The Cost Complexity of Interactive Learning

Steve Hanneke
Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA 15213 USA
shanneke@cs.cmu.edu

June 2006

Abstract

In this paper, I describe a general framework in which a learning algorithm is tasked with learning some concept from a known class by interacting with a teacher via questions. Each question has an arbitrary known cost associated with it, which the learner is required to pay in order to have the question answered. Exploring the information-theoretic limits of this framework, I define a notion called the *cost complexity* of learning, analogous to traditional notions of sample complexity. I discuss this topic for the Exact Learning setting as well as PAC Learning with a pool of unlabeled examples. In the former case, the learner is allowed to ask *any* question, while in the latter case, all questions must concern the target concept's behavior on a set of unlabeled examples. In both settings, I derive upper and lower bounds on the cost complexity of learning, based on a combinatorial quantity I call the *General Identification Cost*.

1 Introduction

The ability to ask questions to a knowledgeable teacher can make learning easier. This fact is no secret to any elementary school student. But how much easier? Some questions are more difficult for the teacher to answer than others. How much inconvenience must even the most conscientious learner cause to a teacher in order to learn a concept? This paper explores these and related questions about the fundamental advantages and limitations of learning by interaction.

In machine learning research, it is becoming increasingly apparent that well-designed interactive learning algorithms can provide valuable improvements in learning performance while reducing the amount of effort required of a human annotator. This research has mainly focused on two formal settings of learning: Exact Learning by queries and pool-based Active PAC Learning. Informally, the objective in the setting of Exact Learning by queries is to perfectly identify a target concept (classifier) by asking questions. In contrast, the pool-based Active PAC setting is concerned only with approximating the concept with high probability with respect to an unknown distribution on the set of possible instances. In this latter setting, the learning algorithm is restricted to asking only questions that relate to the concept's behavior on a particular set of unannotated instances drawn independently from the unknown distribution.

In this paper, I study both of these active learning settings under a broad definition. Specifically, I consider a learning protocol in which the learner can ask *any* question, but each possible question has an associated *cost*. For example, a query of the form "what is the label of example x" might cost \$1, while a query of the form "show me a positive example" might cost \$10. The objective is to learn the concept while minimizing the total *cost* of queries made. One would like to know how much cost even the most clever learner might be required to pay to learn a concept from a particular concept space in the worst case. This can be viewed as a generalization of notions of *sample complexity* or *query complexity* found in the learning theory literature. I refer to this best worst case cost as the *cost complexity* of learning. This quantity is defined without reference to computational feasibility, focusing instead on the information-theoretic boundaries of this setting (in the limit of unbounded computation). Below, I derive bounds on the cost complexity of learning, as a function of the concept space and cost function, for both Exact Learning from queries and pool-based Active PAC Learning.

Section 2 formally introduces the setting of Exact Learning from queries, describes some related work, and defines cost complexity for that setting. It also serves to introduce the notation and fundamental definitions used throughout this paper. The section closely parallels the work of Balcázar et al. [1]. The primary contribution of Section 2 is a derivation of upper and lower bounds on the cost complexity of Exact Learning from queries. This is followed, in Section 3, by a formal definition of pool-base Active PAC Learning and extension of the notion of cost complexity to that setting. The primary contributions of Section 3 include a derivation of upper and lower bounds on the cost complexity of learning in that general setting, as well as an interesting corollary for intersection-closed concept spaces. I know of no previous work giving general results of this type.

2 Active Exact Learning

In this setting, there is an *instance space* $\mathcal{X}$ and *concept space* $\mathcal{C}$ on $\mathcal{X}$ such that any $h \in \mathcal{C}$ is a distinct function $h : \mathcal{X} \to \{0, 1\}$.[1] Additionally, define $\mathcal{C}^* = \{h : \mathcal{X} \to \{0, 1\}\}$. That is, $\mathcal{C}^*$ is the *most general* concept space, containing all possible labelings of $\mathcal{X}$. In particular, any concept space $\mathcal{C}$ is a subset of $\mathcal{C}^*$. For a particular learning problem, there is an unknown *target concept* $f \in \mathcal{C}$, and the task is to identify f using a teacher's answers to queries made by the learning algorithm. Formally, an *actual query* is any function in $\tilde{Q} = \{\tilde{q} : \mathcal{C}^* \to 2^{\mathcal{A}^*} \setminus \{\varnothing\}\}$,[2] for some *answer set* $\mathcal{A}^*$. By a learning algorithm "making an actual query", I mean that it selects a function $\tilde{q} \in \tilde{Q}$, passes it to the teacher, and the teacher returns a single *answer* $\tilde{a} \in \tilde{q}(f)$ where f is the target concept. A concept $h \in \mathcal{C}^*$ is *consistent* with an answer $\tilde{a}$ to an actual query $\tilde{q}$ if $\tilde{a} \in \tilde{q}(h)$. Thus, I assume the teacher always returns an answer that the target concept is consistent with; however, when there are multiple such answers, the teacher may arbitrarily select from amongst them.

Traditionally, the subject of active learning has been studied with respect to specific restricted query types, such as membership queries, and the learning algorithm's objective has been to minimize the *number* of queries used to learn. However, it is often the case that learning with these simple types of queries is difficult, but if the learning algorithm is allowed just a few *special* queries, learning becomes significantly easier. The reason we are initially reluctant to allow the learner to ask certain types of queries is that these queries are difficult, expensive, or sometimes impossible to answer. However, we can incorporate this difficulty level into the framework by assigning each query type a specific *cost*, and then allowing the learning algorithm to explicitly optimize the *cost* needed to learn, rather than the *number* of queries. In addition to allowing the algorithm to trade off between different types of queries, this also gives us the added flexibility to specify different costs within the same family (e.g., perhaps some membership queries are more expensive than others).

Formally, in this framework there is a *cost function*. Let $\alpha > 0$ be a constant. A cost function is any $c : \tilde{Q} \to (\alpha, \infty]$. In practice, c would typically be defined by the user responsible for answering the queries, and could be based on the time, resources, or operating expenses necessary to obtain the answer. Note that if a particular type of query is unanswerable for a particular application, or if the user wishes to work with a reduced set of possible queries, one can always define the costs of those undesirable query types to be ∞, so that any reasonable learning algorithm ignores them if possible.

While the notion of *actual query* closely corresponds to the actual mechanism of querying in practice, it will be more convenient to work with the information-theoretic implications of these queries. Define the set of *effective queries* $\mathcal{Q} = \{q : \mathcal{C}^* \to 2^{2^{\mathcal{C}^*}} \setminus \{\varnothing\} | \forall f \in \mathcal{C}^*, a \in q(f) \Rightarrow [f \in a \wedge \forall h \in a, a \in q(h)]\}$. Each effective query corresponds to an equivalence class of actual queries, defined by mapping any answer to the set of concepts consistent with it. We can thus define the mapping

[1] All of the main results easily generalize to multiclass as well.

[2] The restriction that $\tilde{q}(f) \neq \{\}$ is a bit like an assumption that every valid question has at least one answer for any target concept. However, we can always define some particular answer to mean "there is no answer," so this restriction is really more of a notational convenience than an assumption.

$$\mathcal{E}(q) = \{\tilde{q} | \tilde{q} \in \tilde{Q}, \forall f \in \mathcal{C}^*, [\exists \tilde{a} \in \tilde{q}(f) \text{ with } a = \{h | h \in \mathcal{C}^*, \tilde{a} \in \tilde{q}(h)\}] \Leftrightarrow a \in q(f)\}.$$

By an algorithm "making an effective query q," I mean that it makes an actual query in $\mathcal{E}(q)$,[3] (a good algorithm will pick a cheaper actual query). For the purpose of this best-worst-case analysis, the following definition is appropriate. For a cost function c, define a corresponding *effective cost function* (overloading notation) $c : \mathcal{Q} \rightarrow [\alpha, \infty]$, such that $\forall q \in \mathcal{Q}, c(q) = \inf_{\tilde{q} \in \mathcal{E}(q)} c(\tilde{q})$. The following definitions illustrate how query types can be defined using effective queries.

A *positive example query* is any $\tilde{q} \in \mathcal{E}(q_S)$ for some $S \subseteq \mathcal{X}$, such that $q_S \in \mathcal{Q}$ is defined by $\forall f \in \mathcal{C}^*$ s.t. $[\exists x \in S : f(x) = 1], q_S(f) = \{\{h | h \in \mathcal{C}^*, h(x) = 1\} | x \in S : f(x) = 1\}$, and $\forall f \in \mathcal{C}^*$ s.t. $[\forall x \in S, f(x) = 0], q_S(f) = \{\{h | h \in \mathcal{C}^* : \forall x \in S, h(x) = 0\}\}$.

A *membership query* is any $\tilde{q} \in \mathcal{E}(q_{\{x\}})$ for some $x \in \mathcal{X}$. This special case of a positive example query can equivalently be defined by $\forall f \in \mathcal{C}^*, q_{\{x\}}(f) = \{\{h | h \in \mathcal{C}^*, h(x) = f(x)\}\}$.

These effectively correspond to asking for any example labeled 1 in S or an indication that there are none (positive example query), and asking for the label of a particular example in $\mathcal{X}$ (membership query). I will refer to these two query types in subsequent examples, but the reader should keep in mind that the theorems below apply to *all* types of queries.

Additionally, it will be useful to have a notion of an *effective oracle*, which is an unknown function defining how the teacher will answer the various queries. Formally, an effective oracle T is any function in $\mathcal{T} = \{T : \mathcal{Q} \rightarrow 2^{\mathcal{C}^*} | \forall q \in \mathcal{Q}, T(q) \in \cup_{f \in \mathcal{C}^*} q(f)\}$.[4] For convenience, I also overload this notation, defining for a set of queries $R \subseteq \mathcal{Q}$, $T(R) = \cap_{q \in R} T(q)$.

Definition 2.1. *A* learning algorithm $\mathcal{A}$ *for $\mathcal{C}$ using cost function c is any algorithm which, for any (unknown) target concept $f \in \mathcal{C}$, by a finite number of finite cost actual queries, is guaranteed to reduce the set of concepts in $\mathcal{C}$ consistent with the answers to precisely $\{f\}$. A concept space $\mathcal{C}$ is* learnable *with cost function c using total cost t if there exists a learning algorithm for $\mathcal{C}$ using c guaranteed to have the sum of costs of the queries it makes at most t.*[5]

Definition 2.2. *For any instance space $\mathcal{X}$, concept space $\mathcal{C}$ on $\mathcal{X}$, and cost function c, define the* cost complexity*, denoted CostComplexity$(\mathcal{C}, c)$, as the infimum $t \geq 0$ such that $\mathcal{C}$ is learnable with cost function c using total cost no greater than t.*

Equivalently, we can define cost complexity using the following recurrence. If $|\mathcal{C}| = 1$, *CostComplexity*$(\mathcal{C}, c) = 0$. Otherwise,

$$\textit{CostComplexity}(\mathcal{C}, c) = \inf_{\tilde{q} \in \tilde{Q}} c(\tilde{q}) + \max_{f \in \mathcal{C}, \tilde{a} \in \tilde{q}(f)} \textit{CostComplexity}(\{h | h \in \mathcal{C}, \tilde{a} \in \tilde{q}(h)\}, c)$$

Since

$$\begin{aligned} &\inf_{\tilde{q} \in \tilde{Q}} c(\tilde{q}) + \max_{f \in \mathcal{C}, \tilde{a} \in \tilde{q}(f)} \textit{CostComplexity}(\{h | h \in \mathcal{C}, \tilde{a} \in \tilde{q}(h)\}, c) \\ &\quad = \inf_{q \in \mathcal{Q}} \inf_{\tilde{q} \in \mathcal{E}(q)} c(\tilde{q}) + \max_{f \in \mathcal{C}, \tilde{a} \in \tilde{q}(f)} \textit{CostComplexity}(\mathcal{C} \cap \{h | h \in \mathcal{C}^*, \tilde{a} \in \tilde{q}(h)\}, c) \\ &\quad\quad = \inf_{q \in \mathcal{Q}} c(q) + \max_{f \in \mathcal{C}, a \in q(f)} \textit{CostComplexity}(\mathcal{C} \cap a, c), \end{aligned}$$

we can equivalently define cost complexity in terms of *effective queries* and *effective cost*. That is, *CostComplexity*$(\mathcal{C}, c)$ is the infimum $t \geq 0$ such that there is an algorithm guaranteed to identify any $f \in \mathcal{C}$ using *effective* queries with total of *effective* costs no greater than t.

[3] I assume $\mathcal{A}^*$ is sufficiently expressive so that $\forall q \in \mathcal{Q}, \mathcal{E}(q) \neq \varnothing$; alternatively, we could define $\mathcal{E}(q) = \varnothing \Rightarrow c(q) = \infty$ without sacrificing the main theorems. Additionally, I will assume that it is possible to find an actual query in $\mathcal{E}(q)$ with cost arbitrarily close to $\inf_{\tilde{q} \in \mathcal{E}(q)} c(\tilde{q})$ for any $q \in \mathcal{Q}$ using finite computation.

[4] An effective oracle corresponds to a deterministic stateless teacher, which gives up as little information as possible. It is also possible to analyze a setting in which asking two queries from the same equivalence class, or asking the same question twice, can possibly lead to two different answers. However, the worst case in both settings is identical, so the worst case results obtained for this setting also apply to the more general case.

[5] I have made the dependence of $\mathcal{A}$ on the teacher implicit. To be formally correct, $\mathcal{A}$ should have the teacher's effective oracle T as input, and is guaranteed to output f for any $T \in \mathcal{T}$ s.t. $\forall q \in \mathcal{Q}, T(q) \in q(f)$. Cost is then a book-keeping device recording how $\mathcal{A}$ uses T during execution.

2.1 Related Work

There have been a relatively large number of contributions to the study of Exact Learning from queries. In particular, much interest has been given to settings in which the learning algorithm is restricted to a few specific types of queries (e.g. membership queries and equivalence queries). However, these contributions focus entirely on the *number* of queries needed, rather than *cost*. The most relevant work in this area is by Balcázar, Castro, and Guijarro [1]. Prior to publication of [2], there were a variety of publications in which the learning algorithm could use some specific set of queries, and which derived bounds on the number of queries any algorithm might be required to make in the worst case in order to learn. For example, [3] analyzed the combination of membership and proper equivalence queries, [4] additionally analyzed learning from membership queries alone, while [5] considered learning from just proper equivalence queries. Amidst these various special case analyses, somewhat surprisingly, Balcázar et al. [2] discovered that the query complexity bounds derived in these works were all special cases of a single general theorem, applying to the broad class of *sample-based queries*. They further generalized this result in [1], giving results that apply to any combination of *any* query types. That work defines an abstract combinatorial quantity, which they call the *General Dimension*, which provides a lower bound on the query complexity, and is within a log factor of it. Furthermore, the General Dimension can actually be computed for a variety of interesting combinations of query types. Until now there has not been any analysis I know of that considers learning with *all* query types, but giving each query a cost, and bounding the worst-case *cost* that a learning algorithm might be required to incur. In particular, the analysis of the next subsection can be viewed as a generalization of [1] to add this notion of cost, such that [1] represents the special case of cost that is uniformly 1 on a particular set of queries and ∞ on all other queries.

2.2 Cost Complexity Bounds

I now turn to the subject of exploring the fundamental limits of interactive learning in terms of cost. This discussion closely parallels that of Balcázar, Castro, and Guijarro [1].

Definition 2.3. *For any instance space $\mathcal{X}$, concept space $\mathcal{C}$ on $\mathcal{X}$, and cost function c, define the* General Identification Cost*, denoted $GIC(\mathcal{C}, c)$, as follows.*

$$GIC(\mathcal{C}, c) = \inf\{t | t \geq 0, \forall T \in \mathcal{T}, \exists R \subseteq \mathcal{Q}, \textit{s.t.}[\textstyle\sum_{q \in R} c(q) \leq t] \wedge [|\mathcal{C} \cap T(R)| \leq 1]\}$$

We can also express this as $GIC(\mathcal{C}, c) = \sup_{T \in \mathcal{T}} \inf_{R \subseteq \mathcal{Q} : |\mathcal{C} \cap T(R)| \leq 1} \sum_{q \in R} c(q)$. Note that calculating this corresponds to a much simpler optimization problem than calculating the cost complexity. The General Identification Cost is a direct generalization of the General Dimension of [1], which itself generalizes quantities such as Extended Teaching Dimension [4], Strong Consistency Dimension [5], and the Certificate Sizes of [3]. It can be interpreted as a sort of game. This game is similar to the usual setting, except that the teacher's answers are not restricted to be consistent with a concept. Imagine there is a helpful spy who knows precisely how the teacher will respond to every query. The spy is able to suggest queries to the learner, and wishes to cause the learner to pay as little as possible. If the spy is sufficiently clever at suggesting queries, and the learner follows every suggestion by the spy, then after asking some minimal cost set of queries the learner can narrow the set of concepts in $\mathcal{C}$ consistent with the answers down to at most one. The General Identification Cost is precisely the worst case limiting cost the learner might be forced to pay during this process, no matter how clever the spy is at suggesting queries.

Lemma 2.1. *For any instance space $\mathcal{X}$, concept space $\mathcal{C}$ on $\mathcal{X}$, and cost function c, if $V \subseteq \mathcal{C}$, then $GIC(V, c) \leq GIC(\mathcal{C}, c)$.*

Proof. It clearly holds if $GIC(\mathcal{C}, c) = \infty$. If $GIC(\mathcal{C}, c) < k$, then $\forall T \in \mathcal{T}$, $\exists R \subseteq \mathcal{Q}$ s.t. $\sum_{q \in R} c(q) < k$ and $1 \geq |\mathcal{C} \cap T(R)| \geq |V \cap T(R)|$, and therefore $GIC(V, c) < k$. The limit as $k \rightarrow GIC(\mathcal{C}, c)$ gives the result. □

Lemma 2.2. *For any $\gamma > 0$, instance space $\mathcal{X}$, finite concept space $\mathcal{C}$ on $\mathcal{X}$ with $|\mathcal{C}| > 1$, and cost function c such that $GIC(\mathcal{C}, c) < \infty$, $\exists q \in \mathcal{Q}$ such that $\forall T \in \mathcal{T}$,*

$$|\mathcal{C} \setminus T(q)| \geq c(q) \frac{|\mathcal{C}| - 1}{GIC(\mathcal{C}, c) + \gamma}.$$

That is, regardless of which answer the teacher picks, there are at least $c(q)\frac{|\mathcal{C}|-1}{GIC(\mathcal{C},c)+\gamma}$ *concepts in* $\mathcal{C}$ *inconsistent with the answer.*

Proof. Suppose $\forall q \in \mathcal{Q}, \exists T_q \in \mathcal{T}$ such that $|\mathcal{C} \setminus T_q(q)| < c(q)\frac{|\mathcal{C}|-1}{GIC(\mathcal{C},c)+\gamma}$. Then define an effective oracle T with the property that $\forall q \in \mathcal{Q}, T(q) = T_q(q)$. We have thus defined an oracle such that $\forall R \subseteq \mathcal{Q}, \sum_{q \in R} c(q) \leq GIC(\mathcal{C}, c) + \gamma \Rightarrow$

$$\begin{aligned}|\mathcal{C} \cap T(R)| = |\mathcal{C}| - |\mathcal{C} \setminus T(R)| \geq |\mathcal{C}| - \sum_{q \in R} |\mathcal{C} \setminus T_q(q)| \\ > |\mathcal{C}| - \sum_{q \in R} c(q)\frac{|\mathcal{C}|-1}{GIC(\mathcal{C},c)+\gamma} \geq |\mathcal{C}| - (GIC(\mathcal{C},c)+\gamma)\frac{|\mathcal{C}|-1}{GIC(\mathcal{C},c)+\gamma} = 1.\end{aligned}$$

In particular, this contradicts the definition of $GIC(\mathcal{C}, c)$. □

This brings us to the main theorem of this section.

Theorem 2.1. *For any instance space* $\mathcal{X}$*, concept space* $\mathcal{C}$ *on* $\mathcal{X}$*, and cost function* c*,*

$$GIC(\mathcal{C}, c) \leq \mathit{CostComplexity}(\mathcal{C}, c) \leq GIC(\mathcal{C}, c) \log_2 |\mathcal{C}|$$

Proof. I begin with the lower bound. Let $k < GIC(\mathcal{C}, c)$. By definition of GIC, $\exists T \in \mathcal{T}$, such that $\forall R \subseteq \mathcal{Q}, \sum_{q \in R} c(q) \leq k \Rightarrow |\mathcal{C} \cap T(R)| > 1$. In particular, this implies that an adversarial teacher can answer any sequence of queries with cost no greater than k in a way that leaves at least 2 concepts in $\mathcal{C}$ consistent with the answers, either of which could be the target concept f. This implies *CostComplexity*$(\mathcal{C}, c) > k$. The limit as $k \to GIC(\mathcal{C}, c)$ gives the bound.

Next I prove the upper bound. If $GIC(\mathcal{C}, c) = \infty$ or $|\mathcal{C}| = \infty$, the bound holds vacuously, so let us assume these are finite. Say the teacher's answers correspond to some effective oracle $T \in \mathcal{T}$. Consider a recursive algorithm $\mathcal{A}_\gamma$ that makes effective queries from $\mathcal{Q}$.[6] If $|\mathcal{C}| = 1$, then $\mathcal{A}_\gamma$ halts and outputs the single remaining concept. Otherwise, let q be an effective query having the property guaranteed by Lemma 2.2. That is, $|\mathcal{C} \setminus T(q)| \geq c(q)\frac{|\mathcal{C}|-1}{GIC(\mathcal{C},c)+\gamma}$. Defining $V = \mathcal{C} \cap T(q)$ (a generalized notion of *version space*), this implies that $c(q) \leq (GIC(\mathcal{C}, c) + \gamma)\frac{|\mathcal{C}|-|V|}{|\mathcal{C}|-1}$ and $|V| < |\mathcal{C}|$. Say $\mathcal{A}_\gamma$ makes effective query q, and then recurses on V. In particular, we can immediately see that this algorithm identifies f using no more than $|\mathcal{C}| - 1$ queries.

I now prove by induction on $|\mathcal{C}|$ that *CostComplexity*$(\mathcal{C}, c) \leq (GIC(\mathcal{C}, c) + \gamma)H_{|\mathcal{C}|-1}$, where $H_n = \sum_{i=1}^{n} \frac{1}{i}$ is the n^{th} harmonic number. If $|\mathcal{C}| = 1$, then the cost complexity is 0. For $|\mathcal{C}| > 1$, *CostComplexity*$(\mathcal{C}, c)$

$$\begin{aligned}&\leq c(q) + \mathit{CostComplexity}(V, c) \\ &\leq (GIC(\mathcal{C}, c) + \gamma)\frac{|\mathcal{C}| - |V|}{|\mathcal{C}| - 1} + (GIC(V, c) + \gamma)H_{|V|-1} \\ &\leq (GIC(\mathcal{C}, c) + \gamma)\left(\frac{|\mathcal{C}| - |V|}{|\mathcal{C}| - 1} + H_{|V|-1}\right) \\ &\leq (GIC(\mathcal{C}, c) + \gamma)H_{|\mathcal{C}|-1}\end{aligned}$$

where the second inequality uses the inductive hypothesis along with the properties of q guaranteed by Lemma 2.2, and the third inequality uses Lemma 2.1. Finally, noting that $H_{|\mathcal{C}|-1} \leq \log_2 |\mathcal{C}|$ and taking the limit as $\gamma \to 0$ proves the theorem. □

[6] I use the definition of cost complexity in terms of effective cost, so that we need not concern ourselves with how $\mathcal{A}_\gamma$ chooses its *actual queries*. However, we could define $\mathcal{A}_\gamma$ to make actual queries with cost within γ of the effective query cost, so that the result still holds as $\gamma \to 0$.

2.3 An Example: Discrete Intervals

As a simple example of cost complexity, consider $\mathcal{X} = \{1, 2, \ldots, N\}$, for $N \geq 4$, $\mathcal{C} = \{h_{a,b} : \mathcal{X} \to \{0,1\} | a, b \in \mathcal{X}, a \leq b, \forall x \in \mathcal{X}, [a \leq x \leq b \Leftrightarrow h_{a,b}(x) = 1]\}$, and define an effective cost function c that is 1 for membership queries $q_{\{x\}}$ for any $x \in \mathcal{X}$, k for the positive example query $q_{\mathcal{X}}$ where $3 \leq k \leq N-1$, and ∞ for any other queries. In this case, $GIC(\mathcal{C}, c) = k+1$. In the spy game, say the teacher answers effective queries with an effective oracle T. Let $\mathcal{X}_+ = \{x | x \in \mathcal{X}, T(q_{\{x\}}) = \{h | h \in \mathcal{C}^*, h(x) = 1\}\}$. If $\mathcal{X}_+ \neq \varnothing$, then let $a = \min \mathcal{X}_+$ and $b = \max \mathcal{X}_+$. The spy tells the learner to make queries $q_{\{a\}}$, $q_{\{b\}}$, $q_{\{a-1\}}$ (if $a > 1$), and $q_{\{b+1\}}$ (if $b < N$). This narrows the version space to $\{h_{a,b}\}$, at a worst-case effective cost of 4. If $\mathcal{X}_+ = \varnothing$, then the spy suggests query $q_{\mathcal{X}}$. If $T(q_{\mathcal{X}}) = \{f_-\}$, the "all 0" concept, then no concepts in $\mathcal{C}$ are consistent. Otherwise, $T(q_{\mathcal{X}}) = \{h | h \in \mathcal{C}^*, h(x) = 1\}$ for some $x \in \mathcal{X}$, and the spy suggests membership query $q_{\{x\}}$. In this case, $T(q_{\{x\}}) \cap T(q_{\mathcal{X}}) = \varnothing$, so the worst-case cost is $k+1$ (without $q_{\mathcal{X}}$, it would cost $N-1$). These are the only cases to consider, so $GIC(\mathcal{C}, c) = k+1$. By Theorem 2.1, this implies $k+1 \leq \mathit{CostComplexity}(\mathcal{C}, c) \leq 2(k+1)\log_2 N$.

We can slightly improve this by noting that we only use $q_{\mathcal{X}}$ once. Specifically, if a learning algorithm begins (in the regular setting) by asking $q_{\mathcal{X}}$, revealing that $f(x) = 1$ for some $x \in \mathcal{X}$, then we can reduce to two disjoint learning problems, with concept spaces $\mathcal{C}'_1 = \{h_{x,b} | b \in \{x, \ldots, N\}\}$, and $\mathcal{C}'_2 = \{h_{a,x} | a \in \{1, 2, \ldots, x\}\}$, with cost functions $c_1(q) = c(q)$ for $q \in \{q_{\{x\}}, q_{\{x+1\}}, \ldots, q_{\{N\}}\}$ and ∞ otherwise, and $c_2(q) = c(q)$ for $q \in \{q_{\{1\}}, q_{\{2\}}, \ldots, q_{\{x\}}\}$ and ∞ otherwise, and corresponding $GIC(\mathcal{C}'_1, c) \leq 2$, $GIC(\mathcal{C}'_2, c) \leq 2$. So we can say that $\mathit{CostComplexity}(\mathcal{C}, c) \leq k + \mathit{CostComplexity}(\mathcal{C}'_1, c_1) + \mathit{CostComplexity}(\mathcal{C}'_2, c_2) \leq k + 4\log_2 N$. One algorithm that achieves this begins by making the positive example query, and then performs binary search above and below the indicated positive example to find the boundaries.

3 Pool-Based Active PAC Learning

In many scenarios, a more realistic definition of learning is that supplied by the Probably Approximately Correct (PAC) model. In this case, unlike the previous section, we are interested only in discovering with high probability a function with behavior very *similar* to the target concept on examples sampled from some distribution. Formally, as above there is an instance space $\mathcal{X}$, and a concept space $\mathcal{C} \subseteq \mathcal{C}^*$ on $\mathcal{X}$; unlike above, there is also a distribution $\mathcal{D}$ over $\mathcal{X}$, and I assume $\mathcal{C}$ is well-behaved in a measure-theoretic sense[7]. As with Exact Learning, the learning algorithm interacts with a teacher by making queries. However, in this setting the learning algorithm is given as input a finite sequence[8] of unlabeled examples $\mathcal{U}$, each drawn independently according to $\mathcal{D}$, and *all queries* made by the algorithm must concern only the behavior of the target concept on examples in $\mathcal{U}$.Formally, a *data-dependent cost function* is any function $c : \tilde{Q} \times 2^{\mathcal{X}} \to (\alpha, \infty]$. For a given set of unlabeled examples $\mathcal{U}$, and data-dependent cost function c, define $c_{\mathcal{U}}(\cdot) = c(\cdot, \mathcal{U})$. Thus, $c_{\mathcal{U}}$ is a cost function in the sense of the previous section. For a given $c_{\mathcal{U}}$, the corresponding effective cost function $c_{\mathcal{U}} : \mathcal{Q} \to [\alpha, \infty]$ is defined as in the previous section.

Definition 3.1. *Let $\mathcal{X}$ be an instance space, $\mathcal{C}$ a concept space on $\mathcal{X}$, and $\mathcal{U} = (x_1, x_2, \ldots, x_{|\mathcal{U}|})$ a finite sequence of unlabeled examples. Define $\forall h \in \mathcal{C}, h(\mathcal{U}) = (h(x_1), h(x_2), \ldots, h(x_{|\mathcal{U}|}))$. Define[9] $\mathcal{C}[\mathcal{U}] \subseteq \mathcal{C}$ as any concept space such that $\forall h \in \mathcal{C}, |\{h' | h' \in \mathcal{C}[\mathcal{U}], h'(\mathcal{U}) = h(\mathcal{U})\}| = 1$.*

Definition 3.2. *A* sample-based cost function *is any data-dependent cost function c such that for all finite $\mathcal{U} \subseteq \mathcal{X}, \forall q \in \mathcal{Q}$,*

$$c_{\mathcal{U}}(q) < \infty \Rightarrow \forall f \in \mathcal{C}^*, \forall a \in q(f), \forall h \in \mathcal{C}^*, [h(\mathcal{U}) = f(\mathcal{U}) \Rightarrow h \in a].$$

This corresponds to queries that are about *the target concept's labels on some subset of $\mathcal{U}$. Additionally, $\forall \mathcal{U} \subseteq \mathcal{X}, x \in \mathcal{X}$, and $q \in \mathcal{Q}$, $c(q, \mathcal{U} \cup \{x\}) \leq c(q, \mathcal{U})$. That is, in addition to the above property, adding extra examples to which q's answers do not refer does not increase its cost.*

[7] This mild assumption has almost no practical impact. See [6] for a full description.

[8] I will implicitly overload all notation for sets and sequences, so that if a set is used where a sequence is required, then an arbitrary ordering of the set is implied (though this ordering should be used consistently), and if a sequence is used where a set is required, then the set of distinct elements of the sequence is implied.

[9] The choice of which concept from each equivalence class to include in $\mathcal{C}[\mathcal{U}]$ can be made arbitrarily.

For example, membership queries on $x \in \mathcal{U}$ and positive examples queries on $S \subseteq \mathcal{U}$ could have finite costs under a sample-based cost function. As in the previous section, there is a target concept $f \in \mathcal{C}$, but unlike that section, we do not try to *identify* f, but instead attempt to *approximate* it with high probability.

Definition 3.3. *For instance space $\mathcal{X}$, concept space $\mathcal{C}$ on $\mathcal{X}$, distribution $\mathcal{D}$ on $\mathcal{X}$, target concept $f \in \mathcal{C}$, and concept $h \in \mathcal{C}$, define the* error rate *of h, denoted $error_{\mathcal{D}}(h, f)$, as*

$$error_{\mathcal{D}}(h, f) = \mathcal{P}r_{X \sim \mathcal{D}} \{h(X) \neq f(X)\}$$

Definition 3.4. *For $(\epsilon, \delta) \in (0, 1)^2$, an (ϵ, δ)-*learning algorithm *for $\mathcal{C}$ using sample-based cost function c is any algorithm $\mathcal{A}$ taking as input a finite sequence of unlabeled examples, such that for any target concept $f \in \mathcal{C}$ and finite sequence $\mathcal{U}$, $\mathcal{A}(\mathcal{U})$ outputs a concept in $\mathcal{C}$ after making a finite number of actual queries with finite costs under $c_{\mathcal{U}}$. Additionally, any (ϵ, δ)-*learning algorithm *$\mathcal{A}$ has the property that $\exists m \in [0, \infty)$ such that, for any target concept $f \in \mathcal{C}$ and distribution $\mathcal{D}$ on $\mathcal{X}$,*

$$\mathcal{P}r_{\mathcal{U} \sim \mathcal{D}^m} \{error_{\mathcal{D}}(\mathcal{A}(\mathcal{U}), f) > \epsilon\} \leq \delta.$$

*A concept space $\mathcal{C}$ is (ϵ, δ)-*learnable *given sample-based cost function c using total cost t if there exists an (ϵ, δ)-learning algorithm $\mathcal{A}$ for $\mathcal{C}$ using c such that for all finite example sequences $\mathcal{U}$, $\mathcal{A}(\mathcal{U})$ is guaranteed to have the sum of costs of the queries it makes at most t under $c_{\mathcal{U}}$.*

Definition 3.5. *For any instance space $\mathcal{X}$, concept space $\mathcal{C}$ on $\mathcal{X}$, sample-based cost function c, and $(\epsilon, \delta) \in (0, 1)^2$, define the (ϵ, δ)-*cost complexity*, denoted CostComplexity$(\mathcal{C}, c, \epsilon, \delta)$, as the infimum $t \geq 0$ such that $\mathcal{C}$ is (ϵ, δ)-learnable given c using total cost no greater than t.*

As in the previous section, because it is the *limiting* case, we can equivalently define the (ϵ, δ)-cost complexity as the infimum $t \geq 0$ such that there is an (ϵ, δ)-learning algorithm guaranteed to have the sum of *effective* costs of the *effective* queries it makes at most t.

The main results from this section include a new combinatorial quantity $GPIC(\mathcal{C}, c, m, \tau)$ such that if d is the VC-dimension of $\mathcal{C}$, then

$$GPIC(\mathcal{C}, c, \Theta(\tfrac{1}{\epsilon}), \delta) \leq \textit{CostComplexity}(\mathcal{C}, c, \epsilon, \delta) \leq GPIC(\mathcal{C}, c, \tilde{\Theta}\left(\tfrac{d}{\epsilon}\right), 0)\tilde{\Theta}(d).$$

3.1 Related Work

Previous work on pool-based active learning in the PAC model has been restricted almost exclusively to uniform-cost membership queries on examples in the unlabeled set $\mathcal{U}$. There has been some recent progress on query complexity bounds for that restricted setting. Specifically, Dasgupta [7] analyzes a greedy active learning scheme and derives bounds for the number of membership queries in $\mathcal{U}$ it uses under an *average case* setting, in which the target concept is selected randomly from a known distribution. A similar type of analysis was previously given by Freund et al. [8] to prove positive results for the Query by Committee algorithm. In a subsequent paper, Dasgupta [9] derives upper and lower bounds on the number of membership queries in $\mathcal{U}$ required for active learning for any particular distribution $\mathcal{D}$, under the assumption that $\mathcal{D}$ is known. The results I derive in this section imply *worst-case* results (over both $\mathcal{D}$ and f) for this as a special case of more general bounds applying to *any* sample-based cost function.

3.2 Cost Complexity Upper Bounds

I now derive bounds on the cost complexity of pool-based Active PAC Learning.

Definition 3.6. *For an instance space $\mathcal{X}$, concept space $\mathcal{C}$ on $\mathcal{X}$, sample-based cost function c, and nonnegative integer m, define the* General Identification Cost Growth Function*, denoted $GIC(\mathcal{C}, c, m)$, as follows.*

$$GIC(\mathcal{C}, c, m) = \sup_{\mathcal{U} \in \mathcal{X}^m} GIC(\mathcal{C}[\mathcal{U}], c_{\mathcal{U}})$$

Definition 3.7. *For any instance space $\mathcal{X}$, concept space $\mathcal{C}$ on $\mathcal{X}$, and $(\epsilon, \delta) \in (0, 1)^2$, let $M(\mathcal{C}, \epsilon, \delta)$ denote the* sample complexity *of $\mathcal{C}$ (in the classic* passive learning *sense), or the smallest m such that there is an algorithm $\mathcal{A}$ taking as input a set of examples $\mathcal{L}$ and labels, and outputting a classifier* (without making any queries)*, such that for any $\mathcal{D}$ and $f \in \mathcal{C}$,*

$$\mathcal{P}r_{\mathcal{L} \sim \mathcal{D}^m} \{error_{\mathcal{D}}(\mathcal{A}(\mathcal{L}, f(\mathcal{L})), f) > \epsilon\} \leq \delta.$$

It is known (e.g., [10]) that

$$\max\{\frac{d-1}{32\epsilon}, \frac{1}{2\epsilon}\ln\frac{1}{\delta}\} \leq M(\mathcal{C},\epsilon,\delta) \leq \frac{4d}{\epsilon}\ln\frac{12}{\epsilon} + \frac{4}{\epsilon}\ln\frac{2}{\delta}$$

for $0 < \epsilon < 1/8, 0 < \delta < .01$, and $d \geq 2$, where d is the VC-dimension of $\mathcal{C}$. Furthermore, Warmuth has conjectured [11] that $M(\mathcal{C},\epsilon,\delta) = \Theta(\frac{1}{\epsilon}(d + \log\frac{1}{\delta}))$.

With these definitions in mind, we have the following novel theorem.

Theorem 3.1. *For any instance space $\mathcal{X}$, concept space $\mathcal{C}$ on $\mathcal{X}$ with VC-dimension $d \in (0,\infty)$, sample-based cost function c, $\epsilon \in (0,1)$, and $\delta \in (0,\frac{1}{2})$, if $m = M(\mathcal{C},\epsilon,\delta)$, then*

$$\mathit{CostComplexity}(\mathcal{C},c,\epsilon,\delta) \leq GIC(\mathcal{C},c,m)d\log_2\frac{em}{d}$$

Proof. For the unlabeled sequence, sample $\mathcal{U} \sim \mathcal{D}^m$. If $GIC(\mathcal{C},c,m) = \infty$, then the upper bound holds vacuously, so let us assume this is finite. Also, $d \in (0,\infty)$ implies $|\mathcal{U}| \in (0,\infty)$ [10]. By definition of $M(\mathcal{C},\epsilon,\delta)$, there exists a (passive learning) algorithm $\mathcal{A}$ such that $\forall f \in \mathcal{C}, \forall\mathcal{D}, \mathcal{P}r_{\mathcal{U}\sim\mathcal{D}^m}\{error_{\mathcal{D}}(\mathcal{A}(\mathcal{U},f(\mathcal{U})),f) > \epsilon\} \leq \delta$. Therefore any algorithm that, by a finite sequence of effective queries with finite cost under $c_{\mathcal{U}}$, identifies $f(\mathcal{U})$ and then outputs $\mathcal{A}(\mathcal{U},f(\mathcal{U}))$, is an (ϵ,δ)-learning algorithm for $\mathcal{C}$ using c.

Suppose now that there is a *ghost teacher*, who knows the teacher's target concept $f \in \mathcal{C}$. The ghost teacher uses the $h \in \mathcal{C}[\mathcal{U}]$ with $h(\mathcal{U}) = f(\mathcal{U})$ as its target concept. In order to answer any actual queries $\tilde{q} \in \tilde{Q}$ with $c_{\mathcal{U}}(\tilde{q}) < \infty$, the ghost teacher simply passes the query to the real teacher and then answers the query using the real teacher's answer. This answer is guaranteed to be valid because $c_{\mathcal{U}}$ is a sample-based cost function. Thus, identifying $f(\mathcal{U})$ can be accomplished by identifying $h(\mathcal{U})$, which can be accomplished by identifying h. The task of identifying h can be reduced to an *Exact Learning* task with concept space $\mathcal{C}[\mathcal{U}]$ and cost function $c_{\mathcal{U}}$, where the teacher for the Exact Learning task is the ghost teacher. Therefore, by Theorem 2.1, the total cost required to identify $f(\mathcal{U})$ with a finite sequence of queries is no greater than

$$\mathit{CostComplexity}(\mathcal{C}[\mathcal{U}],c_{\mathcal{U}}) \leq GIC(\mathcal{C}[\mathcal{U}],c_{\mathcal{U}})\log_2|\mathcal{C}[\mathcal{U}]| \leq GIC(\mathcal{C}[\mathcal{U}],c_{\mathcal{U}})d\log_2\frac{|\mathcal{U}|e}{d}, \quad (1)$$

where the last inequality is due to Sauer's Lemma (e.g., [10]). Finally, taking the worst case (supremum) over all $\mathcal{U} \in \mathcal{X}^m$ completes the proof. □

Note that (1) also implies a data-dependent bound, which could potentially be useful for practical applications in which the unlabeled examples are available when bounding the cost. It can also be used to state a distribution-dependent bound.

3.3 An Example: Intersection-Closed Concept Spaces

As an example application, we can use the above theorem to prove new results for any intersection-closed concept space[10] as follows.

Lemma 3.1. *For any instance space $\mathcal{X}$, intersection-closed concept space $\mathcal{C}$ with VC-dimension $d \geq 1$, sample-based cost function c such that membership queries in $\mathcal{U}$ have cost $\leq \mu$ (i.e., $\forall\mathcal{U} \subseteq \mathcal{X}, x \in \mathcal{U}, c_{\mathcal{U}}(q_{\{x\}}) \leq \mu$) and positive example queries in $\mathcal{U}$ have cost $\leq \kappa$ (i.e., $\forall\mathcal{U} \subseteq \mathcal{X}, S \subseteq \mathcal{U}, c_{\mathcal{U}}(q_S) \leq \kappa$), and integer $m \geq 0$,*

$$GIC(\mathcal{C},c,m) \leq \kappa + \mu d$$

Proof. Say we have some set of unlabeled examples $\mathcal{U}$, and consider bounding the value of $GIC(\mathcal{C}[\mathcal{U}],c_{\mathcal{U}})$. In the spy game, suppose the teacher is answering with effective oracle $T \in \mathcal{T}$. Let $\mathcal{U}_+ = \{x|x \in \mathcal{U}, T(q_{\{x\}}) = \{h|h \in \mathcal{C}^*, h(x) = 1\}\}$. The spy first tells the learner to make the $q_{\mathcal{U}\setminus\mathcal{U}_+}$ query (if $\mathcal{U} \setminus \mathcal{U}_+ \neq \varnothing$). If $\exists x \in \mathcal{U} \setminus \mathcal{U}_+$ s.t. $T(q_{\mathcal{U}\setminus\mathcal{U}_+}) = \{h|h \in \mathcal{C}^*, h(x) = 1\}$, then the spy tells the learner to make effective query $q_{\{x\}}$ for this x, and there are no concepts in $\mathcal{C}[\mathcal{U}]$ consistent with the answers to these two queries; the total effective cost for this case is $\kappa + \mu$. If this is not the case, but $|\mathcal{U}_+| = 0$, then there is at most one concept in $\mathcal{C}[\mathcal{U}]$ consistent with the

[10] An intersection-closed concept space $\mathcal{C}$ has the property that for any $h_1, h_2 \in \mathcal{C}$, there is a concept $h_3 \in \mathcal{C}$ such that $\forall x \in \mathcal{X}, [h_1(x) = h_2(x) = 1 \Leftrightarrow h_3(x) = 1]$. For example, conjunctions and axis-aligned rectangles are intersection-closed.

answer to $q_{\mathcal{U}\setminus\mathcal{U}_+}$: namely, the $h \in \mathcal{C}[\mathcal{U}]$ with $h(x) = 0$ for all $x \in \mathcal{U}$, if there is such an h. In this case, the cost is just κ.

Otherwise, let $\bar{S}$ be a largest subset of $\mathcal{U}_+$ such that $\exists h \in \mathcal{C}$ with $\forall x \in \bar{S}, h(x) = 1$. If $\bar{S} = \varnothing$, then making any membership query in $\mathcal{U}_+$ leaves all concepts in $\mathcal{C}[\mathcal{U}]$ inconsistent (at cost μ), so let us assume $\bar{S} \neq \varnothing$. For any $S \subseteq \mathcal{X}$, define

$$CLOS(S) = \{x | x \in \mathcal{X}, \forall h \in \mathcal{C}, [\forall y \in S, h(y) = 1] \Rightarrow h(x) = 1\}$$

the *closure* of S. Let $\bar{S}'$ be a smallest subset of $\bar{S}$ such that $CLOS(\bar{S}') = CLOS(\bar{S})$, known as a *minimal spanning set* of $\bar{S}$ [12]. The spy now tells the learner to make queries $q_{\{x\}}$ for all $x \in \bar{S}'$.

Any concept in $\mathcal{C}$ consistent with the answer to $q_{\mathcal{U}\setminus\mathcal{U}_+}$ must label every $x \in \mathcal{U} \setminus \mathcal{U}_+$ as 0. Any concept in $\mathcal{C}$ consistent with the answers to the membership queries on $\bar{S}'$ must label every $x \in CLOS(\bar{S}') = CLOS(\bar{S}) \supseteq \bar{S}$ as 1. Additionally, every concept in $\mathcal{C}$ that labels every $x \in \bar{S}$ as 1 must label every $x \in \mathcal{U}_+ \setminus \bar{S}$ as 0, since $\bar{S}$ is defined to be maximal. This labeling of these three sets completely defines a labeling of $\mathcal{U}$, and as such there is at most one $h \in \mathcal{C}[\mathcal{U}]$ consistent with the answers to all queries made by the learner. Helmbold, Sloan, and Warmuth [12] proved that, for an intersection-closed concept space with VC-dimension d, for any set $\bar{S}$, all minimal spanning sets of $\bar{S}$ have size at most d. This implies the learner makes at most d membership queries in $\mathcal{U}$, and thus has a total cost of at most $\kappa + \mu d$. □

Corollary 3.1. *Under the conditions of Lemma 3.1, if* $d \geq 10$, *then for* $0 < \epsilon < 1$, *and* $0 < \delta < \frac{1}{2}$,

$$\textit{CostComplexity}(\mathcal{C}, c, \epsilon, \delta) \leq (\kappa + \mu d) d \log_2 \left(\frac{e}{d} \max \left\{ \frac{16d}{\epsilon} \ln d, \frac{6}{\epsilon} \ln \frac{28}{\delta} \right\} \right)$$

Proof. This follows from Theorem 3.1, Lemma 3.1, and Auer & Ortner's result [13] that for intersection-closed concept spaces with $d \geq 10$, $M(\mathcal{C}, \epsilon, \delta) \leq \max\left\{\frac{16d}{\epsilon} \ln d, \frac{6}{\epsilon} \ln \frac{28}{\delta}\right\}$. □

For example, consider the concept space of axis-parallel hyper-rectangles in $\mathcal{X} = \mathbb{R}^n$, $\mathcal{C} = \{h : \mathcal{X} \rightarrow \{0,1\} | \exists((a_1, b_1), (a_2, b_2), \ldots, (a_n, b_n)) : \forall x \in \mathbb{R}^n, h(x) = 1 \Leftrightarrow \forall i \in \{1, 2, \ldots, n\}, a_i \leq x_i \leq b_i\}$. One can show that this is an intersection-closed concept space with VC-dimension $2n$. For a sample-based cost function c of the form stated in Lemma 3.1, we have that *CostComplexity*$(\mathcal{C}, c, \epsilon, \delta) \leq \tilde{O}\left((\kappa + n\mu)n\right)$. Unlike the example in the previous section, if all other query types have infinite cost, then for $n \geq 2$ there are distributions that force any algorithm achieving this bound for small ϵ and δ to use multiple positive example queries q_S with $|S| > 1$. In particular, for finite constant κ, this is an exponential improvement over the cost complexity of PAC active learning with only uniform cost membership queries on $\mathcal{U}$.

3.4 A Cost Complexity Lower Bound

At first glance, it might seem that $GIC(\mathcal{C}, c, \left\lceil \frac{1-\epsilon}{\epsilon} \right\rceil)$ could be a lower bound on *CostComplexity*$(\mathcal{C}, c, \epsilon, \delta)$. In fact, one can show this is true for $\delta < (\frac{\epsilon d}{e})^d$. However, there are simple examples for which this is not a lower bound for general ϵ and δ.[11] We therefore require a slight modification of GIC to introduce dependence on δ.

Definition 3.8. *For an instance space $\mathcal{X}$, finite concept space $\mathcal{C}$ on $\mathcal{X}$, cost function c, and $\delta \in [0, 1)$, define the* General Partial Identification Cost, *denoted $GPIC(\mathcal{C}, c, \delta)$ as follows.*

$$GPIC(\mathcal{C}, c, \delta) = \inf\{t | t \geq 0, \forall T \in \mathcal{T}, \exists R \subseteq \mathcal{Q}, \textit{ s.t. } [\textstyle\sum_{q \in R} c(q) \leq t] \wedge [|\mathcal{C} \cap T(R)| \leq \delta|\mathcal{C}| + 1]\}$$

Definition 3.9. *For an instance space $\mathcal{X}$, concept space $\mathcal{C}$ on $\mathcal{X}$, sample-based cost function c, non-negative integer m, and $\delta \in [0, 1)$, define the* General Partial Identification Cost Growth Function, *denoted $GPIC(\mathcal{C}, c, m, \delta)$, as follows.*

$$GPIC(\mathcal{C}, c, m, \delta) = \sup_{\mathcal{U} \in \mathcal{X}^m} GPIC(\mathcal{C}[\mathcal{U}], c_{\mathcal{U}}, \delta)$$

[11] The infamous "Monty Hall" problem is an interesting example of this. For another example, consider $\mathcal{X} = \{1, 2, \ldots, N\}$, $\mathcal{C} = \{h_x | x \in \mathcal{X}, \forall y \in \mathcal{X}, h_x(y) = I[x = y]\}$, and cost that is 1 for membership queries in $\mathcal{U}$ and infinite for other queries. Although $GIC(\mathcal{C}, c, N) = N - 1$, it is possible to achieve better than $\epsilon = \frac{1}{N+1}$ with probability close to $\frac{N-2}{N-1}$ using cost no greater than $N - 2$.

It is easy to see that $GIC(\mathcal{C}, c) = GPIC(\mathcal{C}, c, 0)$ and $GIC(\mathcal{C}, c, m) = GPIC(\mathcal{C}, c, m, 0)$, so that all of the above results could be stated in terms of $GPIC$.

Theorem 3.2. *For any instance space $\mathcal{X}$, concept space $\mathcal{C}$ on $\mathcal{X}$, sample-based cost function c, $(\epsilon, \delta) \in (0, 1)^2$, and any $V \subseteq \mathcal{C}$,*

$$GPIC(V, c, \left\lceil \tfrac{1-\epsilon}{\epsilon} \right\rceil, \delta) \leq \textit{CostComplexity}(\mathcal{C}, c, \epsilon, \delta)$$

Proof. Let $S \subseteq \mathcal{X}$ be a set with $1 \leq |S| \leq \left\lceil \frac{1-\epsilon}{\epsilon} \right\rceil$, and let $\mathcal{D}_S$ be the uniform distribution on S. Thus, $error_{\mathcal{D}_S}(h, f) \leq \epsilon \Leftrightarrow h(S) = f(S)$. I will show that any algorithm $\mathcal{A}$ guaranteeing $\mathcal{P}r_{\mathcal{U} \sim \mathcal{D}_S^m}\{error_{\mathcal{D}_S}(\mathcal{A}(\mathcal{U}), f) > \epsilon\} \leq \delta$ cannot also guarantee cost strictly less than $GPIC(V[S], c_S, \delta)$. If $\delta|V[S]| \geq |V[S]| - 1$, the result is clear since no algorithm guarantees cost less than 0, so assume $\delta|V[S]| < |V[S]| - 1$. Suppose $\mathcal{A}$ is an algorithm that guarantees, for every finite sequence $\mathcal{U}$ of elements from S, $\mathcal{A}(\mathcal{U})$ incurs total cost strictly less than $GPIC(V[S], c_S, \delta)$ under $c_{\mathcal{U}}$ (and therefore also under c_S). By definition of $GPIC$, $\exists \hat{T} \in \mathcal{T}$ such that for any set of queries R that $\mathcal{A}(\mathcal{U})$ makes, $|V[S] \cap \hat{T}(R)| > \delta|V[S]| + 1$. I now proceed by the probabilistic method. Say the teacher draws the target concept f uniformly at random from $V[S]$, and $\forall q \in \mathcal{Q}$ s.t. $f \in \hat{T}(q)$, answers with $\hat{T}(q)$. Any $q \in \mathcal{Q}$ such that $f \notin \hat{T}(q)$ can be answered with an arbitrary $a \in q(f)$. Let $h_{\mathcal{U}} = \mathcal{A}(\mathcal{U})$; let $R_{\mathcal{U}}$ denote the set of queries $\mathcal{A}(\mathcal{U})$ would make if *all* queries were answered with $\hat{T}$.

$$\begin{aligned}
&\mathbb{E}_f[\mathcal{P}r_{\mathcal{U} \sim \mathcal{D}_S^m}\{error_{\mathcal{D}_S}(\mathcal{A}(\mathcal{U}), f) > \epsilon\}] \\
&\qquad = \mathbb{E}_{\mathcal{U} \sim \mathcal{D}_S^m}[\mathcal{P}r_f\{h_{\mathcal{U}}(S) \neq f(S)\}] \\
&\qquad \geq \mathbb{E}_{\mathcal{U} \sim \mathcal{D}_S^m}[\mathcal{P}r_f\{h_{\mathcal{U}}(S) \neq f(S) \wedge f \in \hat{T}(R_{\mathcal{U}})\}] \\
&\qquad \geq \min_{\mathcal{U} \in S^m} \frac{|V[S] \cap \hat{T}(R_{\mathcal{U}})| - 1}{|V[S]|} > \delta.
\end{aligned}$$

Therefore, there exists a deterministic method for selecting f and answering queries such that $\mathcal{P}r_{\mathcal{U} \sim \mathcal{D}_S^m}\{error_{\mathcal{D}_S}(\mathcal{A}(\mathcal{U}), f) > \epsilon\} > \delta$. In particular, this proves that there are no (ϵ, δ)-learning algorithms that guarantee cost strictly less than $GPIC(V[S], c_S, \delta)$. Taking the supremum over sets S completes the proof. □

Corollary 3.2. *Under the conditions of Theorem 3.2,*

$$GPIC(\mathcal{C}, c, \left\lceil \tfrac{1-\epsilon}{\epsilon} \right\rceil, \delta) \leq \textit{CostComplexity}(\mathcal{C}, c, \epsilon, \delta).$$

Equipped with Theorem 3.2, it is straightforward to prove the claim made in Section 3.3 that there are distributions forcing any (ϵ, δ)-learning algorithm for Axis-parallel rectangles using only membership queries (at cost μ) to pay $\Omega(\frac{\mu(1-\delta)}{\epsilon})$. The details are left as an exercise.

4 Discussion and Open Problems

Note that the usual “query counting” analysis done for Active Learning is a special case of cost complexity (uniform cost 1 on the allowed queries, infinite cost on the others). In particular, Theorem 3.1 can easily be specialized to give a worst-case bound on the query complexity for the widely studied setting in which the learner can make any *membership queries* on examples in $\mathcal{U}$ [9, 14]. However, for this special case, one can derive a slightly tighter bound. Following the proof technique of Hegedüs [4], one can show that for any sample-based cost function c such that $\forall \mathcal{U} \subseteq \mathcal{X}, q \in \mathcal{Q}$, $c_{\mathcal{U}}(q) < \infty \Rightarrow [c_{\mathcal{U}}(q) = 1 \wedge \forall f \in \mathcal{C}^*, |q(f)| = 1]$, $\textit{CostComplexity}(\mathcal{C}, c_{\mathcal{X}}) \leq 2\frac{GIC(\mathcal{C}, c_{\mathcal{X}}) \log_2 |\mathcal{C}|}{\log_2 GIC(\mathcal{C}, c_{\mathcal{X}})}$. This implies for the PAC setting that $\textit{CostComplexity}(\mathcal{C}, c, \epsilon, \delta) \leq 2\frac{GIC(\mathcal{C}, c, m) d \log_2 m}{\log_2 GIC(\mathcal{C}, c, m)}$, for VC-dimension $d \geq 3$ and $m = M(\mathcal{C}, \epsilon, \delta)$. This includes the cost function assigning 1 to membership queries on $\mathcal{U}$ and ∞ to all others.

Active Learning in the PAC model is closely related to the topic of *Semi-Supervised Learning*. Balcan & Blum [15] have recently derived a variety of sample complexity bounds for Semi-Supervised Learning. Many of the techniques can be transfered to the pool-based Active Learning setting in a fairly natural way. Specifically, suppose there is a quantitative notion of "compatibility" between a concept and a distribution, which can be estimated from a finite unlabeled sample. If we know the target concept is highly compatible with the data distribution, we can draw enough unlabeled examples to estimate compatibility, then identify and discard those concepts that are probably highly incompatible. The set of highly compatible concepts may be significantly less expressive, therefore reducing *both* the number of examples for which an algorithm must learn the labels to guarantee generalization *and* the number of labelings of those examples the algorithm must distinguish between, thereby also reducing the cost complexity.

There are a variety of interesting extensions of this framework worth pursuing. Perhaps the most natural direction is to move into the agnostic PAC framework, which has thus far been quite elusive for active learning except for a few results [16, 17]. Another possibility is to derive cost complexity bounds when the cost c is a function of not only the query, but also the target concept. Then every time the learning algorithm makes a query q, it is charged $c(q, f)$, but does not necessarily know what this value is. However, it can always upper bound the total cost so far by the worst case over concepts in the version space. Can anything interesting be said about this setting (or variants), perhaps under some benign smoothness constraints on $c(q, \cdot)$? This is of some practical importance since, for example, it is often more difficult to label examples that occur near a decision boundary.

References

[1] Balcázar, J.L., Castro, J., Guijarro, D.: A general dimension for exact learning. In: 14th Annual Conference on Learning Theory. (2001)

[2] Balcázar, J.L., Castro, J.: A new abstract combinatorial dimension for exact learning via queries. Journal of Computer and System Sciences **64** (2002) 2–21

[3] Hellerstein, L., Pillaipakkamnatt, K., Raghavan, V., Wilkins, D.: How many queries are needed to learn? Journal of the Association for Computing Machinery **43** (1996) 840–862

[4] Hegedüs, T.: Generalized teaching dimension and the query complexity of learning. In: 8th Annual Conference on Computational Learning Theory. (1995)

[5] Balcázar, J.L., Castro, J., Guijarro, D., Simon, H.U.: The consistency dimension and distribution-dependent learning from queries. In: Algorithmic Learning Theory. (1999)

[6] Blumer, A., Ehrenfeucht, A., Haussler, D., Warmuth, M.: Learnability and the vapnik-chervonenkis dimension. Journal of the Association for Computing Machinery **36** (1989) 929–965

[7] Dasgupta, S.: Analysis of a greedy active learning strategy. In: Advances in Neural Information Processing Systems (NIPS). (2004)

[8] Freund, Y., Seung, H.S., Shamir, E., Tishby, N.: Selective sampling using the query by committee algorithm. Machine Learning **28** (1997) 133–168

[9] Dasgupta, S.: Coarse sample complexity bounds for active learning. In: Advances in Neural Information Processing Systems (NIPS). (2005)

[10] Anthony, M., Bartlett, P.L.: Neural Network Learning: Theoretical Foundations. Cambridge University Press (1999)

[11] Warmuth, M.: The optimal pac algorithm. In: Conference on Learning Theory. (2004)

[12] Helmbold, D., Sloan, R., Warmuth, M.: Learning nested differences of intersection-closed concept classes. Machine Learning **5** (1990) 165–196

[13] Auer, P., Ortner, R.: A new PAC bound for intersection-closed concept classes. In: 17^{th} Annual Conference on Learning Theory (COLT). (2004)

[14] Tong, S., Koller, D.: Support vector machine active learning with applications to text classification. Journal of Machine Learning Research **2** (2001)

[15] Balcan, M.F., Blum, A.: A PAC-style model for learning from labeled and unlabeled data. In: Conference on Learning Theory. (2005)

[16] Balcan, M.F., Beygelzimer, A., Langford, J.: Agnostic active learning. In: 23rd International Conference on Machine Learning (ICML). (2006)

[17] Kääriäinen, M.: On active learning in the non-realizable case. In: NIPS Workshop on Foundations of Active Learning. (2005)