \leftrightarrow |\{c' \in C : x \in c'\}| > |C|/2]. \tag{12}$$

That is, an element x is placed in $c_m(C)$ if and only if more than half the concepts in C contain x. Thus, any counterexample to the majority vote concept eliminates at least half the possible concepts. The majority vote concept for C_0 is $\{x_1\} = c_4$.

The *halving algorithm for* C may be described as follows. Starting with the root, construct the tree of XEQ's and the evaluation of the tree on C concurrently. If there is a leaf assigned C' and C' has cardinality more than 1, then extend the tree and its evaluation on C by replacing the leaf with an XEQ labelled by the majority vote concept, $c_m(C')$. Because the set of concepts assigned to a node can be no more than half of the concepts assigned its parent, no path in the tree can contain more than $\lfloor \log |C| \rfloor$ XEQ's. Thus, for any concept class C,

$$\#\text{XEQ}(C) \leq \lfloor \log |C| \rfloor. \tag{13}$$

The halving algorithm is good, but not necessarily optimal [15,16].

7 An Optimal XEQ-Algorithm

Littlestone [15] defines the *standard optimal algorithm*, which achieves an XEQ-algorithm of depth $\#\text{XEQ}(C)$ for any concept class C. He proves the non-obvious result that

$$\#\text{XEQ}(C) = \max_{T \in T_{MQ}(C)} \min_{c \in C} d(c, T). \tag{14}$$

That is, the optimal number of XEQ's to learn a class C is the largest d such that there is a MQ-algorithm successful for C in which the depth of each leaf is at least d.

Maass and Turán [16] use this result to show that

$$\#\mathrm{XEQ}(C)/\log(\#\mathrm{XEQ}(C)+1) \leq \#\mathrm{MQ\&XEQ}(C). \tag{15}$$

This shows that the addition of MQ's cannot produce too much of an improvement over XEQ's alone.

To prove this, let $d = \#\mathrm{XEQ}(C)$ and consider a MQ-algorithm T successful for C such that every leaf is at depth at least d. Let V denote the set of nodes v at depth d in T such that a concept $c \in C$ assigned to a descendant of v is consistent with the all the replies to queries so far. Initially, V contains 2^d nodes.

An adversary answers MQ's and XEQ's so as to preserve at least a fraction $1/(d+1)$ of V as follows. For a membership query with element x, if at least half the current elements would be preserved by the answer 1, then answer 1, else answer 0. For an equivalence query with the concept c' (not necessarily in C), consider the node v at depth d in T that c' is assigned to. If we consider the d nodes that are siblings of nodes along the path from the root to v, at least one of them, say v' must account for a fraction of at least $1/(d+1)$ of the current elements of V. If the label on the parent of v' is x, then answer the equivalence query with x. Thus, after j queries, there are at least $2^d/(d+1)^j$ elements left in V. Thus, the adversary forces at least $d/\log(d+1)$ queries.

8 Dimensions of Exact Learning

In this section we consider some of the dimensions introduced to bound the cost of learning a concept class with various combinations of queries. For some of these we have suggested different names, to try to bring out the relationships between these definitions.

8.1 The Teaching Dimension

Given a concept class C and a concept $c \in C$, a *teaching set* for c with respect to C is a set $S \subseteq X$ such that no other concept in C classifies all the examples in S the same way c does. For example, $\{x_1, x_3\}$ is a teaching set for the concept c_1 with respect to C_0 because no other concept in C_0 contains both elements. Also, $\{x_1\}$ is a teaching set for the concept c_2 with respect to C_0 because every other concept in C_0 contains x_1. If a learner is presented with an unknown concept from C, by making membership queries for each of the elements in a teaching set for c, the learner can verify whether or not the unknown concept is c.

The *teaching dimension* of a concept class C, denoted $\mathrm{TD}(C)$, is the maximum over all $c \in C$ of the minimum size of a teaching set for c with respect to C [10,18]. It is the worst case number of examples a teacher might have to present to a learner of a concept $c \in C$ to eliminate all other possible concepts in C.

Examples. The teaching dimension of C_0 is 2. The teaching dimension of $S(X)$, the set of singletons over X, is 1 because each set contains an element unique to that set. However, the teaching dimension of $S^+(X)$, the singletons together with the empty set, is $|X|$, because the only teaching set for the empty set in this situation is X itself.

In terms of MQ-algorithms, we have

$$\mathrm{TD}(C) = \max_{c \in C} \min_{T \in T_{MQ}(C)} d(c, T). \tag{16}$$

This is true because the labels on any path from the root to a leaf assigned c in a MQ-algorithm successful for C constitute a teaching set for c with respect to C – any other concept in C must disagree with c on at least one of them, or it would have been assigned to the same leaf as c. Conversely, given a teaching set S for c with respect to C, we can construct a MQ-algorithm that asks queries for those elements first, stopping if the answers are those for c, and continuing exhaustively otherwise. This will produce a MQ-algorithm successful for C in which c is assigned to a leaf at depth $|S|$. Thus the minimization finds the size of the smallest teaching set for a given c, and this is maximized over $c \in C$.

Note that the max and min operations are exchanged in the two equations (2) and (16), and therefore by the properties of max and min,

$$\mathrm{TD}(C) \leq \#\mathrm{MQ}(C). \tag{17}$$

(Note that (2), (14) and (16) involve three out of the four possible combinations of max, min, $c \in C$, and $T \in T_{MQ}(C)$.)

8.2 The Exclusion Dimension

The teaching dimension puts a lower bound on the number of examples a teacher may need to convince a skeptical student of the identity of a concept in C. What about concepts not in C? For a concept $c' \notin C$, how many examples does it take to prove that the concept is not in C? For technical reasons, we consider a slightly different notion, namely, the number of examples to reduce to at most one the set of concepts in C that agree with c' on the examples.

If C is a class of concepts and c' an arbitrary concept, then a *specifying set* for c' with respect to C is a set S of examples such that at most one concept $c \in C$ agrees with the classification of c' for all the elements of S. If $c' \in C$, then a specifying set for c' with respect to C is just a teaching set for c' with respect to C.

Suppose $c' \notin C$ and suppose S is a specifying set for c' with respect to C. There are two possibilities: either there is no concept $c \in C$ that agrees with the classification of c' for every example in S, or there is exactly one such concept $c \in C$. If there is one such, say c, we can add to S a single example on which c' and c disagree to construct a set S' such that no concept in C agrees with the classification of c' on every example in S'. Thus, a specifying set may require at most one more example to become a "proof" that $c' \notin C$.

Define the *exclusion dimension*, denoted $\mathrm{XD}(C)$, of a concept class C as the maximum over all concepts $c' \notin C$, of the minimum size of any specifying set for c' with respect to C. If $C = 2^X$, define the exclusion dimension of C to be 0.

This is the same as the unique specification dimension of Hegedüs [12] and the certificate size of Hellerstein *et al.* [13].

Examples. $\mathrm{XD}(S(X)) = |X| - 1$ because for the empty set we must specify $|X|-1$ examples as not belonging to the empty set to reduce the possible concepts to at most one (the singleton containing the element not specified.) However, for $|X| \geq 2$, $\mathrm{XD}(S^+(X)) = 1$ because any concept not in $S^+(X)$ contains at least two elements, and specifying that one of them belongs to the concept is enough to rule out the empty set and all but one singleton subset of X. We have $\mathrm{XD}(C_0) = 1$, because each of the concepts not in C_0 has a specifying set of size 1. For example, the empty set has a specifying set $\{x_1\}$ with respect to C_0, because only c_2 also does not include x_1, and the set $\{x_1, x_2, x_3\}$ has a specifying set $\{x_2\}$ with respect to C_0, because only c_3 also includes x_2.

The argument for (16) generalizes to give

$$\mathrm{XD}(C) = \max_{c' \notin C} \min_{T \in T_{MQ}(C)} d(c', T). \tag{18}$$

Let T be any MQ-algorithm that is successful for C. Consider any concept $c' \notin C$, the leaf ℓ of T that c' is assigned to, and the set S of elements queried on the path from the root to ℓ. Because at most one element of C is assigned to ℓ, S is a specifying set for c'.

Conversely, if $c' \notin C$ and S is a specifying set for c', then we may construct a MQ-algorithm successful for C by querying the elements of S. If an answer disagrees with the classification by c', then continue with the exhaustive MQ algorithm. If the answers for all the elements of S agree with the classifications by c', then there is at most one concept in C consistent with those answers, and the algorithm may halt.

Hence, the smallest specifying set for c' has size equal to the minimum depth of c' in any MQ-tree successful for C, and (18) follows.

Also, for any concept class C,

$$\mathrm{XD}(C) \leq \#\mathrm{MQ}(C). \tag{19}$$

Consider any MQ-tree of depth $\#\mathrm{MQ}(C)$ that is successful for C. Every $c' \notin C$ has a specifying set consisting of the elements queried along the path in T that c' is assigned to, which is therefore of size at most $\#\mathrm{MQ}(C)$.

8.3 The Extended Teaching Dimension

The combination of the teaching dimension and the exclusion dimension yields the extended teaching dimension [12]. The *extended teaching dimension* of a concept class C, denoted $\mathrm{XTD}(C)$, is the maximum over all concepts $c' \subseteq X$, of the minimum size of any specifying set for c' with respect to C. Clearly, for any concept class C,

$$\mathrm{XTD}(C) = \max\{\mathrm{TD}(C), \mathrm{XD}(C)\}. \tag{20}$$

From (16) and (18) we have

$$\mathrm{XTD}(C) = \max_{c \in 2^X} \min_{T \in T_{MQ}(C)} d(c, T). \tag{21}$$

From (17) and (19), we have

$$\mathrm{XTD}(C) \leq \#\mathrm{MQ}(C). \tag{22}$$

Examples. $\mathrm{XTD}(C_0) = 2 = \max\{2, 1\}$. If $|X| \geq 2$, $\mathrm{XTD}(S(X)) = |X| - 1$ and $\mathrm{XTD}(S^+(X)) = |X|$.

9 The Testing Perspective

In the simplest testing framework there is an unknown item, for example, a disease, and a number of possible binary tests to perform to try to identify the unknown item. There is a finite binary relation between the possible items and the possible tests; performing a test on the unknown item is analogous to a membership query, and adaptive testing algorithms correspond to MQ-algorithms. Hence the applicability of Moshkov's results on testing to questions about MQ-algorithms. The frameworks are not completely parallel. Moshkov introduces the analog of equivalence queries for the testing framework [17].

We take a brief excursion to consider the computational difficulty of the problem of constructing an optimal testing algorithm (or, equivalently, MQ-algorithm.) There is a natural (and expensive) dynamic programming method for constructing an optimal MQ-algorithm. Hyafil and Rivest show that it is NP-complete to decide, given a binary relation and a depth bound, whether the relation has a MQ-algorithm with at most that depth [14]. Arkin *et al.* [3] consider this problem in the context of the number of probes needed to determine which one of a finite set of geometric figures is present in an image. They prove an approximation result for the natural (and efficient) greedy algorithm for this problem, which we now describe.

An MQ-algorithm and its evaluation on C are constructed top-down and simultaneously. For each leaf node assigned more than one concept from C, choose a membership query that partitions the set of concepts assigned to the node as evenly as possible, and extend the tree and its evaluation until every leaf node is assigned exactly one concept from C. Arkin *et al.* show that this method achieves a tree whose height is within a factor of $\lceil \log |C| \rceil$ of the optimal height. (This greedy tree-construction method is a standard one in the literature of constructing decision trees from given example classifications, although decision trees compute classifications rather than identifications.)

10 XTD and MQ-Algorithms

Using a specifying set S for a concept c', we can replace an equivalence query with c' by a sequence of membership queries with the elements of S as follows. If a membership query with x gives an answer different from the classification by c', we proceed as though the equivalence query received counterexample x in reply. If the answers for all the elements of S are the same as the classifications by c', then at most one element of C is consistent with all these answers, and the learning algorithm can safely halt.

If we apply this basic method to replace each XEQ of the halving algorithm by a sequence of at most XTD(C) MQ's, we get the following for any concept class C.

$$\#\text{MQ}(C) \leq (\text{XTD}(C)) \cdot (\lfloor \log |C| \rfloor). \tag{23}$$

We could instead replace each XEQ in the standard optimal algorithm by a sequence of at most XTD(C) MQ's to obtain

$$\#\text{MQ}(C) \leq (\text{XTD}(C)) \cdot (\#\text{XEQ}(C)). \tag{24}$$

Hegedüs [12] gives an improvement over (23), achieved by an algorithm with a greedy ordering of the MQ's used in the simulation of one XEQ.

$$\#\text{MQ}(C) \leq (2\text{XTD}(C)/(\log \text{XTD}(C))) \cdot (\lfloor \log |C| \rfloor). \tag{25}$$

He also gives an example of a family of concept classes for which this improved bound is asymptotically tight.

These results give a reasonably satisfying characterization of the number of membership queries needed to learn a concept class C in terms of a combinatorial parameter of the class, the extended teaching dimension, XTD(C). The factor of roughly $\log |C|$ difference between the lower bound and the upper bound may be thought of as tolerably small, being the number of bits needed to name all the concepts in C. Analogous results are achievable for algorithms that use MQ's and EQ's and for algorithms that use EQ's alone.

11 XD and MQ&EQ-Algorithms

Generalizing Moshkov's results, Hegedüs [12] bounds the number of MQ's and EQ's needed to learn a concept class in terms of the exclusion dimension. Independently, Hellerstein *et al.* [13], introduce the idea of polynomial certificates to characterize learnability with a polynomial number of MQ's and EQ's.

For any concept class C,

$$\text{XD}(C) \leq \#\text{MQ\&EQ}(C) \leq (\text{XD}(C)) \cdot (\lfloor \log |C| \rfloor). \tag{26}$$

An adversary argument establishes the lower bound. Let $c' \notin C$ be any concept such that the minimum specifying set for c' has size $d = \#\text{MQ\&EQ}(C)$. An adversary can answer any sequence of at most $(d-1)$ MQ's and EQ's as though the target concept were c'. (Note that because EQ's must use concepts in C, there cannot be an equivalence query with c' itself.) At this point, there must be at least two concepts in C consistent with the answers given, so a successful learning algorithm must ask at least one more query.

The upper bound is established by a simulation of the halving algorithm. If an XEQ is made with concept c', then if $c' \in C$, it is already an EQ and need not be replaced. If $c' \notin C$, then we take a minimum specifying set S for c' with respect to C and replace the XEQ by MQ's about the elements of S, as described in Section 10.

Using the standard optimal algorithm instead of the halving algorithm gives the following.

$$\#\mathrm{MQ}\&\mathrm{EQ}(C) \leq (\mathrm{XD}(C)) \cdot (\#\mathrm{XEQ}(C)). \tag{27}$$

Again Hegedüs improves the upper bound of (26) by making a more careful choice of the ordering of MQ's, and gives an example of a family of classes for which the improved bound is asymptotically tight.

$$\#\mathrm{MQ}\&\mathrm{EQ}(C) \leq (2\mathrm{XD}(C)/(\log \mathrm{XD}(C))) \cdot (\lfloor \log |C| \rfloor). \tag{28}$$

The key difference in the bounds for MQ-algorithms and MQ&EQ-algorithms is that with both MQ's and EQ's, we do not need to replace an XEQ with a concept $c \in C$, so only the specifying sets for concepts *not* in C matter, whereas with only MQ's we may need to simulate XEQ's for concepts in C, so specifying sets for *all* concepts may matter.

12 A Dimension for EQ-Algorithms?

Can we expect a similar characterization for learning a class C with proper equivalence queries only? The short answer is yes, but the story is a little more complicated.

We'll need samples as well as concepts. A *sample* s is a partial function from X to $\{0, 1\}$. A sample may also be thought of as a subset of elements of X and their classifications, or a function from X to $\{0, 1, *\}$, with $*$ standing for "undefined." If we identify a concept c with its characteristic function, mapping X to $\{0, 1\}$, then a concept is a special case of a sample. Two samples are *consistent* if they take the same values on the elements common to both of their domains. A sample s' *extends* a sample s if they are consistent and the domain of s is a subset of the domain of s'.

It is interesting to note that the partial equivalence queries of Maass and Turán [16] can be characterized as equivalence queries with samples instead of just concepts.

12.1 The Fingerprint Dimension

Early work on lower bounds for equivalence queries introduced the property of *approximate fingerprints* [2], which is sufficient to guarantee that a family of classes of concepts cannot be learned with a polynomial number of EQ's. This technique was applied to show that there is no polynomial-time EQ-algorithm for finite automata, DNF formulas, and many other classes of concepts.

Gavaldà [9] proved that a suitable modification of the negation of the approximate fingerprint property is both necessary and sufficient for learnability with a polynomial number of proper equivalence queries. Hayashi *et al.* [11] generalized the definitions to cover combinations of various types of queries. Stripped of details not relevant to this development, the ideas may be formulated as follows.

If C is a concept class, $c \in C$, and d is a positive integer, then we define c to be *$1/d$-good* for C if for every $x \in X$, a fraction of at least $1/d$ of the concepts in C agree with the classification of x by c. This idea generalizes the majority vote concept for a class C, which is $1/2$-good for C. If we make an EQ with a concept c that is $1/d$-good for C, then any counterexample must eliminate a fraction of at least $1/d$ of the concepts in C.

Given a concept class C, we say that $C' \subseteq C$ is *reachable* from C if there exists a sample s such that C' consists of all those concepts in C that are consistent with s. Not every subclass of a concept class is necessarily reachable.

Examples. For $C = S^+(X)$, the subclasses $\{\{x\}\}$ are reachable (using the sample $s = \{(x, 1)\}$), and subclasses consisting of $S^+(Y)$ for $Y \subseteq X$ are reachable (using a sample that maps the elements of $X - Y$ to 0), but the subclass $S(X)$, consisting of the singletons, is not reachable.

Given a concept class C, the *fingerprint dimension* of C, denoted $\mathrm{FD}(C)$, is the least positive integer d such that for every reachable subclass C' of C, there is a concept $c' \in C'$ that is $1/d$-good for C'.

To see that $\mathrm{FD}(C)$ is well-defined, note that for any concept class C and any concept $c \in C$, c is at least $1/|C|$-good for C, because c at least agrees with itself. A concept class C containing only one concept has $\mathrm{FD}(C) = 1$, but any concept class C containing at least two concepts has $\mathrm{FD}(C) \geq 2$.

We now show that the fingerprint dimension gives bounds on the number of EQ's necessary to learn a class of concepts for any class C of concepts, as follows.

$$\mathrm{FD}(C) - 1 \leq \#\mathrm{EQ}(C) \leq \lceil \mathrm{FD}(C) \ln |C| \rceil. \tag{29}$$

If C has only one concept, then $0 = \mathrm{FD}(C) - 1 = \#\mathrm{EQ}(C)$, so both inequalities hold in this case. Assume C has at least two concepts, and let $d = \mathrm{FD}(C)$. Clearly $d \geq 2$.

We describe a learning algorithm to achieve the upper bound. At any point, there is a class C' reachable from C that is consistent with the answers to all the queries made so far. If C' contains one element, then the algorithm halts. Otherwise, by the definition of $\mathrm{FD}(C)$ there is a concept $c' \in C'$ that is $1/d$-good for C', and the algorithm makes an EQ with this concept c'.

Either the answer is "yes," or a counterexample x eliminates a fraction of at least $1/d$ of the concepts in C'. This continues until exactly one concept $c \in C$ is consistent with all the answers to queries. Then i queries are sufficient if $(1 - 1/d)^i |C| \leq 1$. Hence, $\lceil d \ln |C| \rceil$ EQ's suffice.

For the lower bound, because d is a minimum, there is a reachable subclass C' of C that has no $1/(d-1)$-good concept. For this to be true, $|C'| \geq d$. Thus, for each concept $c' \in C'$, there exists an element $x \in X$ such that the fraction of concepts in C' that agree with the classification of x by c' is smaller than $1/(d-1)$. (This x could be termed a $1/(d-1)$-approximate fingerprint for c' with respect to C'.)

Let s be the sample that witnesses the reachability of C' from C. That is, C' consists of those elements of C that are consistent with s. We describe an adversary to answer EQ's for C that maintains a fraction of at least $(d-i-1)/(d-1)$ of the concepts in C' consistent with the answers to the first i EQ's.

This is clearly true when $i = 0$. For an EQ with $c \in C$, if $c \notin C'$, then c must not be consistent with s, and the adversary returns as a counterexample any element x such that s and c classify x differently. If $c \in C'$, then by our choice of C', there is an element x such that the fraction of elements of C' that classify x the same way as c is smaller than $1/(d-1)$. The adversary returns any such x as a counterexample. Queries of the first type do not eliminate any elements of C', and queries of the second type eliminate fewer than $(1/(d-1))|C'|$ elements of C', so after $d-2$ EQ's, there are at least

$$|C'|/(d-1) > 1$$

concepts in C' consistent with all the answers the adversary has given. Hence, any EQ-algorithm must use at least $d-1$ EQ's, establishing the lower bound.

12.2 The Sample Exclusion Dimension

Balcázar *et al.* introduce the strong consistency dimension [6], which also yields bounds on the number of EQ's to learn a concept class. We give a slight variant of that definition, which generalizes the exclusion dimension from concepts to samples.

Let C be a concept class and s a sample. A *specifying set* for s with respect to C is a set S contained in the domain of s such that at most one concept $c \in C$ is consistent with the sample s' obtained by restricting s to the elements of S. Note that this coincides with our previous definition of a specifying set if s is itself a concept.

Define the *sample exclusion dimension* of a class C of concepts, denoted $\mathrm{SXD}(C)$, to be the maximum over all samples s such that s is not consistent with any $c \in C$, of the minimum size of any specifying set for s. This generalizes the exclusion dimension from concepts not in C to samples not consistent with any concept in C. For $C = 2^X$ we stipulate that $\mathrm{SXD}(C) = 0$.

Because the maximization is over samples and not just concepts, for any class of concepts C,

$$\mathrm{XD}(C) \leq \mathrm{SXD}(C). \tag{30}$$

This differs from the strong consistency dimension introduced by Balcázar *et al.* [6] by at most 1, and coincides, in the case of equivalence queries, with the abstract identification dimension, also introduced by Balcázar et al. [4].

Examples. To get a sense of the difference between the exclusion dimension and the sample exclusion dimension, consider the concept class C_1, presented in Figure 7. This is a version of *addressing*, described by Maass and Turán [16].

The empty set is not an element of C_1, but it has a specifying set $\{x_1, x_2\}$, because only c_1 also does not include either x_1 or x_2. However, the sample

$$s = \{(y_1, 0), (y_2, 0), (y_3, 0), (y_4, 0)\},$$

	x_1	x_2	y_1	y_2	y_3	y_4
c_1	0	0	1	0	0	0
c_2	0	1	0	1	0	0
c_3	1	0	0	0	1	0
c_4	1	1	0	0	0	1

Fig. 7. Concept class C_1, a version of *addressing*

which is not defined for x_1 and x_2, is not consistent with any element of C_1, but its smallest specifying sets have 3 elements, for example, $\{y_1, y_2, y_3\}$. Generalizing this example to 2^n concepts with n address bits gives an exponential disparity between the exclusion dimension and the sample exclusion dimension.

The sample exclusion dimension is a lower bound on the number of EQ's needed to learn a concept class C. For any concept class C,

$$\mathrm{SXD}(C) \leq \#\mathrm{EQ}(C). \tag{31}$$

If $C = 2^X$, then $\mathrm{SXD}(C) = 0$ and the bound holds, so assume $C \neq 2^X$. We describe an adversary to enforce at least $d = \mathrm{SXD}(C)$ EQ's. Let s be a sample that is not consistent with any $c \in C$ such that the size of the smallest specifying set for s with respect to C has size d. Any EQ with a concept $c \in C$ can be answered with an element x in the domain of s, because s is not consistent with any $c \in C$. Up to $(d-1)$ EQ's can be answered thus, and there will still be at least two concepts in C consistent with all the answers given, so any successful learning algorithm must make at least one more EQ.

Combining (29) and (31), we have

$$\mathrm{SXD}(C) \leq \#\mathrm{EQ}(C) \leq \lceil \mathrm{FD}(C) \ln |C| \rceil. \tag{32}$$

The sample exclusion dimension also gives an upper bound on the fingerprint dimension.

$$\mathrm{FD}(C) \leq \mathrm{SXD}(C) + 1. \tag{33}$$

If C contains only one concept, then $\mathrm{FD}(C) = 1$ and $\mathrm{SXD}(C) = 0$, and the bound holds. Assume that C contains at least two concepts, and let $d = \mathrm{SXD}(C)$. Clearly $d \geq 1$. Consider any subclass C' reachable from C, and let s be the sample that witnesses the reachability of C'. That is, C' is the set of concepts in C consistent with s. We show that C' contains a concept c' that is $1/(d+1)$-good for C'.

Define another sample s' as follows. Let $s'(x) = 1$ if a fraction of more than $d/(d+1)$ concepts in C' contain x, and let $s'(x) = 0$ if a fraction of more than $d/(d+1)$ concepts in C' do not contain x. Note that s' is not defined for elements x for which the majority vote of C' does not exceed a fraction $d/(d+1)$ of the total number of elements of C'. Note that s' extends s because all of the elements of C' agree on elements in the domain of s.

We claim that s' is consistent with some element of C. If not, then by the definition of SXD(C), there exists a specifying set S for s' with respect to C that contains at most d elements. Consider the set of elements of C' that are consistent with s' for all the elements of S. Agreement with s' on each element of S eliminates a fraction of less than $1/(d+1)$ of the elements of C', so agreement on all the elements of S eliminates a fraction smaller than $d/(d+1)$ of the elements of C'. Thus, at least one element of C' is consistent with s' on all the elements of S, contradicting the assumption that s' is not consistent with any element in C.

Thus, there is some element $c \in C$ consistent with s', and since s' extends s, $c \in C'$. Thus, the concept c is a $1/(d+1)$-good element of C'. Because C' was an arbitrary reachable subclass of C, we have that $\mathrm{FD}(C) \leq (d+1)$, establishing the bound.

As a corollary of (33) and the upper bound in (29), we have

$$\#\mathrm{EQ}(C) \leq \lceil (\mathrm{SXD}(C) + 1) \ln |C| \rceil. \tag{34}$$

12.3 Inequivalence of FD(C) and SXD(C)

Despite their similar properties in bounding #EQ(C), the two dimensions FD(C) and SXD(C) are different for some concept classes.

Let $X_{2k+1} = \{x_1, x_2, \ldots, x_{2k+1}\}$ and let C_k consist of all subsets of X_{2k+1} of cardinality at most k. Then $|C_k| = 2^{2k}$ and $\ln |C| = \Theta(k)$.

We have $\mathrm{SXD}(C_k) = k$ because the only samples inconsistent with every concept in C must take on the value 1 for at least $k+1$ domain elements, and a minimum specifying set will contain k domain elements with the value 1. On the other hand, $\mathrm{FD}(C_k) = 2$, because every reachable subclass of C_k contains its majority vote concept. Of course, $\#\mathrm{EQ}(C_k) = k$, by a strategy that begins by conjecturing the empty set, and adds positive counterexamples to the conjecture until it is answered "yes."

Thus, for the family of classes C_k, the sample exclusion dimension gives a tight lower bound, k, and a loose upper bound, $O(k^2)$, while the fingerprint dimension gives a loose lower bound, 1, and an asymptotically tight upper bound, $O(k)$, on the number of EQ's required for learning. This is asymptotically as large as the discrepancy can be, as witnessed by (32), which is the combination that gives the strongest bounds on #EQ(C) at present.

13 What about the VC-Dimension?

Because the Vapnik-Chervonenkis dimension is so useful in PAC learning, it is natural to ask what its relationship is to learning with queries. A set $S \subseteq X$ is *shattered* by a concept class C if all $2^{|S|}$ possible labellings of elements in S are achieved by concepts from C. The VC-dimension of a class C of concepts, denoted $\mathrm{VCD}(C)$, is the maximum cardinality of any set shattered by C. It is clear that for any concept class C,

$$\mathrm{VCD}(C) \leq \log |C|. \tag{35}$$

This and (3) imply

$$\mathrm{VCD}(C) \leq \#\mathrm{MQ}(C). \tag{36}$$

As Littlestone [15] observed, an adversary giving counterexamples from a shattered set can enforce $\mathrm{VCD}(C)$ XEQ's, and therefore

$$\mathrm{VCD}(C) \leq \#\mathrm{XEQ}(C) \leq \#\mathrm{EQ}(C). \tag{37}$$

Maass and Turán [16] show that for any concept class C,

$$\frac{1}{7}\mathrm{VCD}(C) \leq \#\mathrm{MQ\&EQ}(C). \tag{38}$$

They give an example of a family of concept classes that shows that the constant $1/7$ cannot be improved to be larger than 0.41, and also show that

$$\frac{1}{7}\mathrm{VCD}(C) \leq \#\mathrm{MQ\&XEQ}(C). \tag{39}$$

14 More General Dimensions

Balcázar *et al.* present generalizations of the dimensions $\mathrm{XTD}(C)$, $\mathrm{XD}(C)$ and $\mathrm{SXD}(C)$ to arbitrary kinds of example-based queries [4], and beyond [5]. It is outside the scope of this sketch to treat their results fully, but we briefly describe the settings. For convenience we identify a concept c with its characteristic function, and write $c(x) = 1$ if $x \in c$.

In [4], for an example-based query with a target concept c, the possible replies are identified with samples consistent with c, that is, with subfunctions of c. Thus, for a membership query about x, the reply is the singleton sample $\{(x, c(x))\}$. For an equivalence query with the concept c', the possible replies are either a counterxample x, which is represented by the sample $\{(x, c(x))\}$, or "yes," which is represented by the sample equal to c, completely specifying it. For a subset query with c', the possible replies are either a counterexample, which is a singleton sample $\{(x, 0)\}$ such that $c'(x) = 1$ and $c(x) = 0$, or "yes," which is represented by the sample consisting of all pairs $(x, 1)$ such that $c'(x) = 1$.

A protocol is a ternary relation on queries, target concepts, and possible answers. Two conditions are imposed on the relation. One is *completeness*, which requires that every possible query and target concept, there is at least one possible answer. The other is *fair play*, which requires that if an answer a is possible for a query q and a target concept c, then for any other target concept c' such that the answer a is a subfunction of c', a is a possible answer for q with target concept c'. The fair play condition ensures that an answer cannot "rule out" a candidate hypothesis unless it is inconsistent with it. For this setting, a very general dimension, the *abstract identification dimension*, is defined and shown to generalize the extended teaching dimension, the exclusion dimension, and the sample exclusion dimension.

In [5], Balcázar *et al.* define an even more general setting, covering many kinds of non-example-based queries. In this setting, the answer to a query is identified with a property that is true of the target concept, or equivalently, a subset of concepts that includes the target concept, or a Boolean function on all possible concepts that is true for the target concept. For example, if the target concept is c, a *restricted equivalence query* with the concept c' returns only the answers "yes" (if $c' = c$) and "no" (if $c' \neq c$), with no counterexample. The reply "yes" can be formalized as the singleton $\{c\}$, specifying c completely, while the reply "no" can be formalized as the set $2^X - \{c'\}$, which gives only the information that $c \neq c'$. In this setting, the authors define the *general dimension* for a target class and learning protocol and prove that the optimal number of queries for the class and the protocol is bounded between this dimension and this dimension times $\lceil \ln |C| \rceil$.

15 Remarks

The approach of bounding the number of queries required to learn concepts from a class C using combinatorial properties of C has made great progress. This sketch has omitted very many things, including the fascinating applications of these results to specific concept classes. One major open problem is whether DNF formulas can be learned using a polynomial number of MQ's and EQ's. The reader is strongly encouraged to consult the original works.

Acknowledgements. Preparation of this paper was supported in part by the National Science Foundation, grant CCR-9610295.

References

1. D. Angluin. Queries and concept learning. *Machine Learning*, 2:319–342, 1988.
2. D. Angluin. Negative results for equivalence queries. *Machine Learning*, 5:121–150, 1990.
3. E. M. Arkin, H. Meijer, J. S. B. Mitchell, D. Rappaport, and S. S. Skiena. Decision trees for geometric models. In *Proceedings of the Ninth Annual Symposium on Computational Geometry*, pages 369–378, San Diego, CA, 1993. ACM Press.
4. J. L. Balcázar, J. Castro, and D. Guijarro. Abstract combinatorial characterizations of exact learning via queries. In *Proceedings of the 13th Annual Conference on Computational Learning Theory*, pages 248–254. Morgan Kaufmann, San Francisco, 2000.
5. J. L. Balcázar, J. Castro, and D. Guijarro. A general dimension for exact learning. In *Proceedings of the 14th Annual Conference on Computational Learning Theory*, 2001.

6. J. L. Balcázar, J. Castro, D. Guijarro, and H.-U. Simon. The consistency dimension and distribution-dependent learning from queries. In *Proceedings of the 10th International Conference on Algorithic Learning Theory - ALT '99*, volume 1720 of *LNAI*, pages 77–92. Springer-Verlag, 1999.
7. A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *J. ACM*, 36:929–965, 1989.
8. A. Ehrenfeucht, D. Haussler, M. Kearns, and L. Valiant. A general lower bound on the number of examples needed for learning. *Inform. Comput.*, 82:247–261, 1989.
9. R. Gavaldà. On the power of equivalence queries. In *EUROCOLT: European Conference on Computational Learning Theory*, pages 193–203. Clarendon Press, 1993.
10. S. A. Goldman and M. J. Kearns. On the complexity of teaching. *J. of Comput. Syst. Sci.*, 50:20–31, 1995.
11. Y. Hayashi, S. Matsumoto, A. Shinohara, and M. Takeda. Uniform characterizations of polynomial-query learnabilities. In *Proceedings of the 1st International Conference on Discovery Science (DS-98)*, volume 1532 of *LNAI*, pages 84–92, 1998.
12. T. Hegedüs. Generalized teaching dimensions and the query complexity of learning. In *Proceedings of the 8th Annual Conference on Computational Learning Theory*, pages 108–117. ACM Press, New York, NY, 1995.
13. L. Hellerstein, K. Pillaipakkamnatt, V. Raghavan, and D. Wilkins. How many queries are needed to learn? In *Proceedings of the Twenty-Seventh Annual ACM Symposium on the Theory of Computing*, pages 190–199, 1995.
14. R. Hyafil and R. L. Rivest. Constructing optimal binary trees is NP-complete. *Information Processing Letters*, 5:15–17, 1976.
15. N. Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2:285–318, 1988.
16. W. Maass and G. Turán. Lower bound methods and separation results for on-line learning models. *Machine Learning*, 9:107–145, 1992.
17. M. Moshkov. Test theory and problems of machine learning. In *Proceedings of the International School-Seminar on Discrete Mathematics and Mathematical Cybernetics*, pages 6–10. MAX Press, Moscow, 2001.
18. A. Shinohara and S. Miyano. Teachability in computational learning. *New Generation Computing*, 8(4):337–348, 1991.
19. L. G. Valiant. A theory of the learnable. *Commun. ACM*, 27:1134–1142, 1984.