Approximate Testing and Its Relationship to Learning

Kathleen Romanik[1]
Center for Automation Research
University of Maryland
College Park, MD 20742

Submitted to Theoretical Computer Science

Abstract

Testing plays an integral part in many areas of computer science. In relation to computational learning theory, testing can be viewed as an inverse process to learning. Testing algorithms create a set of examples for a given target concept that distinguish it from other concepts, while learning algorithms use a given set of examples to correctly infer an unknown concept. In this paper we develop a model for *approximate testing* of concepts, which relates to the PAC (probably almost correct) model of learning as well as other learning models. In approximate testing, a concept that passes the given tests is only required to be correct to within a given error tolerance rather than being exactly correct. We define what it means for a concept class to be *approximately testable*, and we investigate general properties of a concept class that make it testable or untestable. We define a new measure that is similar to the VC-dimension, called the *testing dimension* of a concept class, and show how it yields untestability results for certain concept classes. We also compare our testing model to several different learning models, and we discuss the topics of nonredundant test sets and generic test sets.

[1]This work was completed while the author was a student at University of Maryland, College Park, Maryland as part of her dissertation research. This research was supported in part by NSF Grant CCR-9112976 and ONR Grant N00014-92-J-1254.

1 Introduction

Testing plays a part in many areas of computer science, including hardware and software design and machine learning. In each case some type of object is tested, such as a circuit, a program, or a concept, to determine whether it is correct. Testing theory addresses the question of whether a set of inputs or examples can be given to demonstrate that an object is correct; that is, whether a set of inputs or examples distinguishes a correct object from incorrect ones. Computational learning theory, on the other hand, addresses the question of how to infer a correct hypothesis from a given set of examples of an unknown concept. Thus, testing examines an inverse problem to that of learning – testing algorithms find a set of examples that describe a concept, while learning algorithms find a concept that explains a set of examples.

In this paper we develop a model for *approximate testing* of concepts. In our model, an approximate testing algorithm must be able to use a finite number of examples to distinguish one concept from others that differ from it by more than a given error bound. Our testing model relates to the probably almost correct (or PAC) model of learning introduced by Valiant [Val84]. A PAC learning algorithm learns concepts using finite sets of examples. It differs from other learning models in that concepts are learned to within only a given, probabilistically based, error bound. That is, a PAC learning algorithm is not required to produce a hypothesis that exactly matches the unknown concept that it is trying to learn, but instead it can produce a hypothesis that is "close" to the correct one.

The relationship between testing and learning has been investigated previously by Cherniavsky and Smith [CS87]. However, their approach related testing to Gold's model of learning by identification in the limit.

Our approximate testing model also has relationships with other learning models. It bears some resemblance to the work by Benedek and Itai [BI88] on learning with respect to a fixed probability distribution. However, we show in Section 4 that these two ideas are actually different. It also corresponds directly to the "helpful teacher" learning model [GK95, SDHK91, SM91, ABCST92, JT92]. In this model, the teacher knows a concept and attempts to teach it using the fewest possible examples. The number of examples that a teacher must present for all students to learn a concept is analogous to the notion of the number of test points needed to test a concept. However, our approach differs from the work in [GK95, SM91, ABCST92, JT92] because it deals with uncountable domains, and it differs from the work in [SDHK91, JT92] because they assume a particular strategy on the part of the learner. Also, all of these papers deal for the most part with exact learning, where the hypothesis of a learner is not allowed to have any errors in it, while we study approximate testing, where a concept that tests correctly on a test set is allowed to have some small error. Finally, these papers examine the issue of the size of a set of examples needed to learn a particular class, while this paper deals with properties that determine testability.

Besides being related to various learning models, the study of approximate testing can also be used to obtain lower bounds on certain learning algorithms in the following manner. If it is demonstrated that m test points are needed to test a concept to within some error bound, then a learning algorithm must use at least m examples to insure that its hypothesis is within the error bound.

The remainder of this paper is organized as follows. In Section 2 we give a formal definition of approximate testability. We define four levels of testability – testable, k-testable, two-sided testable and untestable – which yield a testing hierarchy. In Section 3 we give examples from geometry of concept classes at each level, and we examine the hierarchy that these examples define.

In Section 4 we compare our work on approximate testing to the work by Benedek and Itai [BI88] on learning with respect to a fixed probability distribution. We show that the idea of a concept class being approximately testable with respect to a probability distribution P is different from the idea of a concept class being learnable with respect to P.

In Section 5 we discuss the topics of nonredundant test sets and generic test sets. A test set is *nonredundant* if removing any example from it causes it to no longer be a test set. A test set is *generic* if it is a test set for any concept in a given concept class. We prove that there exist testable concept classes for which there are nonredundant test sets of arbitrarily large size and there exist testable concept classes for which no generic test sets exist. We also prove that generic testability implies learnability with respect to a fixed probability distribution.

In Section 6 we investigate properties of a concept class that determine its testability. We define the notion of *testing dimension*, which is similar to Vapnik-Chervonenkis dimension, and show how it yields untestability results for certain concept classes. In Section 7 we summarize our work and mention open problems.

2 Approximate Testing

We use the definition of a concept class from [BEHW89] in our work, so we define it now.

Definition [BEHW89]. A *concept* is a measurable subset q of a set of points X over which a probability measure P is defined. A *concept class* is a set $Q \subseteq 2^X$ of concepts. X is assumed to be a fixed set, usually finite, countable, or E^d (Euclidean d-dimensional space) for some fixed $d \geq 1$. If X is E^d, each $q \in Q$ is a Borel set.

Some examples of concept classes are the class B of all boolean functions on n variables (in this case $X = \{0,1\}^n$) and the class R of closed orthogonal rectangles in E^2. In this paper we use only geometric concept classes defined on E^d in our examples. However, our definitions and results apply to all concept classes.

Each concept class Q has an associated representation class Q_r, which is used when learning or testing a concept from the class. In general, there are several representation classes that can be associated with a concept class, and one of these is chosen when developing a learning or testing algorithm. For example, the class B defined above can be represented by the class of boolean formulas or by the class of circuits. Similarly, a concept in the class R can be represented by its four vertices, or as the cross product of two intervals. In this paper we represent each geometric concept in E^d by a list of vertices that describes it. Once a representation class Q_r is chosen, the class Q is associated with this class, so the class Q_r need not be mentioned explicitly.

When a representation class is used for a concept class, each concept q has associated with it a representation size s_q, which is the size of the shortest representation of q in the representation class. For example, if boolean formulas are used to represent concepts in B, then the size of a concept q could be the number of literals in the shortest formula for q. If a vertex list is used to represent concepts in R then the size of a concept will be 2, since a minimal representation uses two opposite corners to describe an orthogonal rectangle. In this paper, since we represent each concept by a list of vertices, the size of a concept is the number of vertices in its minimal representation list.

We can extend the idea of a representation size for an individual concept to a concept class and define the *size complexity* of a class. This complexity measure indicates how large, in terms of representation, a concept in the class can be. It is useful when examining the testability of classes.

Definition. Given a concept class Q with associated representation class Q_r, the *size complexity* of Q is the largest integer k such that there exists $q \in Q$ with representation size at least k in Q_r. If no such k exists, the size complexity of Q is infinite.

The domain over which concepts are defined has a size parameter as well. The size of the domain for B is n since B contains boolean functions on n variables. The size of the domain E^d is d.

We now give the definition of PAC learnability from Blumer, Ehrenfeucht, Haussler and Warmuth [BEHW89] and define approximate testability in a similar manner. We also define four levels of approximate testability – testable, k-testable, two-sided testable and untestable.

Given two concepts $r, q \in Q$, r is *consistent* with q on some finite set of points $t = \{t_1, t_2, \ldots, t_m\}$ if it agrees with q on all points in t, i.e. if $t \cap q = t \cap r$. The *error* of r, with respect to q and the probability measure P, is given by $P(q \triangle r)$, where $q \triangle r$ denotes the symmetric difference of the sets. Thus, the error of a tested or learned concept is measured as the probability of the region that forms the symmetric difference between it and the target concept.

Let $\mathbf{S_Q}$ denote the set of all finite subsets of X that are labeled according to some $q \in Q$. Let $\mathbf{A}$ be the set of all computable functions $A\colon \mathbf{S_Q} \rightarrow 2^X$. Let I denote the open interval of rationals in $(0,1)$ and let $n\colon I \times I \rightarrow Z^+$ be a positive integer valued function defined on $I \times I$.

Definition[BEHW89]. $A \in \mathbf{A}$ is a *PAC learning algorithm for Q with sample size $n(\epsilon, \delta)$* if for all probability measures P on X and for all $\epsilon, \delta \in I$ and for all $q \in Q$, $A(S_q) = p$, where S_q is a finite random sample (or subset) of X labeled according to q of size $n(\epsilon, \delta)$, and p is consistent with q on S_q and $P(q \triangle p) \leq \epsilon$ with probability greater than $1 - \delta$. If such an A exists, then Q is *PAC learnable.*

Let $\mathbf{S}$ denote the set of all finite subsets of X. Let $\mathbb{Q}^+$ denote the positive rationals and let $m\colon \mathbb{Q}^+ \times I \rightarrow Z^+$ be a positive integer valued function defined on $\mathbb{Q}^+ \times I$. Furthermore, let $\mathbf{T}$ be the set of all computable functions $T\colon Q \times I \rightarrow \mathbf{S}$ and P be a probability measure on X.

Definition. $T \in \mathbf{T}$ is an *approximate testing algorithm for Q with respect to P with test set size $m(s_q, \epsilon)$* if for all $\epsilon \in I$ and for all $q \in Q$, $T(q, \epsilon) = t, |t| \leq m(s_q, \epsilon)$ and for all $r \in Q$, if r is consistent with q on t, then $P(q \triangle r) \leq \epsilon$. If m is a polynomial in s_q (the representation size of q) and $\frac{1}{\epsilon}$ and T runs in polynomial time, then T is a *polynomial approximate testing algorithm for Q with respect to P*. $T(q, \epsilon)$ is called a *test set for q with respect to the class Q*. For each $t_i \in T(q, \epsilon)$, if $t_i \in q$ then t_i is a *positive test point*; otherwise, t_i is a *negative test point.*

Thus given an error bound $\epsilon \in I$ and a target concept $q \in Q$, T produces a test set for q such that any tested concept that is consistent with q on this set has error no more than ϵ. The size of this set may be a function of both ϵ and the representation size of the target concept. If such a T and m exist, then Q is *approximately testable with respect to P with test set size $m(s_q, \epsilon)$*. From now on we will refer to approximately testable concept classes as simply *testable*, and we will not mention P explicitly if the probability measure being used is clear from the context. If T produces a constant size test set (i.e. if m is a constant function k), then we say that Q is *k-testable.* If no testing algorithm T exists for Q, then Q is *untestable.*

Note that a confidence parameter (i.e. δ in the definition of a PAC learning algorithm) is not needed since the selection of test points is deterministic, as opposed to PAC learning algorithms that guess a hypothesis based on randomly chosen examples. PAC learning algorithms use randomly chosen examples because the probability measure on these examples is unknown, so if a deterministic strategy were used, an adversary could choose a probability measure that would cause the strategy to fail.

We also want testing algorithms that reveal something about the error of concepts that are inconsistent with the target concept on a given test set. For this purpose we define two-sided testing algorithms, which are a subset of the set of testing algorithms. Let $e: Q \times I \to \mathbb{Q}^+$ be a positive rational valued function.

Definition. $T \in \mathbf{T}$ is a *two-sided testing algorithm for Q with respect to P with error margin $e(q, \epsilon)$* if T is a testing algorithm for Q with respect to P and for all $\epsilon \in I$ and $q, r \in Q$, if r is inconsistent with q on $T(q, \epsilon)$, then $P(q \triangle r) \geq e(q, \epsilon)$.

Thus if T is a two-sided testing algorithm for Q, then it produces a test set for a target concept q such that any tested concept that is consistent with q on this set has error no more than ϵ and any concept that is inconsistent with q on this set has error at least $e(q, \epsilon)$. If such a T exists, then Q is *two-sided testable with respect to P.* If T is not a two-sided testing algorithm, then a tested concept that is inconsistent with the target concept q on the test set might have an arbitrarily small error.

3 Levels of Testability

In this section we give geometric examples of concept classes at each level of testability, and we use these examples to demonstrate a testing hierarchy.

3.1 Examples

The following examples will help to clarify the distinction between untestable concept classes and the three types of testability – testable, k-testable and two-sided testable – defined above. Each class in the examples below consists of concepts that are subsets of the closed subspace $[0, 1] \times [0, 1]$ in E^2, over which a uniform distribution is defined. In the next section we will use these examples to illustrate a testing hierarchy.

Example 3.1 *Class Q_1 consists of concepts composed of one of the two closed rectangles $[0, \frac{1}{2}] \times [0, \frac{1}{2}]$ or $[\frac{1}{2}, 1] \times [\frac{1}{2}, 1]$ unioned with a finite set of points.*

Proposition 3.1a. Q_1 is 1-testable.

Proof: To obtain a test set with an error bound of ϵ for $q \in Q_1$, simply choose one negative test point inside the rectangle that is not a part of q. For example, if q contains $[0, \frac{1}{2}] \times [0, \frac{1}{2}]$, choose a point in $[\frac{1}{2}, 1] \times [\frac{1}{2}, 1]$ that is not in q. Any other concept $r \in Q_1$ that is consistent with q on the test set must contain the same rectangle as q. Therefore, r only differs from q on a finite set of points, and any finite set of points has probability 0, so r has error $P(q \triangle r) = 0$. $\square$

Proposition 3.1b. Q_1 is not two-sided testable.

Proof: Consider $q \in Q_1$ consisting of only the rectangle $[0, \frac{1}{2}] \times [0, \frac{1}{2}]$. Any testing algorithm T for Q_1 must produce at least one negative test point (x, y) in the rectangle $[\frac{1}{2}, 1] \times [\frac{1}{2}, 1]$ for q whenever $\epsilon < \frac{1}{4}$. The concept $r \in Q_1$ consisting of $[0, \frac{1}{2}] \times [0, \frac{1}{2}]$ and the point (x, y) is then inconsistent with q on the test set but has error $P(q \triangle r) = 0$. Therefore, Q_1 is not two-sided testable. $\square$

Example 3.2 *Class R_1 consists of closed orthogonal rectangles in the region $[0,1] \times [0,1]$. A learning algorithm for this class was given in [BEHW89].*

Proposition 3.2 *R_1 is 6-testable and two-sided testable with an error margin of $e(q, \epsilon) = w\alpha$, where w is the width of the target rectangle and α is a function of w and ϵ.*

Proof: Let $r \in R_1$, with a minimum side of length w, be given. Choose two positive test points near the lower left and upper right corners of r, each a distance α from the edges adjacent to the corner. Choose four negative test points, each midway along a side of r and α away from the side. Let r^- be the minimum area orthogonal rectangle containing the negative test points, and let r^+ be the minimum area orthogonal rectangle containing the positive test points. The value α is chosen such that it is less than $\frac{w}{2}$ and the area of $(r^- \triangle r)$ is less than ϵ.

If a concept $q \in R_1$ is consistent with r on the given test set, then it must contain r^+ and cannot be larger than r^-. Therefore, q has error $P(q \triangle r) < \epsilon$. If a concept $q \in R_1$ is inconsistent with r, then it must have a side that is either at least a distance of α outside r, or a distance of α inside r. In either case, $P(q \triangle r) \geq w\alpha$. $\square$

Example 3.3 *Class Q_P consists of all closed convex polygons in the region $[0,1] \times [0,1]$.*

Proposition 3.3 *Q_P is two-sided testable with a test set of size $2n$ where n is the number of vertices of the target polygon.*

Proof: (Proof Sketch – A complete proof can be found in [Rom92], where it is proven that $2n$ test points are both necessary and sufficient to test an n-sided closed convex polygon.) Given an n-sided closed convex polygon p as a target, choose for each vertex v of p a positive test point that is a distance α from v and that is equidistant from each of the two edges adjacent to v. Next, for each edge of p, choose a negative test point midway along the edge and a distance β from it. Convexity is used to show that when α and β are sufficiently small, these points bound a finite region of the plane that can be occupied by a convex polygon consistent with p on the test set. By decreasing α and β this region can be made arbitrarily small. Convexity is also used to show that this is a two-sided testing algorithm for p. □

Example 3.4 *Class R_2 consists of concepts composed of one or two closed orthogonal rectangles in the region $[0,1] \times [0,1]$.*

Proposition 3.4 *R_2 is testable with a test set of size $\frac{36}{\epsilon^2}$, but it is not k-testable for any k.*

Proof: First we show that R_2 is testable by a testing algorithm that chooses a set of $(\lceil\frac{6}{\epsilon}\rceil - 1)^2$ points distributed evenly over $[0,1] \times [0,1]$. For $r \in R_2$ and $\epsilon > 0$, let $T(r,\epsilon) = \{(\frac{i\epsilon}{6}, \frac{j\epsilon}{6}) \mid 1 \leq i,j \leq (\lceil\frac{6}{\epsilon}\rceil - 1)\}$. The probability of a region bounded by 4 adjacent test points is no greater than $\frac{\epsilon^2}{36}$. If a tested concept $q \in R_2$ is consistent with r on $T(r,\epsilon)$, then q and r can only differ in at most $\frac{36}{\epsilon}$ of these regions. This means $P(q \triangle r) \leq (\frac{36}{\epsilon})(\frac{\epsilon^2}{36}) = \epsilon$.

Now we prove that R_2 is not k-testable for any k by assuming that it is and finding a contradiction. Assume that T is a k-testable testing algorithm for R_2. Consider the target concept $r = [0,1] \times [0,1]$, and let $\epsilon = \frac{1}{k+1}$. There are two cases to consider.

Case 1: There exist two test points $(x_1, y_1), (x_2, y_2) \in [0,1] \times [0,1]$ such that $(x_2 - x_1) > \frac{1}{k+1}$ and there does not exist a test point (x_3, y_3) such that $x_1 < x_3 < x_2$. In this case, choose q to be the two rectangles $[0, x_1] \times [0,1]$ and $[x_2, 1] \times [0,1]$. Then q is consistent with r on $T(r,\epsilon)$, but $P(q \triangle r) > \frac{1}{k+1} = \epsilon$.

Case 2: If Case 1 does not hold, then the minimum area rectangle containing all the test points in $T(r,\epsilon)$ that lie in $[0,1] \times [0,1]$ has x length no more than $\frac{k-1}{k+1}$. Choose q to be this rectangle. (If $T(r,\epsilon)$ contains no points in $[0,1] \times [0,1]$, then choose q to be the rectangle $[0, \frac{k-1}{k+1}] \times [0,1]$.) Then q is consistent with r on $T(r,\epsilon)$, but $P(q \triangle r) \geq 1 - \frac{k-1}{k+1} = \frac{2}{k+1} > \epsilon$. □

Example 3.5 *Class Q_2 consists of concepts composed of an orthogonal rectangle unioned with a finite set of points in the region* $[0,1] \times [0,1]$.

Proposition 3.5 *Q_2 is untestable.*

Proof: Assume that T is a testing algorithm for Q_2. Let $q \in Q_2$ be the rectangle $[0,1] \times [0,1]$, and let $T(q, \frac{1}{2})$ be the finite test set generated by T for q. A concept $r \in Q_2$ that is consistent with q on $T(q, \frac{1}{2})$ but has an error greater than $\frac{1}{2}$ is the rectangle $[0, \frac{1}{2}] \times [0, \frac{1}{2}]$ unioned with the set of positive test points in $T(q, \frac{1}{2})$. The error of r with respect to q is $P(q \triangle r) = \frac{3}{4} > \frac{1}{2}$, so T is not a testing algorithm. □

These five examples illustrate that concept classes that appear similar at first glance may actually have very different levels of testability. In addition, Examples 3.1 and 3.3 demonstrate that classes with infinite size complexity can be testable.

It is interesting to note that in several of these classes, concepts contain finite sets of points, and although these finite sets of points have probability measure 0, they can affect the testability of the class. For example, concepts in R_1 and concepts in Q_2 differ by only a finite set of points. However, R_1 is testable and Q_2 is not. One could argue that concept classes such as Q_2 should not be allowed since a part of the concept with 0 measure determines the testability of the class. However, these classes are very important with respect to testing because they represent classes of concepts where noise occurs in the measurements used to test an unknown concept against a target concept. For example, class Q_2 is a representation of the class R_1 of orthogonal rectangles where there is a possibility of a finite number of negative misclassification errors (i.e. errors where a negative example is misclassified as positive). Examples 3.2 and 3.5 illustrate that a finite number of misclassification errors can cause a testable concept class to become untestable.

3.2 Testing Hierarchy

In Section 2 we defined four different notions of testability – testable, k-testable, two-sided testable, untestable – by which concept classes can be categorized. These definitions yield a hierarchy of testability. By definition, the family of k-testable concept classes and the family of two-sided testable concept classes are both subsets of the family of testable concept classes. However, the rest of the hierarchy differs depending on the probability measure P defined on the space X.

With respect to a uniform distribution defined over a bounded subspace of E^2, we have the hierarchy shown in Figure 1. The concept classes given in subsection 3.1 are shown in

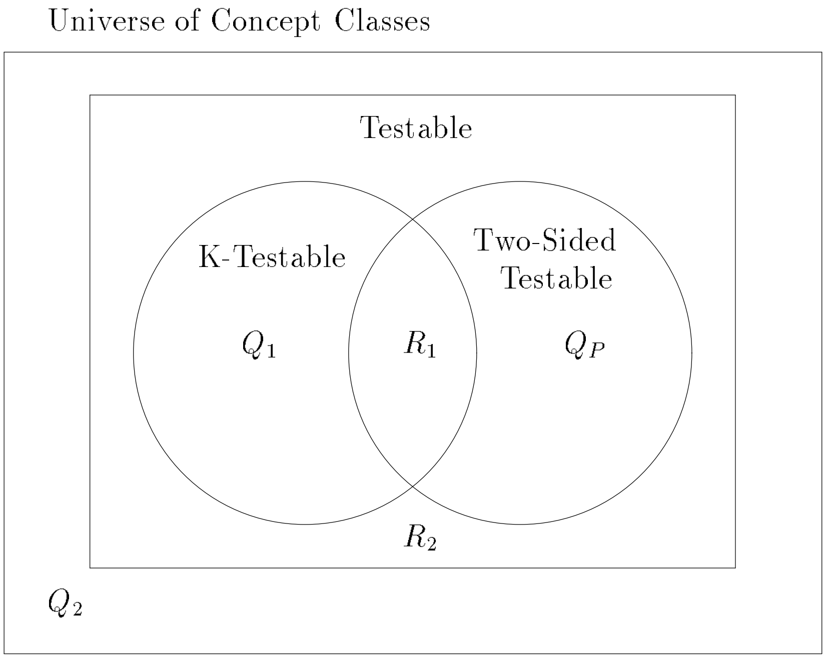

Figure 1: Testing hierarchy on E^2 with respect to a uniform distribution

the figure. We can observe from Figure 1 that for concept classes in E^2 with a uniform distribution, the k-testable and two-sided testable families are incomparable. For a different probability distribution, the hierarchy would look different. For example, if the probability distribution P only assigns a positive probability to a finite set t of points, then any concept class is k-testable and two-sided testable with respect to P. This is because t can be used as a test set for any concept with respect to any concept class. Since the cardinality of t is a constant, the class is k-testable. Since any inconsistent concept will differ from the target concept on t, it will have a positive error.

4 Testability vs. Learnability by Fixed Distributions

One way in which our work on testing differs from the work on learning in [Val84] and [BEHW89] is that they examine learning algorithms that are independent of the underlying probability distribution, while we study testing algorithms that depend on the underlying probability distribution. In other words, a PAC learning algorithm learns a concept regardless of the underlying distribution, whereas an approximate testing algorithm generates a test set for a concept *with respect to a specific distribution.*

Benedek and Itai [BI88] extend the idea of learning to the case where the learning algorithm has knowledge of the underlying distribution on the sample points. We will show, however, that this idea is different from testing with respect to a distribution. They define the following notions, where Q is a concept class defined over a set X, and P is a probability measure on X.

Definition[BI88]. Q is *learnable with respect to* P if there exists a function F such that for all $\epsilon, \delta > 0$, there is a $k > 0$ such that for every $q \in Q$ and $x \in X^k$ selected at random according to P, if F is given x with each point labeled as to whether or not it is in q, then F returns a set $r \subseteq X$ that, with probability $1 - \delta$, has $P(q \triangle r) < \epsilon$.

Thus, given a set of examples drawn according to the distribution P and labeled according to an unknown concept q, F will return a set that, with high probability, is close to q. The function F is a *learning algorithm for* Q *with respect to* P. Note that F may not be polynomially computable, and it does not necessarily return a concept in the class Q.

Definition[BI88]. A set $Q_\epsilon \subseteq 2^X$, where $\epsilon > 0$, is an *ϵ-cover of* Q *with respect to* P if for every $q \in Q$ there is a $q' \in Q_\epsilon$ such that $P(q \triangle q') \leq \epsilon$. Q is *finitely coverable with respect to* P if for every $\epsilon > 0$, there is a finite ϵ-cover Q_ϵ of Q.

Benedek and Itai use the notion of a finite ϵ-cover to characterize when a concept class is learnable with respect to a fixed distribution.

Theorem 4.1 *[BI88] Q is finitely coverable with respect to P if and only if Q is learnable with respect to P.*

Although testability with respect to a fixed distribution appears to be similar to learnability with respect to a fixed distribution, the following two theorems show that the two ideas are actually incomparable.

Theorem 4.2 *There exist a concept class Q and a probability distribution P such that Q is learnable with respect to P, but Q is not testable with respect to P.*

Proof: Let P be the uniform distribution on $[0, 1] \times [0, 1]$, and let class Q_3 consist of concepts composed of either the rectangle $[0, \frac{1}{2}] \times [0, \frac{1}{2}]$ or a finite set of points in the region $[0, 1] \times [0, 1]$. Then Q_3 is finitely coverable with respect to P since the rectangle $[0, \frac{1}{2}] \times [0, \frac{1}{2}]$ and any finite set of points form an ϵ-cover for any $\epsilon > 0$. Therefore, by the above theorem Q_3 is learnable with respect to P. However, Q_3 is not testable with respect to P for $\epsilon < \frac{1}{4}$ since there is no test set for $[0, \frac{1}{2}] \times [0, \frac{1}{2}]$. □

Theorem 4.3 *There exist a concept class Q and a probability distribution P such that Q is testable with respect to P, but Q is not learnable with respect to P.*

Proof: Let $X = [-1, 1]$, let P be the uniform distribution on $[-1, 1]$, and let each concept q in the class Q_4 be composed of a nonempty, finite set of nonempty, disjoint intervals on $[-1, 0]$ and the following additional interval. If q contains n intervals in $[-1, 0]$, then the interval $[\frac{1}{n+1} + \frac{1}{2n(n+1)}, \frac{1}{n}]$ is also in q.

Q_4 is testable with respect to P since each $q \in Q_4$ contains one interval in $(0, 1]$ that tells how many intervals it has in $[-1, 0]$. A test set for $q \in Q_4$ with n intervals in $[-1, 0]$ contains the point $\frac{1}{n}$ and $4n$ other test points that delineate the n intervals in $[-1, 0]$ to within ϵ.

Q_4 is not, however, finitely coverable with respect to P since the part of Q_4 restricted to $[-1, 0]$ is not finitely coverable. Suppose it were coverable and suppose $Q_{\frac{1}{5}}$ were a $\frac{1}{5}$-cover for Q_4. Let t be the set of all endpoints of intervals in $Q_{\frac{1}{5}}$ in the range $[-1, 0]$. Let r be a concept that contains an interval of length $\frac{1}{2}(t_{i+1} - t_i)$ between every two adjacent points $t_i, t_{i+1} \in t$, and contains a similar interval between -1 and the smallest point in t and 0 and the largest point in t, assuming -1 and 0 are not in t. Let concept r also contain the appropriate interval in $(0, 1]$. Concept r is in Q_4, but it differs by at least $\frac{1}{4}$ from every concept in $Q_{\frac{1}{5}}$, so $Q_{\frac{1}{5}}$ is not a $\frac{1}{5}$-cover for Q_4. □

Not only does testability with respect to P not imply learnability with respect to P, but neither does two-sided testability. Concept class Q_4 from the proof of the previous theorem is two-sided testable, but it is not learnable. The notion of k-testability also does not imply learnability with respect to a fixed distribution. Concept class Q_4 can easily be converted into a 2-testable concept class Q'_4 by only allowing intervals with rational endpoints and by changing the part of a concept contained in $(0, 1]$. If $q \in Q'_4$ contains n intervals in $[-1, 0]$, then it contains the point $\frac{1}{n+1}$ and another point that encodes the endpoints of the n intervals. These 2 points form a test set for q for any ϵ since they uniquely identify it.

5 Nonredundant Test Sets and Generic Test Sets

5.1 Nonredundant Test Sets

When looking for a testing algorithm for a class of concepts, it is desirable to find one that produces a minimum number of test points to test a concept. By minimum we mean that no other testing algorithm produces a test set for any concept/error bound pair with fewer test points. Such an algorithm is called *optimal* for the concept class. One property of the test sets produced by an optimal algorithm is that they contain no redundant test points. That is, if any point is removed from the test set, it is no longer a test set. In this section we study testing algorithms that produce nonredundant test sets to see if the size of these test sets can vary. The reason for doing this is to determine whether it is sufficient to require only that a testing algorithm be nonredundant in order to ensure that it produces a "good" (i.e. minimum sized) test set. First we give definitions and an example.

Definition. Let Q be a concept class defined on a set X and let P be a probability measure on X. A test set t for a concept $q \in Q$ and $\epsilon > 0$ is *nonredundant* if it has the property that removing any point from it causes it to no longer be a test set. A testing algorithm T for Q is *nonredundant* if it produces a nonredundant test set for every $q \in Q$ and $\epsilon > 0$. A nonredundant test set is *maximal* if there does not exist a nonredundant test set for q and ϵ of larger cardinality. A test set is *minimal* if there does not exist a test set for q and ϵ of smaller cardinality. A testing algorithm T is *optimal for* Q if it produces a minimal test set for every $q \in Q$ and $\epsilon > 0$.

Thus, a test set t for $q \in Q$ and ϵ is nonredundant if it has the property that for each $x \in t$, there exists a concept $r \in Q$ that is consistent with q on $t - \{x\}$ but has error $P(q \triangle r) > \epsilon$.

We give an example to illustrate the difference between minimal and maximal nonredundant test sets.

Example 5.1 *Class Q_5 consists of three intervals, $Q_5 = \{[0, \frac{1}{3}), [\frac{1}{3}, \frac{2}{3}), [\frac{2}{3}, 1]\}$, where a uniform distribution is defined over $[0, 1]$.*

Proposition 5.1 *For any $q \in Q_5$ and $\epsilon < \frac{2}{3}$, q has a minimal test set of size 1 and a maximal nonredundant test set of size 2.*

Proof: For a minimal test set, choose a positive test point in the target interval. For a maximal nonredundant test set, choose two negative test points, one in each of the other intervals. This test set is maximal because any test set containing more than two points would either contain multiple points in one of the two intervals or a positive test point in the target interval. In either case, one of the three points is redundant. $\square$

When we examine nonredundant testing algorithms for a testable concept class Q, one question to consider is whether there is a maximal size test set produced by these algorithms for any $q \in Q$ and $\epsilon > 0$. It turns out that for many classes there is not.

Theorem 5.1 *There exist testable concept classes that contain concepts for which there exist no maximal nonredundant test sets.*

Proof: Consider the class Q_6 consisting of concepts composed of one or two closed intervals between 0 and 1 on the real line, where a uniform distribution is defined over the interval $[0, 1]$. This class is testable by a result in [Rom92, RS94] (this result is restated in Section 6). Let $q \in Q_6$ be the interval $[0, 1]$ and let $\epsilon = \frac{1}{4}$. There is no maximal nonredundant test set for q and ϵ. That is, given any $m > 15$, there exists a nonredundant test set of size m for q and ϵ. In particular, the test set $t = \{i\delta \mid 1 \leq i \leq m\}$, where $\delta = \frac{3}{4(m-2)}$, is a nonredundant test set of size m for q and ϵ. The distance between two adjacent points in t is δ, and the distance between $m\delta$ and 1 is $\frac{1}{4} - 2\delta$. Any $r \in Q_6$ that is consistent with q on t only differs from q in at most three intervals between test points, thus it has error $P(q \triangle r) \leq \delta + \delta + (\frac{1}{4} - 2\delta) = \frac{1}{4} = \epsilon$. However, if a point is removed from t, then a consistent concept r can differ from q in four intervals and thus have error $P(q \triangle r) = 3\delta + (\frac{1}{4} - 2\delta) = \frac{1}{4} + \delta > \epsilon$. Therefore, t is a nonredundant test set for q and ϵ. □

Notice that the class Q_6 in the previous theorem is not k-testable. However, there are also k-testable concept classes that contain concepts with no maximal nonredundant test sets.

Theorem 5.2 *There exist k-testable concept classes that contain concepts for which there exist no maximal nonredundant test sets.*

Proof: Consider the class R_1 of closed orthogonal rectangles defined in Example 3.2 and consider test sets for $r = [\frac{1}{4}, \frac{3}{4}] \times [\frac{1}{4}, \frac{3}{4}]$ and $\epsilon = \frac{1}{8}$. For a nonredundant test set with at least m points choose the 4 positive test points $(\frac{1}{2}, \frac{1}{4})$, $(\frac{1}{2}, \frac{3}{4})$, $(\frac{1}{4}, \frac{1}{2})$, $(\frac{3}{4}, \frac{1}{2})$, the 4 negative test points $(\frac{1}{2}, \frac{1}{8} + \delta)$, $(\frac{1}{2}, \frac{7}{8} - \delta)$, $(\frac{1}{8} + \delta, \frac{1}{2})$, $(\frac{7}{8} - \delta, \frac{1}{2})$ (where $\delta \ll \epsilon$), and the following additional negative test points. Consider $\lceil \frac{m-10}{2} \rceil$ rectangles that are ϵ larger than r but do not expand past any of the 4 negative test points already chosen. Each of these rectangles should expand in all four directions from r and no two should expand the same distance in any one direction. Also, one rectangle should expand equally in all four directions from r (see Figure 2).

Choose negative test points at the upper left and lower right corners of the intersections of each consecutive pair of these rectangles and the rectangles obtained from expanding r in only two opposite directions (see Figure 2). If k rectangles were chosen in the previous step, then this will add $2(k+1)$ test points, for a total of $2(k+1) + 8 \geq 2(\frac{m-10}{2} + 1) + 8 = m$ test points. None of these test points are redundant, since removing one would allow one of the rectangles to expand in some direction without becoming inconsistent on the remaining points. Since each rectangle is ϵ larger than r, if it expands in any direction it will have error greater than ϵ. Depending on how the rectangles were chosen, additional points may need to be added to obtain a test set. This set of m or more points is a nonredundant test set for r and ϵ. □

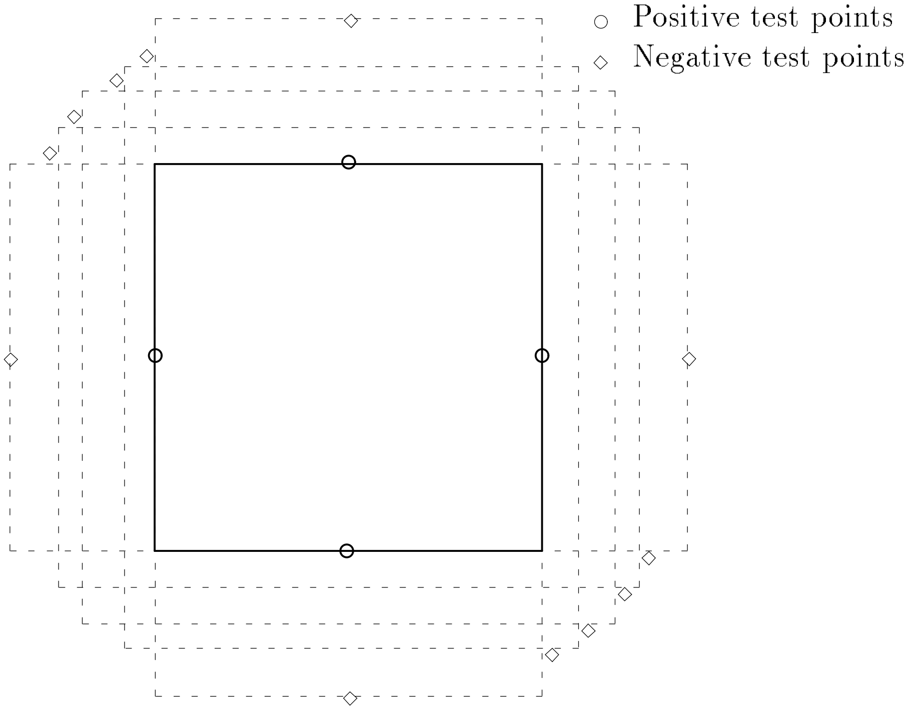

Figure 2: A nonredundant test set for a rectangle

The results of this section demonstrate that it is not sufficient to require only that a testing algorithm be nonredundant in order to ensure that it produces a "good" test set. This is because there are concept classes that have nonredundant testing algorithms that produce test sets with arbitrarily many more test points than an optimal algorithm.

5.2 Generic Test Sets

An approximate testing algorithm, as defined in Section 2, takes as input a concept and an error bound, and it produces a test set that depends upon the concept that was input. In some situations it is desirable to have a testing algorithm that takes as input only an error bound and produces a set of points that is a test set for any concept in the class. Such a test set is called a *generic test set* for the concept class. We now define this notion formally.

Let Q be a concept class defined on a set X, and let P be a probability measure on X. Let $\mathbf{S}$ denote the set of all finite subsets of X and let I denote the open interval of rationals in $(0, 1)$. Let $m: I \to Z^+$ be a positive integer valued function defined on I, and let $\mathbf{T}$ be the set of all computable functions $T: I \to \mathbf{S}$.

Definition. $T \in \mathbf{T}$ is a *generic approximate testing algorithm for* Q *with respect to* P *with* *test set size* $m(\epsilon)$ if for all $\epsilon \in I$, $T(\epsilon) = t, |t| \leq m(\epsilon)$ and for all $q, r \in Q$, if q and r are consistent on t, then $P(q \triangle r) \leq \epsilon$. $T(\epsilon)$ is called a *generic test set for* Q *with error bound* ϵ. If such a T and m exist, then Q is *generically testable with respect to* P *with test set size* $m(\epsilon)$.

The following example demonstrates a generic approximate testing algorithm for a generically testable concept class.

Example 5.2 *Let class* Q_6 *be the concept class defined in the proof of Theorem* 5.1. *Each* $q \in Q_6$ *contains one or two closed intervals in* $[0,1]$. *Given* $\epsilon > 0$, *the set* $t = \{\frac{i\epsilon}{4} \mid 1 \leq i \leq \lfloor \frac{4}{\epsilon} \rfloor\}$ *is a generic test set for* Q_6 *with error bound* ϵ. *Any two concepts* $q, r \in Q_6$ *that are consistent on* t *can only differ in at most four intervals between test points, so the error of one with respect to the other is* $P(q \triangle r) \leq 4\frac{\epsilon}{4} = \epsilon$. *Thus* Q_6 *is generically testable.*

One question to consider is how the notion of generic testability relates to the original notion of approximate testability. It is easy to see that any generic approximate testing algorithm is an approximate testing algorithm. However, the reverse implication does not hold. As the next theorem shows, there are testable concept classes that are not generically testable.

Theorem 5.3 *There exist a concept class* Q *and a probability measure* P *such that* Q *is testable with respect to* P, *but* Q *is not generically testable with respect to* P.

Proof: Let P be the uniform distribution on $[-1,1]$, and let Q_4 be the concept class defined in the proof of Theorem 4.3. That is, every concept $q \in Q_4$ is composed of a nonempty, finite set of nonempty, disjoint intervals on $[-1,0]$ and the following additional interval. If q contains n intervals in $[-1,0]$, then the interval $[\frac{1}{n+1} + \frac{1}{2n(n+1)}, \frac{1}{n}]$ is also in q.

Q_4 is testable with respect to P, as shown in the proof of Theorem 4.3. However, for any $\epsilon < \frac{1}{2}$ and any finite set t of points in $[-1,1]$, there exist concepts $q, r \in Q_4$ that are consistent on t, but for which $P(q \triangle r) > \epsilon$. Let $t_- \subseteq t$ be the points of t that are contained in $[-1,0]$, let $n_{\max}$ be the largest integer such that t contains a point in the interval $[\frac{1}{n_{\max}+1} + \frac{1}{2n_{\max}(n_{\max}+1)}, \frac{1}{n_{\max}}]$, and let $n := \max\{|t_-| + 1, n_{\max} + 1\}$. Let q and r each be composed of n disjoint intervals on $[-1,0]$ and the interval $[\frac{1}{n+1} + \frac{1}{2n(n+1)}, \frac{1}{n}]$. Choose the intervals of q and r so that the interiors of the intervals of q do not intersect those of r, the union of these intervals has probability greater than ϵ, and none of these intervals contains any point in t. Then every point in t will be a negative test point for both q and r, but $P(q \triangle r) > \epsilon$, so t is not a generic test set for Q with error bound ϵ. $\square$

Not only is the notion of generic testability not equivalent to that of testability, but it also differs from the notions of two-sided testability and k-testability. The class Q_4 in the previous proof is an example of a concept class that is two-sided testable but not generically testable, class Q_1 from Example 3.1 of Section 3.1 is an example of a concept class that is k-testable but not generically testable, and class R_2 from Example 3.4 of Section 3.1 is an example of a concept class that is generically testable but not k-testable or two-sided testable.

Class Q_4 from the previous proof is not generically testable and is also not learnable with respect to a uniform distribution on $[-1, 1]$. This is not a coincidence, as the following theorem shows.

Theorem 5.4 *If a concept class Q is generically testable with respect to a probability measure P, then it is learnable with respect to P.*

Proof: Let T be a generic approximate testing algorithm for concept class Q with respect to the probability measure P, and for a given ϵ let $T(\epsilon)$ be a generic test set for Q with error bound ϵ. To obtain an ϵ-cover Q_ϵ of Q choose, for each subset $t \subseteq T(\epsilon)$ that is obtainable by some concept in Q, one concept $q_t \in Q$ that obtains this subset (i.e. $q_t \cap T(\epsilon) = t$). Q_ϵ is a finite ϵ-cover for Q because any $q \in Q$ is consistent with some $q_i \in Q_\epsilon$ on $T(\epsilon)$, and thus $P(q \triangle q_i) \leq \epsilon$ since $T(\epsilon)$ is a generic test set. Since the choice of ϵ was arbitrary, Q is finitely coverable with respect to P, and by Theorem 4.1 it is thus learnable with respect to P. □

It is easy to see that the implication in the previous theorem cannot be reversed since Theorem 4.2 showed that learnability with respect to a fixed probability distribution does not imply testability, but generic testability does imply testability.

6 Testability and Dimension

In this section we examine general properties of a concept class that determine whether or not it is testable. Both properties of a concept class Q and properties of the probability measure P used to measure the error between two concepts can cause a class Q to be testable with respect to P. We now give a result that relates certain probability distributions to testability.

Proposition 6.1 *If a discrete probability distribution P is defined on the domain X, then any concept class is generically testable with respect to P.*

Proof: Let $Q \subseteq 2^X$ be a concept class defined on X and let P be a discrete probability distribution defined on X. There is a countable subset $X' = \{x_1, x_2, \dots\}$ of X such that $\sum_{i=1}^{\infty} P(x_i) = 1$ and for all $x \in X - X'$, $P(x) = 0$. Given any $\epsilon > 0$ there exists an n such that $\sum_{i=1}^{n} P(x_i) \geq 1 - \epsilon$, and thus the set of points $t = \{x_1, x_2, \dots, x_n\}$ is a generic test set for Q with error bound ϵ. □

Although many of our results are applicable to all domains and probability distributions, because of the above result they often become trivial for discrete domains and probability distributions. Even if the probability distribution P is not discrete, if it has the property that $P(p \triangle q) = 0$ for all $p, q \in Q$, then Q is generically testable with respect to P.

We now examine properties of a concept class that cause it to be testable. It is easy to see that any class of n concepts is testable using at most $n-1$ test points for any concept. In [GK95] it was shown that there exist classes of n concepts that *require* $n-1$ test points to test certain concepts. It is also simple to see that any concept class of pairwise disjoint, nonempty concepts is 1-testable for any probability distribution, since any point inside a concept forms a test set for it for any given error bound.

In computational learning theory, a complexity measure known as the Vapnik-Chervonenkis dimension of a concept class characterizes when a concept class is PAC learnable. This complexity measure is also useful for characterizing when a concept class is testable, so we define it now.

Definition[VC71, BEHW89]. Given a concept class Q defined over X and a finite $S \subseteq X$, $\Pi_Q(S)$ denotes the set of all subsets of S that can be obtained by intersecting S with a concept in Q, i.e. $\Pi_Q(S) := \{S \cap q \mid q \in Q\}$. If $\Pi_Q(S) = 2^S$, then S is *shattered* by Q. The *Vapnik-Chervonenkis dimension* of Q (or simply the *VC-dimension* of Q) is the largest integer k such that there exists a subset S of X of cardinality k that is shattered by Q. If no such k exists, the VC-dimension of Q is infinite.

As an illustration of this dimension, the class R of closed orthogonal rectangles, defined in Section 2, has VC-dimension 4. The set $S = \{(-1,0), (1,0), (0,-1), (0,1)\}$ can be shattered by R, but no set of 5 points can be shattered by R. In Theorem 2.1 of [BEHW89] it was proven that a concept class is PAC learnable if and only if it has finite VC-dimension.

VC-dimension can also be used to partially characterize when concept classes are testable, as the following theorem demonstrates. This theorem was proven in [Rom92, RS94]. The notion of well-behaved classes mentioned in the theorem is a measure-theoretic condition on concept classes given in [BEHW89]. Virtually any concept class considered in the context of learning or testing will be well-behaved.

Theorem 6.1 *[Rom92] Let Q be a concept class defined over X, and let P be a probability measure on X. If Q is well-behaved and has finite VC-dimension, then Q is testable with respect to P.*

By this theorem we can conclude that the concept classes R_1 and R_2 defined in Examples 3.2 and 3.4 of Section 3.1 are testable. Note, however, that this theorem does not show how to deterministically choose test points. Also, VC-dimension does not completely characterize when a concept class is testable. Classes Q_1 and Q_P, from Examples 3.1 and 3.3 of Section 3.1, are both testable but they have infinite VC-dimension (it was shown in [HW87] that the class Q_P of convex polygons has infinite VC-dimension).

Goldman and Kearns [GK95] relate the VC-dimension of a concept class to the number of examples that a helpful teacher must present in order for all students to learn a concept. As we stated in the introduction, this notion is analogous to the size of a test set needed to test a concept. They show that there are some classes that require a test set much larger than the VC-dimension of the class to test them, and there are other classes that can be tested with a number of test points smaller than the VC-dimension of the class. They also give an upper bound on the number of test points needed to test a class that is a function of the VC-dimension of the class and the cardinality of the class.

It was shown in [RS94] that concept classes with the same VC-dimension can even have different levels of testability. This was shown by proving that the class consisting of concepts composed of exactly two closed intervals on the real line is k-testable, while the class consisting of concepts composed of one or two closed intervals on the real line is not k-testable. Both of these concept classes have VC-dimension 4.

Since any class with finite VC-dimension is testable, we now consider classes with infinite VC-dimension to determine properties that make them testable or untestable. First we examine a few examples that illustrate how a small change in the definition of a concept class can have a large effect on the testability of the class. Consider the following three concept classes defined on $[0, 1]$, where $q_1 := [0, \frac{1}{2}]$, $q_2 := [\frac{1}{2}, 1]$, and a uniform distribution is defined on $[0, 1]$.

Example 6.1 *Class* $Q_7 := \{q_1, q_2\}$.

Class Q_7 is obviously 1-testable, since it only contains two concepts. Now consider the following concept class containing concepts that are similar to those in Q_7:

Example 6.2 *Class* $Q_8 := \{q_1 \triangle p \mid p$ *is any finite set of points in* $[0, 1]\} \cup \{q_2 \triangle p \mid p$ *is any finite set of points in* $[0, 1]\}$.

Although class Q_8 only contains two types of concepts – those that differ from q_1 by a finite set of points, and those that differ from q_2 by a finite set of points – the addition of these finite point sets makes this class untestable, as the following proposition demonstrates.

Proposition 6.2 *Q_8 is untestable.*

Proof: Assume that T is a testing algorithm for Q_8, and let $T(q_1, \frac{1}{2})$ be the finite test set generated by T for q_1 and error bound $\frac{1}{2}$. A concept $r \in Q_8$ that is consistent with q_1 on $T(q_1, \frac{1}{2})$ but has an error of 1 can be constructed as follows. Let $r = q_2 \triangle p'$, where p' consists of all positive test points generated by T in $[0, \frac{1}{2})$ and all negative test points in $[\frac{1}{2}, 1]$. $\square$

By making a small change to class Q_8, we can again make it testable.

Example 6.3 *Class $Q_9 := \{q_1 \triangle p \mid p$ is any finite set of points in $[0,1]$ containing the point $\frac{3}{4}\} \cup \{q_2 \triangle p \mid p$ is any finite set of points in $[0,1]$ containing the point $\frac{3}{4}\}$.*

Class Q_9 is 1-testable using the point $\frac{3}{4}$. These examples illustrate that a finite set of points with probability zero can greatly effect the testability of a concept class. Also, since both class Q_8 and Q_9 have infinite VC-dimension, Theorem 6.1 cannot be used to distinguish between them.

We wish to find properties that can distinguish between classes such as Q_8 and Q_9. One such property is the testing dimension of a concept class, which is similar to the VC-dimension. We define this complexity measure now.

Definition. Given a concept class Q defined over X, the *testing dimension* of Q is the largest integer k such that *all* subsets $S \subset X$ of cardinality k are shattered by Q. If no such k exists, the testing dimension of Q is infinite.

As an illustration of this dimension, the class Q_P of convex polygons, which has infinite VC-dimension, has testing dimension two since three collinear points cannot be shattered by the class.

The testing dimension of a concept class never exceeds the VC-dimension of the class. This dimension was independently defined by Sontag [Son92] and Abu-Mostafa and St. Jacques [AMJ85] to characterize the complexity of classes of neural nets.

The following two results relate testing dimension to testability. For these results we define two subclasses of a concept class that are determined by a target concept q and an error bound ϵ.

Definition.

$$Q_{>\epsilon}(q) := \{r \in Q \mid P(q\triangle r) > \epsilon\}$$
$$Q_{<\epsilon}(q) := \{r \in Q \mid P(q\triangle r) < \epsilon\}$$

Proposition 6.3 *If Q is a concept class and there exist $q \in Q, \epsilon > 0$ such that $Q_{>\epsilon}(q)$ has infinite testing dimension, then Q is untestable.*

Proof: Suppose Q is testable, and let $t = T(q, \epsilon)$ be a test set for q and ϵ. Since $Q_{>\epsilon}(q)$ has infinite testing dimension, there exists $r \in Q_{>\epsilon}(q)$ such that $r \cap t = q \cap t$. Therefore r is consistent with q on t, but $P(q\triangle r) > \epsilon$, which is a contradiction. $\square$

Proposition 6.3 applies to class Q_2 in Section 3.1 and class Q_8 above and shows that they are untestable. Class Q_9 above also has infinite testing dimension, but there does not exist $q \in Q_9$ such that $Q_{9>\epsilon}(q)$ has infinite testing dimension, so the proposition cannot be applied to it.

Proposition 6.4 *If Q is a concept class and there exist $p, q \in Q$ and $\epsilon > 0$ such that $P(p\triangle q) > 2\epsilon$ and $Q_{<\epsilon}(q)$ has infinite testing dimension, then Q is untestable.*

Proof: If $p, q \in Q$ are such that $P(p\triangle q) > 2\epsilon$, then for all $r \in Q$, either $P(r\triangle q) > \epsilon$ or $P(r\triangle p) > \epsilon$ or both. This is because if there were an $r \in Q$ that was within ϵ of both p and q, then p and q would differ by no more than 2ϵ. In particular, for all $r \in Q_{<\epsilon}(q)$, $P(r\triangle p) > \epsilon$. Therefore, since $Q_{<\epsilon}(q)$ has infinite testing dimension, $Q_{>\epsilon}(q)$ has infinite testing dimension, and by Proposition 6.3 Q is untestable. $\square$

The following theorem applies to concept classes defined over uncountable domains (such as E^d) where the probability distribution P is continuous. That is, P has the property that for all $x \in X$, $P(x) = 0$.

Theorem 6.2 *If a concept class Q has infinite testing dimension, is closed under arbitrary intersections, and contains a concept $q \in Q$ with nonzero probability, then it is untestable.*

Proof: Let Q be a concept class with the above properties and let $q \in Q$ be such that $P(q) = \delta > 0$. Assume that Q is testable, and let $t = T(q, \frac{\delta}{2})$ be a test set for q with error bound $\frac{\delta}{2}$. Let t^+ be the set of positive test points in t, define $Q_{t^+} := \{r \in Q \mid r \cap t = t^+\}$, and let q^+ be the intersection of all concepts in Q_{t^+}. Since Q is closed under arbitrary intersections, $q^+ \in Q$, and $q^+ \cap t = t^+$. Since Q has infinite testing dimension, for any $x \notin t$ there exists $r \in Q$ such that $r \cap (t \cup \{x\}) = t^+$. Since $r \in Q_{t^+}$ and $x \notin r$, x cannot be in q^+. Therefore, $q^+ = t^+$, which means that $P(q^+) = 0$ and $P(q^+ \triangle q) = \delta$. Since q^+ is consistent with q on t, this contradicts the fact that t is a test set for q with error bound $\frac{\delta}{2}$. $\square$

Note that any two of the three properties in Theorem 6.2 are not sufficient to show untestability for a concept class. Also, if we weaken the second property to "closed under finite intersections" then the result no longer holds, as the following example illustrates.

Example 6.4 *Let* $X = [0,1]$ *and define class* $Q_{10} := \{[0, \frac{1}{2}] \triangle p \mid p$ *is any finite set of points in* $[0,1]$ *that does not contain the point* $\frac{3}{8}\} \cup \{[0, \frac{1}{4}] \triangle p \mid p$ *is any finite set of points in* $[0,1]$ *that does not contain the point* $\frac{3}{8}\}$.

Class Q_{10} has infinite testing dimension and is closed under finite intersections, but it is testable with the point $\frac{3}{8}$, since any concept that contains this point differs from $[0, \frac{1}{2}]$ by a finite set of points, and any concept that does not contain this point differs from $[0, \frac{1}{4}]$ by a finite set of points.

The results of this section show that finite VC-dimension implies testability, while infinite testing dimension combined with certain other properties implies untestability. There are no results, however, for concept classes with infinite VC-dimension and finite testing dimension. Therefore, a complete characterization of which concept classes are testable remains an open problem.

7 Conclusion and Open Problems

In this paper we have defined what it means for a class of concepts to be approximately testable and have identified several levels of testability that define a hierarchy. We have given examples of concept classes for each level and have shown the hierarchy defined by them. We have explored the relationship between testing and learning and have compared our testing model to several learning models. We have discussed the ideas of nonredundant test sets and generic test sets. In an attempt to characterize testable concept classes, we have defined a new notion called testing dimension and have examined its relationship to approximate testing.

Since this is a new area of research, there are many open problems and many directions to be taken. We only mention a few open problems here. One major open problem is the exact characterization of when a concept class is approximately testable. Also, there is the task of identifying properties that distinguish k-testable concept classes from ones that are testable but require larger than a constant size test set, and of identifying properties that distinguish two-sided testable concept classes from ones that are not two-sided testable. There are also more relationships between learning and testing to be explored. Finally, there is the application of testability ideas to non-geometric concept classes.

8 Acknowledgements

The author would like to thank Dick Lipton for suggesting the idea of maximal nonredundant test sets, and Lenny Pitt and Sally Goldman for their helpful comments on an earlier paper on testing. Also, thanks to Paul Green for several discussions on topology.

References

[ABCST92] Martin Anthony, Graham Brightwell, Dave Cohen, and John Shawe-Taylor. Specification By Examples. In David Haussler, editor, *Proceedings of the Fifth ACM Workshop on Computational Learning Theory*, Pittsburgh, PA, July 27-29 1992. ACM Press.

[AMJ85] Yaser S. Abu-Mostafa and Jeannine-Marie St. Jacques. Information Capacity of the Hopfield Model. *IEEE Transactions on Information Theory*, IT-31(4):461–464, July 1985.

[BEHW89] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred Warmuth. Learnability and the Vapnik-Chervonenkis Dimension. *Journal of the Associaton for Computing Machinery*, 36(4):929–965, 1989.

[BI88] Gyora M. Benedek and Alon Itai. Learnability by Fixed Distributions. In David Haussler and Leonard Pitt, editors, *Proceedings of the 1988 Workshop on Computational Learning Theory*, pages 80–90, MIT, August 3-5 1988. Morgan Kaufmann Publishers.

[CS87] John C. Cherniavsky and Carl H. Smith. A Recursion Theoretic Approach to Program Testing. *IEEE Transactions on Software Engineering*, SE-13(7):777–784, July 1987.

[GK95] Sally A. Goldman and Michael J. Kearns. On the Complexity of Teaching. *Journal of Computer and System Sciences*, 50(1):20–31, February 1995.

[HW87] David Haussler and Emo Welzl. ε-Nets and Simplex Range Queries. *Discrete Computational Geometry*, 2:127–151, 1987.

[JT92] Jeffrey Jackson and Andrew Tomkins. A Computational Model of Teaching. In David Haussler, editor, *Proceedings of the Fifth ACM Workshop on Computational Learning Theory*, Pittsburgh, PA, July 27-29 1992. ACM Press.

[Rom92] Kathleen Romanik. *Approximate Testing Theory*. PhD thesis, University of Maryland at College Park, College Park, MD, August 1992. Technical Report CS-TR-2947/UMIACS-TR-92-89.

[RS94] Kathleen Romanik and Carl Smith. Testing Geometric Objects. *Computational Geometry: Theory and Applications*, 4:157–176, 1994.

[SDHK91] Steven Salzberg, Arthur Delcher, David Heath, and Simon Kasif. Learning with a Helpful Teacher. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence*, pages 705–711, Sydney, Australia, August 1991. Morgan Kaufmann Publishers.

[SM91] Ayumi Shinohara and Satoru Miyano. Teachability in Computational Learning. In *New Generation Computing*, volume 8, pages 337–347. Ohmsha, Ltd. and Springer-Verlag, 1991.

[Son92] Eduardo D. Sontag. FeedForward Nets for Interpolation and Classification. *Journal of Computer and System Sciences*, 45(1):20–48, August 1992.

[Val84] Leslie G. Valiant. A Theory of the Learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[VC71] V.N. Vapnik and A.Ya. Chervonenkis. On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities. *Theory Probability and its Applications*, 16(2):264–280, 1971.