Bounded Memory Active Learning through Enriched Queries

Max Hopkins NMHOPKIN@ENG.UCSD.EDU
Department of Computer Science and Engineering, UCSD, California, CA 92092.

Daniel Kane DAKANE@ENG.UCSD.EDU
Department of Computer Science and Engineering / Department of Mathematics, UCSD, California, CA 92092.

Shachar Lovett SLOVETT@CS.UCSD.EDU
Department of Computer Science and Engineering, UCSD, California, CA 92092.

Michal Moshkovitz MMOSHKOVITZ@ENG.UCSD.EDU
Qualcomm Institute, UCSD, California, CA 92092.

Editors: Mikhail Belkin and Samory Kpotufe

Abstract

The explosive growth of easily-accessible unlabeled data has lead to growing interest in *active learning*, a paradigm in which data-hungry learning algorithms adaptively select informative examples in order to lower prohibitively expensive labeling costs. Unfortunately, in standard worst-case models of learning, the active setting often provides no improvement over non-adaptive algorithms. To combat this, a series of recent works have considered a model in which the learner may ask *enriched* queries beyond labels. While such models have seen success in drastically lowering label costs, they tend to come at the expense of requiring large amounts of memory. In this work, we study what families of classifiers can be learned in *bounded memory*. To this end, we introduce a novel streaming-variant of enriched-query active learning along with a natural combinatorial strategy called *lossless sample compression* that is sufficient for learning not only with bounded memory, but in a query-optimal and computationally efficient manner as well. Finally, we give three fundamental examples of classifier families with small, easy to compute lossless compression schemes when given access to basic enriched queries: axis-aligned rectangles, decision trees, and halfspaces in two dimensions.

Keywords: Active Learning, Bounded Memory, Enriched Queries, RPU-Learning

1. Introduction

Today's learning landscape is dominated mostly by data-hungry algorithms, each requiring a massive supply of labeled training samples in order to reach state of the art accuracy. Such algorithms are excellent when labeled data is cheap and plentiful, but in many important scenarios acquiring labels requires the use of human experts, making popular supervised methods like deep learning infeasible both in time and cost. In recent years, a framework meant to address this issue called *active learning* has gained traction in both theory and practice. Active learning posits that not all labeled samples are equal: some may be particularly informative, others useless. While a standard supervised (passive) learning algorithm receives a stream or pool of labeled training data, an active learner instead receives *unlabeled* data along with the ability to query an expert *labeling oracle*. By choosing only to query the most informative examples, the hope is that an active learner can achieve state of the art accuracy using only a small fraction of the labels required by passive techniques.

While active learning saw initial success with simple classifiers such as thresholds in one-dimension, it quickly became clear that inherent structural barriers barred it from improving substantially over the passive case even for very basic examples such as halfspaces in two-dimensions or axis-aligned rectangles (Dasgupta, 2005). A number of modifications to the model have been proposed to remedy this issue. One such strategy that has gained increasing traction in the past few years is empowering the learner to ask questions beyond simple label queries. One might ask the oracle, for instace, to *compare* two pieces of data rather than simply label them—the idea being that such additional information might break down the structural barriers inherent in standard lower bounds. Indeed, in 2017, Kane et al. (2017) showed not only how this was true for halfspaces,[1] but introduced a combinatorial complexity parameter called *inference dimension* to characterize when a family of classifiers is efficiently actively learnable with respect to some set of enriched queries.

The model proposed by Kane et al. (2017), however, is not without its downsides. One issue is that the model is *pool-based*, meaning the algorithm receives a large pool of unlabeled samples ahead of time, and can query or otherwise access any part of the sample at any time. This type of model can be infeasible in practice due to its unrealistic memory requirements—the learner is assumed to always have the full training data in storage. In this work, we aim to resolve this issue by studying when a family of classifiers can be efficiently active learned in *bounded memory*, meaning the amount of memory used by the algorithm should remain *constant* regardless of the desired accuracy. Such algorithms open up potential applications of active learning to scenarios where storage is severely limited, e.g. to smartphones and other mobile devices.

To this end, we introduce a new *streaming* variant of active learning with enriched queries in which the learner has access to a stream of unlabeled data and chooses one-by-one whether to store or forget points from the stream. Instead of having query access to the full training set at any time, our algorithm is then restricted to querying only points it has stored in memory. Along with this model, we introduce a natural strengthening of Littlestone and Warmuth (1986)'s *sample compression*, a standard learning technique for Valiant's Probably Approximately Correct (PAC) model (Valiant, 1984; Vapnik and Chervonenkis, 1974), called *lossless* sample compression, and show that any class with such a scheme may be learned query-optimally and with constant memory. In doing so, we make the first non-trivial advance towards answering an open question posed by Kane et al. (2017) regarding the existence of a combinatorial characterization for bounded-memory active learning. Further, we show that lossless compression schemes imply learnability not only in the standard PAC-model, but also in a much stronger sense known as *Reliable and Probably Useful Learning* (Rivest and Sloan, 1988). This model, which carries strong connections to the active learning paradigm (El-Yaniv and Wiener, 2012; Kane et al., 2017), demands that the learner makes no errors with the caveat that it may abstain from classifying a small fraction of examples.

Finally, we conclude by showing that a number of classifier families fundamental to machine learning exhibit small lossless compression schemes with respect to natural enriched queries. We focus in particular on three such classes: axis-aligned rectangles, decision trees, and halfspaces in two dimensions. In each of these three cases our lossless compression scheme is efficiently computable, resulting in computationally efficient as well as query-optimal and bounded memory learners. All three classes provide powerful examples of how natural enriched queries can turn fundamental learning problems from prohibitively expensive to surprisingly feasible.

1. We note their work requires structural assumptions on the data to work beyond two dimensions.

Before moving on, we give a brief roadmap of our work. In Section 2, we discuss preliminaries, including our learning and enriched query models. In Section 3.1, we introduce lossless compression schemes and give an overview of the proof that they are a sufficient condition for bounded memory RPU-learning. In Section 3.2 we introduce natural enriched queries for three fundamental classes, axis-aligned rectangles, decision trees, and halfspaces in two dimensions, and show they have small, efficiently computable lossless compression schemes. In the Appendix we cover related work (Section 4), further research directions (Appendix C), and give in-detail proofs regarding lossless compression (Appendix A) and our example constructions (Appendix B).

2. Preliminaries

2.1. Reliable and Probably Useful Learning

Let X be a set and H a family of binary classifiers on X. We study the learnability of a pair (X, H) (called a *hypothesis class*) under a strong model introduced by Rivest and Sloan (1988) called *Reliable and Probably Useful Learning* (RPU-Learning). Unlike the more standard PAC setting, RPU-Learning requires that the learner never makes an error. To compensate for this stringent requirement, the learner may respond "I don't know", denoted "$\perp$," on a small fraction of the space. In the standard, passive version of RPU-Learning, the learner has access to a labeled sample oracle from an adversarily chosen distribution. The goal is to analyze the number of labeled samples required from this oracle to learn almost all inputs with high probability.

Definition 1 (RPU-Learning) *A hypothesis class (X, H) is RPU-Learnable with sample complexity $n(\varepsilon, \delta)$ if there exists for all $\varepsilon, \delta > 0$ a learning algorithm A such that for any choice of distribution D over X and $h \in H$, the learner is:*

1. *Probably useful:*
$$\Pr_{S \sim D^{n(\varepsilon,\delta)}} \left[\Pr_{x \sim D} [A(S, h(S))(x) = \perp] > \varepsilon \right] < \delta,$$
2. *Reliable:*
$$\forall S, x \text{ s.t. } A(S, h(S))(x) \neq \perp, A(S, h(S))(x) = h(x),$$

where $S, h(S)$ is shorthand for the set of labeled samples $(x, h(x))$ for $x \in S$.

In other words, the learner outputs a label with high probability, and never makes a mistake. This model of learning is substantially stronger than the more standard PAC model, which need only be approximately correct. In fact, it is known that RPU-learning with only labels often has infeasibly large (or even infinite) sample complexity (Kivinen, 1995, 1990; Hopkins et al., 2020b). Recently, Kane et al. (2017) proved that this barrier can be broken by allowing the learner to ask enriched questions, and gave an efficient algorithm for RPU-learning halfspaces in two dimensions (later extended in Hopkins et al. (2020a) to arbitrary dimensions with suitable distributional assumptions). While the algorithms in these works give a substantial improvement over previous impossibility results, they come with a practical caveat: reaching high accuracy guarantees requires an infeasible amount of storage. In this work we show not only how to build efficient RPU-learning algorithms for a broader range of queries and hypothesis classes, but also show that this strong model remains surprisingly feasible even in scenarios where memory is severely limited.

2.2. Active Learning

Unfortunately, even with the addition of enriched queries, it is not in general possible to RPU (or even PAC) learn in fewer than $\text{poly}(1/\varepsilon)$ labeled samples. In cases where labels are prohibitively expensive (e.g. medical imagery), this creates a substantial barrier to learning even the simplest classes such as 1D-thresholds. To side-step this issue, we consider the well-studied model of *active learning*. Unlike the previously discussed (passive) models, an active learner receives *unlabeled samples* from the distribution and may choose whether or not to send each sample to a labeling oracle. The overall complexity of learning, called *query complexity*, is then measured not by the total number of samples, but by the number of calls to the labeling oracle required to achieve the guarantees laid out in Definition 1.

One might hope the additional adaptivity afforded by active learning allows scenarios with high labeling cost to become feasible, and indeed it does in some basic or restricted scenarios, lowering the overall complexity from $\text{poly}(1/\varepsilon)$ to $\text{poly}(\log(1/\varepsilon))$ (see e.g. Settles (2009), Dasgupta (2011) or Hanneke et al. (2014)). Unfortunately, it has long been known that active learning fails to give any substantial improvement for fundamental classes such as halfspaces, even in the weaker PAC-model (Dasgupta, 2005). We continue a recent line of work showing this barrier may be broken by the same technique that permits efficient RPU-learning: asking more informative questions.

2.3. Enriched Queries

Instead of having access only to a labeling oracle, the learners we discuss in this work will have the ability to ask a range of natural questions regarding the data. Enriched queries we discuss will be defined on a fixed input size we denote by k. A label query, for instance, is defined on a single point and thus has $k = 1$. A comparison between two points would have $k = 2$. Further, while we only consider binary labels, we will in general allow enriched queries to have an arbitrary arity r denoting the total number of possible answers. Binary queries like labels or comparisons have $r = 2$, but other questions we study (e.g. asking "what is wrong with (example) x?") might draw from many possible answers. In fact, such questions might even have multiple valid answers at once—a meal, for instance, can easily be both too salty and undercooked!

To formalize these notions, we consider each type of query to be an oracle of the form:

$$\mathcal{O} : H \times X^k \to P([r]) \setminus \{\varnothing\},$$

where $P([r])$ denotes the powerset of $[r] = \{1, \ldots, r\}$, and $\mathcal{O}(h, T) \subseteq [r]$ denotes the set of valid responses to the query represented by $\mathcal{O}$ on T under hypothesis h. Since in practice a user is unlikely to give the complete list of valid responses in $\mathcal{O}(h, T)$, we do not allow the learner direct access to the oracle response. Instead, when the learner queries T the adversary sends back a valid response from $\mathcal{O}(h, T)$.

In this work, we consider hypothesis classes (X, H) endowed with a collection of oracles $\{\mathcal{O}_i\}_{i=1}^{\ell}$, which we collectively denote as the *query set* Q.[2] While each oracle in Q is defined only on a certain fixed sample size, we will often wish to make every possible query associated to some larger sample $S \subset X$. In particular, given a hypothesis $h \in H$, we denote by $Q_h(S)$ the set

2. There is a slight subtlety here in defining H that we have swept under the rug. In the enriched query model, $h \in H$ should really be defined by its output across *all* enriched queries, not just across labels. This is slightly more general than what we have described, and allows for scenarios where $h \in H$ may not be uniquely determined by its labels (such as when each h really corresponds to some underlying richer structure like a hyperplane or decision tree).

of all possible responses to queries on S.[3] We will generally think of the learner making queries like this in batches on a larger set S, and receiving some $q(S) \in Q_h(S)$ from the adversary. Additionally, it will often be useful to consider the restriction of a given $q(S)$ to queries on some subset $S' \subset S$, which we denote by $q(S)|_{S'}$. For simplicity and when clear from context, we will write just $q(S')$ as shorthand for $q(S)|_{S'}$. While there may exist many $q(S') \in Q_h(S')$ that are not equal to $q(S)|_{S'}$, we will generally be able to assume without loss of generality that our learners do not re-query anything in S', which ensures the notation is well-defined.

2.4. Inference

Similar to the framework introduced in Kane et al. (2017), we will often wish to analyze what information is *inferred* by a certain query response $q(S) \in Q_h(S)$. Let (X, H) be a hypothesis class with associated query set $Q = \{\mathcal{O}_i\}$. Given a sample $S \subset X$ and query response $q(S) \in Q_h(S)$, denote by $H|_{q(S)}$ the set of hypotheses consistent with $q(S)$. For any oracle $\mathcal{O}_i$ and appropriately sized subset $T \subset X$, we say that $q(S)$ infers $\alpha \in \mathcal{O}_i(h, T)$ if α is a valid query response with respect to every consistent hypothesis:

$$\forall h' \in H|_{q(S)}, \alpha \in \mathcal{O}_i(h', T). \tag{1}$$

It is worth noting that since $\mathcal{O}_i(h', T)$ may contain multiple valid responses, it is possible that $q(S)$ may infer several of them. As in the previous section, it will often be useful to consider this process in batches. In particular, given a query set Q, $q(S) \in Q_h(S)$, and $q(S') \in Q_h(S')$, we may wish to know when $q(S)$ infers that $q(S')$ is a valid response in $Q_h(S')$ for all $h \in H|_{q(S)}$. In such cases, we say $q(S)$ infers $q(S')$ and write:

$$q(S) \to q(S').$$

As in previous works (Kane et al., 2017, 2018; Har-Peled et al., 2020; Hopkins et al., 2020b,a), we pay particular attention to the case where $\mathcal{O}_i = L$ is the labeling oracle (notation we will use throughout). We introduce two important concepts for this special case. Given a sample S and query response $q(S)$, it will be useful to analyze the set of points in X whose labels are inferred by $q(S)$. We denote this set by $I(q(S))$. Similarly, it will be useful to analyze how much of X $I(q(S))$ covers (with respect to the distribution over X), which we call the *coverage* of $q(S)$ and denote by $\text{Cov}(q(S))$. Finally, when dealing with labels we may wish to restrict the scope of our inference for the sake of computational efficiency. In such scenarios, we will define an *inference rule* R, which for each query response $q(S)$ determines some subset $S \subseteq I_R(q(S)) \subseteq I(q(S))$. We let Cov_R denote the coverage with respect the rule R, and $T_{I_R}(n)$ denote the inference time under rule R—that is the worst-case time across $x \in X$, $S \subset X$, and $q(S) \in Q_h(S)$ to determine whether $x \in I_R(q(S))$. Finally, we call the rule R efficiently computable if it is polynomial in n. When R is trivial (i.e. $\forall q(S) : I_R(q(S)) = I(q(S))$), we drop it from all notation.

2.5. Bounded Memory

The main focus of our work lies in understanding not only when active learning can achieve exponential improvement over passive learning, but when this can be done by a learner with limited memory. Previous works studying active learning with enriched queries mostly focus on the pool-based model, where the learner receives a large batch of unlabeled samples rather than access to a

3. Formally, this is the product space of valid responses to each possible query on S.

sampling oracle. In this case, the implicit assumption is that the learner may query any subset of samples from the pool, but this requires the learner to use a large amount of storage.

Adapting definitions from the passive learning literature (Haussler, 1988; Floyd, 1989; Ameur et al., 1996), we define a new, more realistic model for active learning with enriched queries in which the learner may only store some finite number of points from a stream of samples. At any given step, we restrict the learner to querying only points it currently remembers. More formally, we consider learners equipped with two tapes, the *query* and *work* tapes, and two counters, the *sample* and *query* counters. The query tape stores points in the instance space X that the learner has saved and may wish to query. The work tape stores bits which provide any extra information about these points needed for computation—typically this entails query responses, but we will see cases where other types of information are useful as well. The sample and query counters, true to name, track the total number of unlabeled samples drawn and queries made by the algorithm at any given step.

We note that the complexity of the query tape is measured in the number of points stored there at any given time, rather than in the total number of bits required to represent it. This matches early works on bounded-memory learning in the passive regime (Haussler, 1988; Floyd, 1989; Ameur et al., 1996) and is necessary due to the fact that we are interested mainly in working over infinite instance spaces like the reals (where representing even a single point may take infinite bits). It is also worth noting that this avoids the fact that different representations of data may have different bit complexities—given a certain representation of the data (say with finite bit-complexity), it is easy to convert our memory bounds if desired to a model counting only bits.

We now discuss our model in greater depth. Given the query and work tapes, the learner may choose at each step from the following options:

1. Sample a point and add it to the query tape.
2. Remove a point from the query tape.
3. Query any subset of points on the query tape, writing the results on the work tape.
4. Write or remove a bit from the work tape.

Further, as in previous work (Ameur et al., 1996), we allow the action taken by the algorithm at any step to depend on the contents of the query and sample counters. This can be formalized in one of two ways. The first, considered implicitly by Ameur et al. (1996), is to think of the algorithm as governed by a non-uniform transition function that may depend on the entire content of the sample and query counters. We will generally take this view throughout the paper since it is simpler, but if one wishes to use a uniform model of computation, another method is to allow the algorithm to run "simple" randomized procedures that only take about loglog space in the size of the counter. Since in general our algorithms use at most $n = \text{poly}(\varepsilon^{-1}, \log(1/\delta))$ samples, the latter view essentially allocates a special block of $O(\log\log(1/\varepsilon) + \log\log(1/\delta))$ memory[4] to deal with the counter. We note that because we allow this procedure to be randomized, this version of the model requires expected rather than worst-case bounds on query and time complexity.

So far we have only defined RPU-learning with respect to the standard passive model of learning, so we pause briefly to discuss its extension to the above. Namely, a hypothesis class (X, H)

4. In a bit more detail, we can use approximate counting to probabilistically estimate the counter up to a small constant factor with probability at least $1-\delta$ in space $O(\log(\log(n)) + \log(\log(1/\delta)))$ (Nelson and Yu, 2020), which explains the discrepancy in dependence on δ in these two equations.

is RPU-learnable with respect to Q with sample complexity $n(\varepsilon, \delta)$ and query complexity $q(\varepsilon, \delta)$ if there exists an algorithm using at most $n(\varepsilon, \delta)$ calls to the sample oracle and $q(\varepsilon, \delta)$ queries that is probably useful and reliable over the randomness of the sample oracle in the sense of Theorem 1. With this in mind, we say that a hypothesis class is *RPU-learnable with bounded memory* if there exists an RPU-learner for the class whose query and work tapes never exceed some constant $M(X, H)$ length independent of the learning parameters ε and δ. At a finer grain level, we say that such a class is learnable with memory $M(X, H)$. We emphasize that as in Ameur et al. (1996), we do not measure the memory usage of the counter. Indeed, it is not hard to see that some sort of counter or memory scaling with ε and δ is necessary to give a stopping condition for the learner. Finally, we note that while previous techniques such as the inference dimension algorithm of Kane et al. (2017) can certainly be modified to fit into the above framework, they do not result in bounded memory learners, requiring storage that scales with ε and δ in both the query and work tapes.

3. Lossless Sample Compression and Bounded Memory Active Learning

Now that we have covered sufficient background, we are in position to state our main contributions. We do this in two parts. In the first we cover lossless sample compression, a novel sufficient condition for query and computationally efficient bounded memory active RPU-learnability. In the second, we introduce natural enriched queries for three families of classifiers fundamental to machine learning, axis-aligned rectangles, decision trees, and halfspaces in two dimensions, and show that each has a small, efficiently computable lossless commpression scheme. Without enriched queries, all three of these classes have strong lower bounds (Kivinen, 1995; Dasgupta, 2005; Hopkins et al., 2020b) that show not only how they cannot be active learned efficiently, but moreover that they cannot be RPU-learned at all. Thus in each case our proposed queries move the class from completely intractable to highly efficient, even on low memory devices.

3.1. Lossless Sample Compression

We begin with some background and intuition. Standard sample compression schemes (Littlestone and Warmuth, 1986) posit the existence of a compression algorithm A and decompression scheme D such that for any sample S, $A(S)$ is small, and $D(A(S))$ outputs a hypothesis correctly labeling all elements of S. Lossless sample compression strengthens this idea in two ways. First, the output hypothesis must correctly label not just S, but every point whose label can be inferred from queries on S. Second, the hypothesis output by D must be zero-error on the remaining points (but, like our learners, is allowed to abstain).

Definition 2 (Lossless Compression Schemes (Informal)) *We say a hypothesis class (X, H) has a lossless compression scheme (LCS) of size k with respect to a query set Q and inference rule R if for all classifiers $h \in H$, subsets $S \subset X$, and $q(S) \in Q_h(S)$ there exists $W = W(q(S)) \subseteq S$ of size at most k such that:*

$$I_R(q(S)) = I_R(q(S)|_W).$$

Given the existence of a lossless compression scheme for some class (X, H), we prove in Appendix A that (a slight variant of) the following simple algorithm is a computationally efficient, query-optimal, and bounded memory RPU-learner for the class.

It is worth noting that in Algorithm 1, the set of remaining uninferred points X_i need not be kept in memory. Membership in X_i can be checked in an online fashion in step 1.

Algorithm 1: Bounded Memory RPU-Learning via Lossless Compression

Input: Query set Q, class (X, H), sample oracle $\mathcal{O}_X$, and lossless compression scheme W.
Output: Zero-error classifier that labels a $1-\varepsilon$ fraction of X with probability $1-\delta$.
Parameters: LCS size k, Query cap $T_1 = O\left(\log\left(\frac{1}{\varepsilon\delta}\right)\right)$, Sample cap $T_2 = \tilde{O}\left(\frac{k\log^2(1/\delta)}{\varepsilon}\right)$
Algorithm:
Initialize $i = 0$, $j = 0$, $C_0 = \{\varnothing\}$, $X_0 = X$;
While $i \leq T_1$:

1. Sample a subset $S_i \subseteq X_i$ of size $6k$ (via rejection sampling on $\mathcal{O}_X$).
   (a) For each point drawn from X in this process, increment j.
   (b) If j reaches T_2, abort and return labels inferred by queries on C_i
2. Make all queries on $S_i \cup C_i$, and compute $C_{i+1} = W(q(S_i \cup C_i))$
3. Remove all points in (and queries on) $(S_i \cup C_i) \setminus C_{i+1}$ from memory and increment i.
4. Set $X_i \subseteq X$ to be the set of points uninferred by queries on C_i

Return labels inferred by queries on C_i

Theorem 3 (Informal Theorem 15) *Algorithm 1 actively RPU-learns (X, H) in only*

$$q(\varepsilon, \delta) \leq O_k\left(\log(1/\varepsilon)\right)$$

queries,

$$T(\varepsilon, \delta) \leq \tilde{O}_k\left(T_{X,H,W}\frac{\log^2(1/\delta)}{\varepsilon}\right)$$

time, and

$$M(X, H) \leq O_k(1)$$

memory, where we have suppressed dependence on k (the size of the LCS), and $T_{X,H,W}$ is a parameter dependent only on the class and compression scheme W. In all examples we study $T_{X,H,W}$ is small and dependence on k is at worst quadratic.

It is worth noting that $O(\log(1/\varepsilon))$ query complexity is information-theoretically optimal for most non-trivial concept classes. As long as the class has poly$(1/\varepsilon)$ $\Omega(\varepsilon)$-separated concepts, any active learner must make $\Omega(\log(1/\varepsilon))$ queries to distinguish between them (Kulkarni et al., 1993). Indeed all classes we consider satisfy this property, making Algorithm 1 query-optimal in these cases.

Before moving on to discussing examples of classes with lossless compression schemes, we give a high-level overview of the proof of Theorem 3, split into three main components.

Step 1: LCS $\implies$ Passive RPU-Learning. We begin by proving that classes with lossless compression schemes are efficiently RPU-learnable in the *passive* regime. In fact, we prove this is true for a weaker form of the parameter we call a perfect compression scheme (PCS) that requires the compressed set to infer just the labels of the original sample itself (rather than all labels *inferred* by queries on the original sample). The proof is in essence a double sampling argument that follows

along the lines of Littlestone and Warmuth (1986)'s original analysis of sample compression. While one must be slightly more careful in our regime, the idea remains the same: given a PCS of size k, we upper bound the probability over samples S of size m that there exists a set $T \subseteq S$ of size at most k such that $S \in I(q(S)|_T)$, but the coverage of $q(S)|_T$ is bad. To show this, we union bound over the event that this occurs on a given set of up to k indices in $[m]$ (corresponding to points in the sample S). Since we can think of points outside of the fixed set of indices as being drawn independently at random, if the coverage of the fixed indices is bad, the probability that the remaining points all lie in the coverage is small, which gives the desired bound. See Theorem 13 for details.

Step 2: Passive RPU-learning $\implies$ Active RPU-Learning. The second step is to modify the boosting procedure of Kane et al. (2017) (who analyze only finite instance spaces) to show that passive RPU-learnability implies query-optimal active RPU-learnability. The idea is just a simplified version of Algorithm 1. Since the class is passively RPU-learnable, there exists some constant sample size k' which covers half the the distribution with probability at least a half. By repeatedly sampling k' points and removing any part of the instance space inferred by the sample, after $O(\log(1/(\varepsilon\delta)))$ rounds the learner will have coverage at least $1-\varepsilon$ with probability at least $1-\delta$. The only issue is that this process alone may draw an unbounded number of samples, which can be fixed by implementing a sample cut-off as in Algorithm 1. See Theorem 14 for details.

Step 3: Compressing Active RPU-Learning. Finally, the third step is to show how, given a lossless compression scheme, Step 2 can be modified to work in bounded memory. This process is explained in Steps 2 and 3 of Algorithm 1. The idea is to store a compressed representation which is updated in each round by compressing its union with the k' new points drawn in that round. By the guarantees of lossless sample compression, this compressed set remains a constant size while preserving the coverage guarantees of step 2. This gives the desired bounded memory RPU-learner. See Theorem 15 for details.

3.2. Three Classes with Lossless Compression Schemes

We give three fundamental examples of classes which are impossible to RPU-learn with standard label queries, but have small, efficiently computable lossless compression schemes with respect to natural enriched queries: axis-aligned rectangles, decision trees, and halfspaces in two dimensions. As a result, these classes are all efficiently RPU-learnable with bounded memory. Below we introduce each class, our proposed enriched queries, overview the construction of our compression scheme, and state the resulting implications on each classes learnability.

3.2.1. AXIS-ALIGNED RECTANGLES

We start our discussion with the simplest of the three, axis-aligned rectangles in $\mathbb{R}^d$, which correspond to indicator functions for products of intervals over $\mathbb{R}$:

$$R = [a_1, b_1] \times \ldots \times [a_d, b_d]$$

where $a_i \leq b_i$. Thus, axis-aligned rectangles essentially define a certain "acceptable" range for every feature—a point is labeled 1 iff it lies in this range for all coordinates. We introduce a natural enriched query that builds off this intuition we call the "odd-one-out" oracle $\mathcal{O}_{\text{odd}}$. Informally, given a point $x \in \mathbb{R}^d$ lying outside the rectangle, an "odd-one-out" query simply asks the user "why do

you dislike x?". Concretely, one might imagine a chef is trying to cook a dish for a particularly picky patron. After each failed attempt, the chef may ask the patron what went wrong—perhaps the patron thinks the meat was overcooked! More formally, the "odd-one-out" query asks for a violated coordinate (i.e. a feature lying outside the acceptable range), and whether the coordinate was too large (in our example, overcooked) or too small (undercooked). See Appendix B.1 for a formal explanation of the odd-one-out oracle in the query framework described in Section 2.3.

We prove that axis-aligned rectangles have a small, easy to compute lossless compression scheme with respect to the odd-one-out oracle.

Proposition 4 (Informal Proposition 16) *The class of axis-aligned rectangles over $\mathbb{R}^d$ has a lossless compression scheme of size at most $4d$ with respect to $\mathcal{O}_{odd}$.*

We give an informal overview of the construction. Given a sample $S \in \mathbb{R}^d$, consider separately the subset of S inside the rectangle (denoted S^+), and outside the rectangle (denoted S^-). To compress S^+, it is sufficient to store a subset $T \subseteq S^+$ such that the rectangle spanning T contains S^+. This can be done by taking the at most $2d$ points with maximal and minimal values in every coordinate. To compress S^-, notice that if two points $x, x' \in \mathbb{R}^d$ lying outside the rectangle violate the same coordinate in the same manner (i.e. both are too small or too large), it is enough to store the point which is closer to the rectangle in that coordinate. Thus for each of d possible violated coordinates, it is sufficient to store the 'smallest' point (in that coordinate) which the odd-one-out oracles measures as too large, and the 'largest' point it measures as too small. Taken together, these give the desired LCS of size $4d$.

Applying Theorem 3, we get a computationally efficient and query-optimal, bounded memory RPU-learning algorithm for axis-aligned rectangles.

Corollary 5 (Informal Corollary 17) *The class of axis-aligned rectangles in $\mathbb{R}^d$ is RPU-learnable in only*

$$q(\varepsilon, \delta) = O\left(d \log\left(\frac{1}{\varepsilon\delta}\right)\right)$$

queries, $O(d)$ memory, and time $\tilde{O}\left(\frac{d^2 \log^2(1/\delta)}{\varepsilon}\right)$ when the learner has access to $\mathcal{O}_{odd}$.

3.2.2. DECISION TREES

While rectangles provide an excellent theoretical example of a class for which basic enriched queries break standard barriers in active and bounded memory learning, they are often too simple to be of much practical use. To this end, we next consider a broad generalization of the class of axis-aligned rectangles called decision trees. A decision tree over $\mathbb{R}^d$ is a binary tree where each node in the tree corresponds to an inequality:

$$x_i \overset{?}{\geq} b \text{ or } x_i \overset{?}{\leq} b,$$

measuring the i-th feature (coordinate) of any $x \in \mathbb{R}^d$. Each leaf in the tree is assigned a label, and the label of any $x \in X$ is uniquely determined by the leaf resulting from following the decision tree from root to leaf, always taking the path determined by the inequality at each node. Informally, a decision tree may then be thought of as a partition of $\mathbb{R}^d$ into clusters (axis-aligned rectangle) given by the leaves. Our proposed enriched query for decision trees, the "same-leaf" oracle $\mathcal{O}_{\text{leaf}}$, builds off this intuition. Given a decision tree T and two points $x, x' \in \mathbb{R}^d$, $\mathcal{O}_{\text{leaf}}(T, x, x')$ determines

whether x and x' lie in the same leaf of the decision tree. Thinking of each leaf as a cluster, this query may be seen as a variant of the "same-cluster" query paradigm studied in many recent works (Ashtiani et al., 2016; Verroios et al., 2017; Mazumdar and Saha, 2017; Ailon et al., 2018; Firmani et al., 2018; Dasgupta et al., 2018; Saha and Subramanian, 2019; Bressan et al., 2020). For our scenario, think of asking a user whether two movies they like are of the same genre. We prove that the class of decision trees of size s (at most s leaves) has a small, efficiently computable lossless compression scheme. In this case, however, we need to restrict our inference via a basic inference rule we call R_{rect} which in essence enforces that inference is done independently across each leaf (see Appendix B for details).

Proposition 6 (Informal Proposition 18) *The class of size s decision trees over $\mathbb{R}^d$ has an LCS of size at most $2ds$ with respect to $\mathcal{O}_{leaf}$ and R_{rect}.*

Our construction follows similarly to the positive case of Proposition 4. Each leaf of the decision tree is an axis-aligned rectangle, so given a sample $S \subset \mathbb{R}^d$, we pick a set T of size at most $2d$ in each leaf such that the rectangle spanning T contains all points in S restricted to that leaf. Since we restrict inference to being done independently across leaves, taking the union of these sets across all s leaves gives the desired LCS. As a corollary, we get efficient, bounded-memory RPU-learnability.

Corollary 7 (Informal Corollary 19) *The class of size s decision trees over $\mathbb{R}^d$ is RPU-learnable in only*

$$q(\varepsilon, \delta) = O\left(ds^2 \log\left(\frac{1}{\varepsilon\delta}\right)\right)$$

queries, $O(ds)$ memory, and time $\tilde{O}\left(\frac{d^2 s^2 \log(1/\delta)^2}{\varepsilon}\right)$ when the learner has access to $\mathcal{O}_{leaf}$.

It is worth noting that while the class of decision trees of arbitrary size is not learnable (even in the PAC-setting (Hancock et al., 1996)), we can bootstrap Theorem 3 and Corollary 7 to build an algorithm that learns decision trees *attribute-efficiently*. That is to say an algorithm whose *expected* time, number of queries, and memory scales with the size of the unknown decision tree.

Corollary 8 (Informal Corollary 20) *There exists an algorithm for RPU-learning decision trees over $\mathbb{R}^d$ which in expectation:*

1. *Makes $O\left(ds^2 \log\left(\frac{s}{\varepsilon\delta}\right)\right)$ queries*
2. *Runs in time poly$(s, d, \varepsilon^{-1}, \log(\delta^{-1}))$*
3. *Uses $O(ds)$ memory,*

where s is the size of the underlying decision tree.

We cover the algorithm and analysis of Corollary 8 in more detail in Appendix B, but the idea is simply to apply Corollary 7 as a blackbox to increasingly large guesses for the size of the underlying decision tree. After each iteration we check the coverage by drawing a large unlabeled sample. If the coverage is poor, we double our guess and repeat the process.

3.2.3. HALFSPACES

Despite being vastly more expressive than axis-aligned rectangles, decision trees in $\mathbb{R}^d$ are still simplistic in the sense that they remain axis-aligned. For our final example, we study a fundamental class without any such restriction: (non-homogeneous) halfspaces in two dimensions. Recall that a halfspace in two dimensions is given by the sign of $h = \langle v, \cdot \rangle + b$ for some $v \in \mathbb{R}^2$ and $b \in \mathbb{R}$. Following a number of prior works (Jamieson and Nowak, 2011; Karbasi et al., 2012; Wauthier et al., 2012; Xu et al., 2017; Hopkins et al., 2020a,b; Cui and Sato, 2020), we study the learnability of halfspaces with comparison queries. Informally, given two points $x, y \in \mathbb{R}^2$, a comparison query simply asks which is further away from the separating hyperplane, or more formally:

$$h(x) \overset{?}{\geq} h(y).$$

This type of query is natural in scenarios like halfspaces where the class has an underlying ranking. One might ask a doctor, for instance, "which patient is sicker?". We show halfspaces in two dimensions with comparisons have a small, efficiently computable lossless compression scheme.

Proposition 9 (Informal Proposition B.3) *The class of halfspaces over $\mathbb{R}^2$ has a size $O(1)$ lossless compression scheme with respect to the comparison oracle.*

Our LCS construction for halfspaces is a bit more involved than previous examples. To simplify the construction we prove a slightly weaker form of compression which is lossless only on monochromatic (same-label) subsets. We prove in Appendix A that this weakening recovers the same guarantees of Theorem 3 by a slight variant of Algorithm 1. With this in mind, consider a sample S of positive points. The crucial idea behind our compression scheme is that a comparison query on a pair $x, y \in \mathbb{R}^2$ determines a non-decreasing line with respect to the underlying hyperplane (i.e. $t(x - y)$ if $h(x) \geq h(y)$). Combined with any point in $s \in S$ (for which we know $h(s) \geq 0$), $s + t(x - y)$ gives a ray of positive points (for $t \geq 0$). Moreover, two such rays with the same base point form a cone of positive points. Our compression set is formed by choosing up to 5 points (base point plus two non-decreasing lines) that define the largest cone and thus encompass the rest. Figure 1 provides a pictorial description of finding this cone. See Proposition B.3 for details.

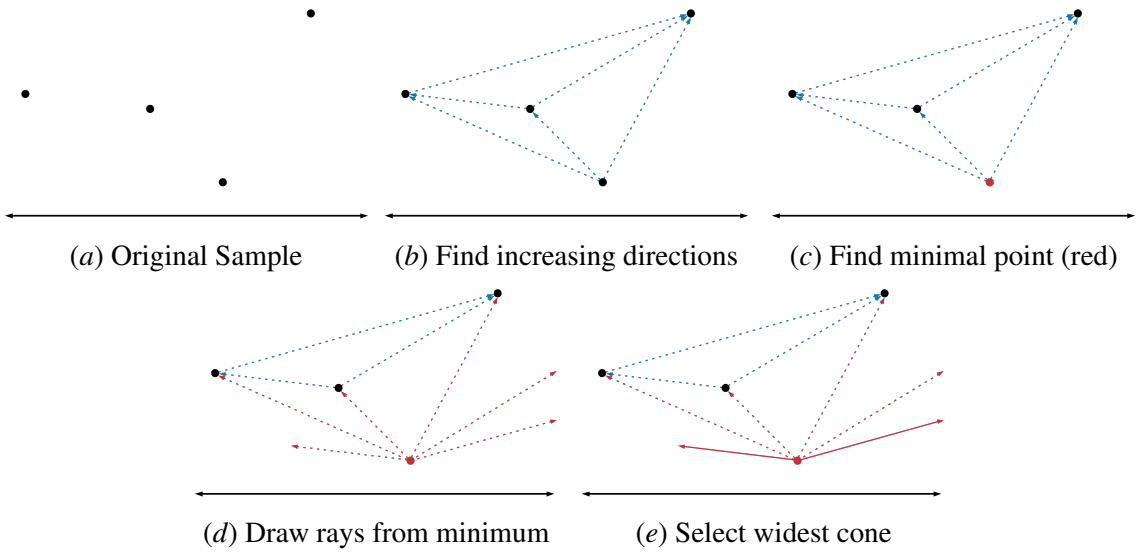
As a corollary, we get efficient, bounded memory RPU-learning for halfspaces in two-dimensions.

Corollary 10 (Informal Corollary 22) *The class of halfspaces over $\mathbb{R}^2$ is actively RPU-learnable in*

$$q(\varepsilon, \delta) \leq O\left(\log\left(\frac{1}{\varepsilon\delta}\right)\right)$$

queries, $O(1)$ memory, and time $\tilde{O}\left(\frac{\log^2(1/(\delta))}{\varepsilon}\right)$.

Together, Corollary 5, Corollary 7, and Corollary 10 make a strong argument for the practicality of lossless sample compression and the enriched query model. Since compression tends to be efficient (and is in all examples we consider), its analysis results in simple, computationally efficient, query-optimal algorithms for fundamental classes which may otherwise be theoretically impossible or computationally infeasible to learn. The fact that the resulting algorithms have the additional benefit of running in bounded memory is not only of theoretical interest, but also breaks down barriers to the use of active learning in practice, opening the door to everyday applications on mobile or other devices with limited storage.



(*a*) Original Sample (*b*) Find increasing directions (*c*) Find minimal point (red)

(*d*) Draw rays from minimum (*e*) Select widest cone

Figure 1: Pictorial intuition behind our LCS for halfspaces. Given a (positive) monochromatic sample (diagram (a)), we find a minimal point (diagram (c)) and all directions of increase (diagram (b)). Combining these gives a set of positive rays which form cones (diagram (d)). Our LCS is given by the points which contribute to the widest cone, depicted in diagram (e) (here all 4 points).

4. Related Work

4.1. Bounded Memory Learning

Bounded memory learning in the sense we consider was first introduced in Haussler (1988), who showed the existence of passive PAC-learners for restricted classes of decision trees and basic functions on $\mathbb{R}$ such as finite unions of intervals with memory independent of the accuracy parameters ε and δ. Floyd (1989), and later Ameur et al. (1996) extended Haussler's work to a general theory of bounded memory passive learning including features we consider such as a sample counter. The latter in particular give necessary and sufficient conditions based upon a compression scheme stronger than standard sample compression, but weak enough to cover important classifiers such as halfspaces. In the decades since, many variants and relaxations of bounded memory learning (e.g. memory scaling with ε, δ, learning a finite stream, storing only bits, time-space tradeoffs, etc.) have seen a substantial amount of study (Shamir, 2014; Steinhardt et al., 2016; Raz, 2017; Moshkovitz and Moshkovitz, 2017; Moshkovitz and Tishby, 2017; Moshkovitz and Moshkovitz, 2018; Beame et al., 2018; Raz, 2018; Garg et al., 2018; Sharan et al., 2019; Dagan et al., 2019; Gonen et al., 2020; Assadi and Raz, 2020). To our knowledge, however, the subject has seen little to no work within the active learning literature, though a few do consider query settings beyond just labels including statistical (Steinhardt et al., 2016; Gonen et al., 2020) and equivalence queries (Ameur, 1995).

4.2. Active Learning with Enriched Queries

Active learning with enriched queries has become an increasingly popular alternative to the standard model in cases like halfspaces where strong lower bounds prevent adaptivity from providing a

significant advantage over standard passive learning. While most prior works in the area consider specific examples of enriched queries such as comparisons (Jamieson and Nowak, 2011; Karbasi et al., 2012; Wauthier et al., 2012; Xu et al., 2017; Hopkins et al., 2020a,b; Cui and Sato, 2020), cluster-queries (Ashtiani et al., 2016; Vikram and Dasgupta, 2016), mistake queries (Balcan and Hanneke, 2012), and separation queries (Har-Peled et al., 2020), our work is more closely related to the general paradigm for enriched query active learning introduced by Kane et al. (2017). In their work, Kane et al. (2017) introduce *inference dimension*, a combinatorial parameter that exactly characterizes when a concept class is actively learnable in $O(\log(1/\varepsilon))$ rather than $O(1/\varepsilon)$ queries. Lossless sample compression can be seen as a strengthening of inference dimension (albeit extended to a richer regime of queries than that considered in Kane et al. (2017)) which implies both $O(\log(1/\varepsilon))$ query complexity and bounded memory. In fact, Kane et al. (2017) mention in their work that halfspaces in two dimensions should be learnable in bounded memory (though they give no indication of how this might be done), and ask whether every class with finite inference dimension has such a property. Our work marks the first non-trivial progress towards resolving this open problem.

Acknowledgments

Work done when MH was supported by NSF Award DGE-1650112, DK was supported by NSF Award CCF-1553288 (CAREER) and a Sloan Research Fellowship, and SL was supported by NSF Award CCF-1909634.

References

Nir Ailon, Anup Bhattacharya, and Ragesh Jaiswal. Approximate correlation clustering using same-cluster queries. In *Latin American Symposium on Theoretical Informatics*, pages 14–27. Springer, 2018.

Foued Ameur. A space-bounded learning algorithm for axis-parallel rectangles. In *European Conference on Computational Learning Theory*, pages 313–321. Springer, 1995.

Foued Ameur, Paul Fischer, Klaus-Uwe Höffgen, and Friedhelm Meyer auf der Heide. Trial and error: A new approach to space-bounded learning. *Acta Inf.*, 33:621–630, 10 1996. doi: 10.1007/s002360050062.

Hassan Ashtiani, Shrinu Kushagra, and Shai Ben-David. Clustering with same-cluster queries. *arXiv preprint arXiv:1606.02404*, 2016.

Sepehr Assadi and Ran Raz. Near-quadratic lower bounds for two-pass graph streaming algorithms. *arXiv preprint arXiv:2009.01161*, 2020.

Maria Florina Balcan and Steve Hanneke. Robust interactive learning. In *Conference on Learning Theory*, pages 20–1, 2012.

Paul Beame, Shayan Oveis Gharan, and Xin Yang. Time-space tradeoffs for learning finite functions from random evaluations, with applications to polynomials. In *Conference On Learning Theory*, pages 843–856, 2018.

Guy Blanc, Neha Gupta, Jane Lange, and Li-Yang Tan. Universal guarantees for decision tree induction via a higher-order splitting criterion. *arXiv preprint arXiv:2010.08633*, 2020.

Marco Bressan, Nicolò Cesa-Bianchi, Silvio Lattanzi, and Andrea Paudice. Exact recovery of mangled clusters with same-cluster queries. *arXiv preprint arXiv:2006.04675*, 2020.

Zhenghang Cui and Issei Sato. Active classification with uncertainty comparison queries. *arXiv preprint arXiv:2008.00645*, 2020.

Yuval Dagan, Gil Kur, and Ohad Shamir. Space lower bounds for linear prediction in the streaming model. *arXiv preprint arXiv:1902.03498*, 2019.

Sanjoy Dasgupta. Analysis of a greedy active learning strategy. In *Advances in neural information processing systems*, pages 337–344, 2005.

Sanjoy Dasgupta. Two faces of active learning. *Theoretical computer science*, 412(19):1767–1781, 2011.

Sanjoy Dasgupta, Akansha Dey, Nicholas Roberts, and Sivan Sabato. Learning from discriminative feature feedback. *Advances in Neural Information Processing Systems*, 31:3955–3963, 2018.

Andrzej Ehrenfeucht and David Haussler. Learning decision trees from random examples. *Information and Computation*, 82(3):231–246, 1989.

Ran El-Yaniv and Yair Wiener. Active learning via perfect selective classification. *Journal of Machine Learning Research*, 13(Feb):255–279, 2012.

Donatella Firmani, Sainyam Galhotra, Barna Saha, and Divesh Srivastava. Robust entity resolution using a crowdoracle. *IEEE Data Eng. Bull.*, 41(2):91–103, 2018.

Sally Floyd. Space-bounded learning and the vapnik-chervonenkis dimension. In *Proceedings of the second annual workshop on Computational learning theory*, pages 349–364, 1989.

Sally Floyd and Manfred Warmuth. Sample compression, learnability, and the vapnik-chervonenkis dimension. *Machine learning*, 21(3):269–304, 1995.

Sumegha Garg, Ran Raz, and Avishay Tal. Extractor-based time-space lower bounds for learning. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 990–1002, 2018.

Alon Gonen, Shachar Lovett, and Michal Moshkovitz. Towards a combinatorial characterization of bounded memory learning. *arXiv preprint arXiv:2002.03123*, 2020.

Thomas Hancock, Tao Jiang, Ming Li, and John Tromp. Lower bounds on learning decision lists and trees. *Information and Computation*, 126(2):114–122, 1996.

Steve Hanneke. The optimal sample complexity of pac learning. *The Journal of Machine Learning Research*, 17(1):1319–1333, 2016.

Steve Hanneke et al. Theory of disagreement-based active learning. *Foundations and Trends® in Machine Learning*, 7(2-3):131–309, 2014.

Sariel Har-Peled, Mitchell Jones, and S. Rahul. Active learning a convex body in low dimensions. In *ICALP*, 2020.

D. Haussler. Space efficient learning algorithms. Technical Report UCSC-CRL-88-2, University of Calif. Computer Research Laboratory, Santa Cruz, CA, 1988.

Max Hopkins, Daniel Kane, and Shachar Lovett. The power of comparisons for actively learning linear classifiers. *Advances in Neural Information Processing Systems*, 33, 2020a.

Max Hopkins, Daniel Kane, Shachar Lovett, and Gaurav Mahajan. Noise-tolerant, reliable active classification with comparison queries. In *Conference on Learning Theory*, pages 1957–2006. PMLR, 2020b.

Badr Hssina, Abdelkarim Merbouha, Hanane Ezzikouri, and Mohammed Erritali. A comparative study of decision tree id3 and c4. 5. *International Journal of Advanced Computer Science and Applications*, 4(2):13–19, 2014.

Kevin G Jamieson and Robert Nowak. Active ranking using pairwise comparisons. In *Advances in neural information processing systems*, pages 2240–2248, 2011.

Adam Tauman Kalai and Shang-Hua Teng. Decision trees are pac-learnable from most product distributions: a smoothed analysis. *arXiv preprint arXiv:0812.0933*, 2008.

Daniel Kane, Shachar Lovett, and Shay Moran. Generalized comparison trees for point-location problems. In *International Colloquium on Automata, Languages and Programming*, 2018.

Daniel M Kane, Shachar Lovett, Shay Moran, and Jiapeng Zhang. Active classification with comparison queries. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 355–366. IEEE, 2017.

Amin Karbasi, Stratis Ioannidis, et al. Comparison-based learning with rank nets. *arXiv preprint arXiv:1206.4674*, 2012.

J. Kivinen. Reliable and useful learning with uniform probability distributions. In *Proceedings of the First International Workshop on Algorithmic Learning Theory*, 1990.

J. Kivinen. Learning reliably and with one-sided error. *Mathematical Systems Theory*, 28(2):141–172, 1995.

Sanjeev R Kulkarni, Sanjoy K Mitter, and John N Tsitsiklis. Active learning using arbitrary binary valued queries. *Machine Learning*, 11(1):23–35, 1993.

Eyal Kushilevitz and Yishay Mansour. Learning decision trees using the fourier spectrum. *SIAM Journal on Computing*, 22(6):1331–1348, 1993.

Nick Littlestone and Manfred Warmuth. Relating data compression and learnability. 1986.

Arya Mazumdar and Barna Saha. Clustering with noisy queries. In *Advances in Neural Information Processing Systems*, pages 5788–5799, 2017.

Dana Moshkovitz and Michal Moshkovitz. Mixing implies lower bounds for space bounded learning. In *Conference on Learning Theory*, pages 1516–1566, 2017.

Dana Moshkovitz and Michal Moshkovitz. Entropy samplers and strong generic lower bounds for space bounded learning. In *9th Innovations in Theoretical Computer Science Conference (ITCS 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

Michal Moshkovitz and Naftali Tishby. A general memory-bounded learning algorithm. *arXiv preprint arXiv:1712.03524*, 2017.

Jelani Nelson and Huacheng Yu. Optimal bounds for approximate counting. *arXiv preprint arXiv:2010.02116*, 2020.

Ryan O'Donnell and Rocco A Servedio. Learning monotone decision trees in polynomial time. *SIAM Journal on Computing*, 37(3):827–844, 2007.

J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

Ran Raz. A time-space lower bound for a large class of learning problems. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 732–742. IEEE, 2017.

Ran Raz. Fast learning requires good memory: A time-space lower bound for parity learning. *Journal of the ACM (JACM)*, 66(1):1–18, 2018.

Ronald L Rivest and Robert H Sloan. Learning complicated concepts reliably and usefully. In *AAAI*, pages 635–640, 1988.

Barna Saha and Sanjay Subramanian. Correlation clustering with same-cluster queries bounded by optimal cost. *arXiv preprint arXiv:1908.04976*, 2019.

Burr Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009.

Ohad Shamir. Fundamental limits of online and distributed algorithms for statistical learning and estimation. *Advances in Neural Information Processing Systems*, 27:163–171, 2014.

Vatsal Sharan, Aaron Sidford, and Gregory Valiant. Memory-sample tradeoffs for linear regression with small error. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 890–901, 2019.

Sonia Singh and Priyanka Gupta. Comparative study id3, cart and c4. 5 decision tree algorithm: a survey. *International Journal of Advanced Information Science and Technology (IJAIST)*, 27 (27):97–103, 2014.

Jacob Steinhardt, Gregory Valiant, and Stefan Wager. Memory, communication, and statistical queries. In *Conference on Learning Theory*, pages 1490–1516, 2016.

M Umanol, Hirotaka Okamoto, Itsuo Hatono, Hiroyuki Tamura, Fumio Kawachi, Sukehisa Umedzu, and Junichi Kinoshita. Fuzzy decision trees by fuzzy id3 algorithm and its application to diagnosis systems. In *Proceedings of 1994 IEEE 3rd International Fuzzy Systems Conference*, pages 2113–2118. IEEE, 1994.

Leslie G Valiant. A theory of the learnable. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 436–445. ACM, 1984.

Vladimir Vapnik and Alexey Chervonenkis. Theory of pattern recognition, 1974.

Vasilis Verroios, Hector Garcia-Molina, and Yannis Papakonstantinou. Waldo: An adaptive human interface for crowd entity resolution. In *Proceedings of the 2017 ACM International Conference on Management of Data*, pages 1133–1148, 2017.

Sharad Vikram and Sanjoy Dasgupta. Interactive bayesian hierarchical clustering. In *International Conference on Machine Learning*, pages 2081–2090, 2016.

Fabian L Wauthier, Nebojsa Jojic, and Michael I Jordan. Active spectral clustering via iterative uncertainty reduction. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1339–1347, 2012.

Yichong Xu, Hongyang Zhang, Kyle Miller, Aarti Singh, and Artur Dubrawski. Noise-tolerant interactive learning using pairwise comparisons. In *Advances in Neural Information Processing Systems*, pages 2431–2440, 2017.

Appendix A. Lossless Sample Compression

In this section, we prove that lossless sample compression is sufficient for query-optimal, bounded memory RPU-learning. While we introduced the concept of lossless compression in Section 1, we now state the full formal definition which weakens the parameter to hold only over monochromatic subsets.[5]

Definition 11 (Lossless Compression Schemes) *We say a hypothesis class (X, H) has a lossless compression scheme (LCS) of size k with respect to a query set Q and inference rule R if for all classifiers $h \in H$, monochromatic subsets $S \subset X$, and any $q(S) \in Q_h(S)$ there exists $W = W(q(S)) \subseteq S$ of size at most k such that:*

$$I_R(q(S)) = I_R(q(S)|_W).$$

Let $T_C(n)$ denote the worst-case time required to determine such a subset. We call the LCS efficiently computable if $T_C(n)$ it is polynomial in n. If no inference rule R is stated, it is assumed to be the trivial rule.

It is worth briefly noting the intuition behind requiring compression only for monochromatic sets. The general idea is that in RPU-learning it is sufficient to be able to learn X restricted to the set of positive and negative points. While this strategy fails in the weaker PAC-model (the learner could output the all 1's function for positive points for instance), the fact that RPU-learners cannot make mistakes circumvents this issue.

We will begin by proving that lossless compression implies sample-efficient passive RPU-learning, and then show how this can be leveraged to give query-efficient and bounded-memory active learning. In fact, if all one is interested in is the former, lossless sample compression is needlessly strong. As a result, we start by analyzing a strictly weaker variant that lies in between standard and lossless sample compression, and bears close similarities to Kane et al. (2017)'s theory of inference dimension.

Definition 12 (Perfect Compression Schemes (PCS)) *We say a hypothesis class (X, H) with corresponding query set Q has a perfect compression scheme (PCS) of size k with respect to inference rule R if for all subsets $S \subset X$, $h \in H$, and any $q(S) \in Q_h(S)$ there exists $T = T(Q(S)) \subseteq S$ of size at most k such that:*

$$S \subseteq I_R(q(S)|_T).$$

Thus in a perfect compression scheme, one need only recover the labels of the original sample, while lossless compression requires that any labels *inferred* by queries on the original sample are also preserved. Rather than directly analyzing the effect of PCS on active learning, we first prove as an intermediate theorem that such schemes are sufficient for near-optimal passive RPU-learnability. Combining this fact with the basic boosting procedure of Kane et al. (2017) gives query-optimal (but not bounded memory) active RPU-learnability. The argument for the passive case closely follows the seminal work of (Floyd and Warmuth, 1995, Theorem 5) on learning via sample compression.

5. A subset S is monochromatic with respect to a classifier h if it consists entirely of one label.

Theorem 13 *Let (X, H) have a perfect compression scheme of size k with respect to inference rule R. Then the sample complexity of passively RPU-learning (X, H) is at most*

$$n(\varepsilon, \delta) \leq O\left(\frac{k \log(1/\varepsilon) + \log(1/\delta)}{\varepsilon}\right).$$

Proof Let $h \in H$ be an arbitrary classifier. Our goal is to upper bound by δ the probability that for a random sample S, $|S| = m$, there exists $q(S) \in Q_h(S)$ such that $\text{Cov}_R(q(S)) \leq 1 - \varepsilon$. Notice that it is equivalent to prove that for some fixed worst-case choice of $q(S)$ (minimizing coverage across each sample):

$$\Pr_S[\text{Cov}_R(q(S)) \leq 1 - \varepsilon] \leq \delta.$$

Given this formulation, the argument proceeds along the lines of (Floyd and Warmuth, 1995, Theorem 5). It is enough to upper bound by δ the probability across samples S of size m that there exists $T \subset S$ of size at most k such that:

1. $S \subseteq I_R(q(S)|_T)$
2. $\text{Cov}_R(q(S)|_T) < 1 - \varepsilon$.

Since the PCS implies that every set S has a subset T satisfying the first condition, if both hold only with some small probability then the following basic algorithm A suffices to RPU-learn with sample complexity m: on input $x \in X$, $A(S)(x)$ checks whether all $h \in H$ consistent with $q(S)$ satisfy $L(h, x) = z$ for some $z \in \{0, 1\}$.[6] If this property holds, the algorithm outputs z. Otherwise, the algorithm outputs "$\perp$." Since with probability at least $1 - \delta$ the coverage of $q(S)$ is at least $1 - \varepsilon$, this algorithm will label at least a $1 - \varepsilon$ fraction of the space while never making a mistake.

Proving this statement essentially boils down to a double sampling argument. The idea is to union bound over sets of indices in $[m]$ of size up to k, noting that in each case the remaining $m - k$ points can be treated independently. In greater detail, for $I \subset [m]$, denote by B_I the set of samples S which are inferred by the subsample with indices given by I, that is:

$$B_I = \{S = \{s_1, \ldots, s_m\} : S \subseteq I_R(q(S)|_{\{s_i\}_{i \in I}})\}.$$

On the other hand, let U_I denote samples where the coverage of the subsample given by I is worse than $1 - \varepsilon$:

$$U_I = \left\{S = \{s_1, \ldots, s_m\} : \text{Cov}_R\left(q(S)|_{\{s_i\}_{i \in I}}\right) < 1 - \varepsilon\right\}.$$

Notice that the intersection of B_I and U_I is exactly what we are trying to avoid. By a union bound, the probability that we draw a sample such that there exists a subset T satisfying 1 and 2 is then at most:

$$\sum_{I \subset [m], |I| \leq k} \Pr_S[S \in B_I \cap U_I].$$

It is left to bound the probability for fixed $I \subset [m]$ of drawing a sample in $B_I \cap U_I$. Since we are sampling i.i.d, we can think of independently sampling the coordinates in and outside of I. If the samples given by I have coverage at least $1 - \varepsilon$, we are done. Otherwise, the probability that we

6. Note that this process is class-dependent.

draw remaining samples that are in the coverage is at most $(1-\varepsilon)^{m-|I|}$. Thus we get that the event is bounded by:

$$(1-\varepsilon)^{m-k}\sum_{i=1}^{k}\binom{m}{i},$$

which is at most δ for $m = O\left(\frac{k\log(1/\varepsilon)+\log(1/\delta)}{\varepsilon}\right)$ ■

Theorem 13 implies that hypothesis classes with perfect compression schemes may be passively learned with nearly equivalent asymptotic sample complexity to the much weaker PAC-model (off only by a $\log(1/\varepsilon)$ factor recently removed in Hanneke (2016) in the latter case). Further, since perfect sample compression is preserved over subsets of the instance space (a PCS for X is also a PCS when restricted to $S \subset X$), a bound on passive RPU learning immediately implies efficient active learning via a modification to the basic boosting strategy for finite instance spaces of (Kane et al., 2017, Theorem 3.2).

Theorem 14 (Active Learning) *Let (X, H) have a perfect compression scheme of size k with respect to query set Q and inference rule R. Then (X, H) is actively RPU-learnable in only*

$$q(\varepsilon,\delta) \leq O\left(b(6k)\log\left(\frac{1}{\varepsilon\delta}\right)\right)$$

queries, and

$$T(\varepsilon,\delta) \leq O\left(\left(t(6k)+\frac{kT_{I_R}(60k\log(1/(\varepsilon\delta)))\log(k/(\varepsilon\delta))}{\varepsilon}\right)\log\left(\frac{1}{\varepsilon\delta}\right)\right)$$

time where $b(n)$ is the worst-case number of queries needed to infer some valid $q(S) \in Q_h(S)$ across all $h \in H$ and $|S| = n$, and $t(n)$ is the worst-case time.

Proof For notational convenience, let X' be a copy of X we use to track un-inferred points throughout our algorithm. Consider the following strategy:

1. Sample S of size $6k$ from X' (note this may require taking many samples from X in later rounds)
2. Infer some $q(S) \in Q_h(S)$ via queries on S
3. Restrict X' to points whose labels are not inferred by $q(S)$, (i.e. remove any $x \in X'$ s.t. $q(S) \to L(h, x)$)[7]
4. Repeat $O\left(\log\left(\frac{1}{\varepsilon\delta}\right)\right)$ times, or until $n = O\left(\frac{k\log(k/(\varepsilon\delta))\log(1/(\varepsilon\delta))}{\varepsilon}\right)$ total points have been drawn from X.

Notice that for either of these stopping conditions, one of two statements must hold:

1. The algorithm has performed at least $O\left(\log\left(\frac{1}{\varepsilon\delta}\right)\right)$ rounds.

7. In reality this is done by rejection sampling in step 1. When a point is drawn, we check whether it is inferred by queries on any previous sample.

2. The algorithm has drawn $O\left(\frac{\log\left(\frac{n}{\delta}\right)}{\varepsilon}\right)$ inferred samples in a row.

We argue both of these conditions imply that the coverage of the learner is at least $1-\varepsilon$ with probability at least $1-\delta$. For the first condition, notice that Theorem 13 implies the coverage of $q(S)$ on a random sample of size $6k$ is at least $1/2$ with probability at least $1/2$. Call a round 'good' if it has coverage at least $1/2$. It is sufficient to prove we have at least $\log(1/\varepsilon)$ good rounds with probability at least $1-\delta$. Since each round can be thought of as an independent process, this follows easily from a Chernoff bound. For the latter condition, notice that the probability $O\left(\frac{\log\left(\frac{n}{\delta}\right)}{\varepsilon}\right)$ inferred samples appear in a row (at some fixed point) when the coverage is less than $1-\varepsilon$ is at most $O(\delta/n)$. Union bounding over samples implies the algorithm has the desired coverage guarantees.

Finally, we compute the query and computational complexity. The former follows immediately from noting that we make at most $b(6k)$ queries in each round, and run at most $O\left(\log\left(\frac{1}{\varepsilon\delta}\right)\right)$ rounds. The latter bound stems from noting that each round takes at most $t(6k)$ time to compute queries, and each *sample* requires at most $T_{I_R}(ck\log(1/\varepsilon\delta))$ time for inference for some $60 > c > 0$. ■

A similar result may also be proved by defining a suitable extension of inference dimension to our generalized queries and noting that a perfect compression scheme implies finite inference dimension. It is also worth noting that in most cases of interest the above algorithm will be computationally efficient, as the cost of compression, querying, rejection sampling, and inference tend to be fairly small (and are in all examples we consider). Finally, we show how Theorem 14 can be modified with the addition of an efficiently computable lossless compression scheme to give query-optimal, computationally efficient, and bounded memory RPU-learning.

Theorem 15 (Bounded Memory Active Learning) *Let (X, H) have an LCS of size k with respect to query set Q and inference rule R, and let $B(n)$ denote the maximum number of bits required to express any $q(S) \in Q_h(S)$ for $S \subset X$ of size n and $h \in H$. Then (X, H) is actively RPU-learnable in only:*

$$q(\varepsilon,\delta) \leq O\left(b(6k)\log\left(\frac{1}{\varepsilon\delta}\right)\right)$$

queries,

$$M(X,H) \leq O(B(7k))$$

memory, and

$$T(\varepsilon,\delta) \leq O\left(\left(t(7k) + T_C(7k) + \frac{kT_{I_R}(k)\log(k/(\varepsilon\delta))}{\varepsilon}\right)\log\left(\frac{1}{\varepsilon\delta}\right)\right)$$

time.

Proof We follow the overall strategy laid out in Theorem 14 with two main modifications. First, instead of storing subsamples and queries from all previous rounds to use for rejection sampling, each sub-sample will be merged in between rounds using the guarantees of lossless sample compression. Without this change, the strategy of Theorem 14 would require $O\left(B(6k)\log\left(\frac{1}{\varepsilon\delta}\right)\right)$ memory from the buildup across rounds. Second, we will learn the set of positive and negative points separately since the LCS only guarantees compression for monochromatic subsamples.

More formally, assume for the moment that we draw samples from the distribution restricted to positive (or negative) labels. Following the strategy of Theorem 14, let the sample of size $6k$ drawn at step i be denoted by S_i, the points stored in the query-tape at the start of step i by C_{i-1}, and their corresponding query response $q(C_{i-1})$. The existence of an LCS of size k implies for any $h \in H$ and $q(S_i \cup C_{i-1}) \in Q_h(S_i \cup C_{i-1})$, there exists a subset $C_i \subset S_i \cup C_{i-1}$ of size at most k such that:

$$I_R(q(C_i)) = I_R(q(S_i \cup C_{i-1})),$$

where we recall $q(C_i)$ is shorthand for $q(S_i \cup C_{i-1})|_{C_i})$. Further, since restrictions have strictly less information (that is for all $S, S' \subseteq X$ and $q(S \cup S') \in Q_h(S \cup S')$, $I_R(q(S)) \cup I_R(q(S')) \subseteq I_R(q(S \cup S'))$, we may write:

$$I_R(q(S_i)) \cup I_R(q(C_{i-1})) \subseteq I_R(q(S_i \cup C_{i-1})) = I_R(q(C_i)).$$

By induction on the step i, $q(C_{i-1})$ infers the label of every point in $I(q(S_j))$ for $j < i$, and therefore:

$$\bigcup_{j=1}^{i} I_R(q(S_j)) \subseteq I_R(q(C_i)).$$

Since the coverage guarantees of Theorem 14 at a given step i rely only on this left-hand union, following the same strategy (plus compression) still results in at least $1-\varepsilon$ coverage with probability at least $1-\delta$ in only $O(\log(\frac{1}{\varepsilon\delta}))$ rounds. Further, since we merge our stored information every round into C_i and $q(C_i)$, we never exceed storage of $7k$ points and $O(B(7k))$ bits. The analysis of query and computational complexity follow the same as in Theorem 14 with the addition of compression time.

Finally, we argue that we may learn the general distribution by separately applying the above to the set of positive and negative samples. Namely, at step i we sample from the remaining uninferred points until we receive either $6k$ positive or $6k$ negative samples, and apply the standard algorithm on whichever is reached first. Assume at step i that the measure of remaining positive points is at least that of the remaining negative (the opposite case will follow similarly). Then the probability that we draw $6k$ positive points before $6k$ negative points is at least $1/2$. Recall that such a positive sample has coverage at least $1/2$ over the remaining positive points with probability at least $1/2$. Moreover, since the positive samples make up at least $1/2$ of the space, the coverage of each round over the *entire* distribution is at least $1/4$ with probability at least $1/4$ (factoring in the probability we draw the majority sign). Achieving the same guarantees as above then simply requires running a small constant times as many rounds. Thus there is no asymptotic change to any of the complexity measures and we have the desired result. ■

We note that in all examples considered in this paper, $b(n)$, $T_{I_R}(n)$, and $T_C(n)$ are at worst quadratic, resulting in computationally efficient algorithms. Finally, it is worth briefly discussing the fact that it is possible to remove the query counter in the proof of Theorem 15 if one is willing to measure *expected* rather than *worst-case* query-complexity. This mainly involves running the same algorithm using only the sample cutoff and analyzing the probability that the algorithm makes a large number of queries.

Appendix B. Three Classes with Finite LCS

In this section we cover three examples of fundamental classes with small, efficiently computable lossless compression schemes: axis-aligned rectangles, decision trees, and halfspaces in two dimensions. In each of these cases standard lower bounds show that without additional queries, active learning provides no substantial benefit over its passive counterpart (Dasgupta, 2005). Furthermore, these classes cannot be actively RPU-learned at all (each requires an infinite number of queries) (Hopkins et al., 2020b). Thus we see that the introduction of natural enriched queries brings learning from an infeasible or even impossible state to one that is highly efficient, even on a very low memory device.

B.1. Axis-Aligned Rectangles

Despite being one of the most basic classes in machine learning, axis-aligned rectangles provide an excellent example of the failure of both standard active and RPU-learning. In this section we show that the introduction of a natural enriched query, the *odd-one-out* query, completely removes this barrier, providing a query-optimal, computationally efficient RPU-learner with bounded memory. Recall that axis-aligned rectangles over $\mathbb{R}^d$ are the indicator functions corresponding to products of intervals over $\mathbb{R}$:

$$R = [a_1, b_1] \times \ldots \times [a_d, b_d]$$

for $a_i \leq b_i$. Additionally we allow a_i to be $-\infty$, and b_i to be ∞ (though in this case the interval should be open). In other words, thinking of each coordinate in $\mathbb{R}^d$ as a feature, axis-aligned rectangles capture scenarios where each individual feature has an acceptable range, and the overall object is acceptable if and only if every feature is acceptable. For instance, one might measure a certain dish by flavor profile, including features such as saltiness, sourness, etc. If the dish is too salty, or too sour, the diner is unlikely to like it independent of its other features. In this context, the "odd-one-out" query asks the diner, given a negative example, to pick a specific feature they did not like, and further to specify whether the feature was either missing, or too present. Perhaps the dish was too sour, or needed more umami.

More formally, the odd-one-out oracle $\mathcal{O}_{\text{odd}} : H \times X \to P\left(([d] \times \{0, 1\}) \cup \{*\}\right)$ on input $h = [a_1, b_1] \times \ldots \times [a_d, b_d]$ and $x \in \mathbb{R}^d$ outputs $\{*\}$ if $L(h, x) = 1$, and otherwise outputs the set of pairs $(i, 1)$ such that $x_i > b_i$ and $(i, 0)$ such that $x_i < a_i$.

Proposition 16 *The class of axis-aligned rectangles over $\mathbb{R}^d$ has a lossless compression scheme of size at most $2d$ with respect to $\mathcal{O}_{odd}$.*

Proof Recall that lossless sample compression separately examines the set of positively and negatively labeled points. We first analyze the positive samples—those which lie inside the axis-aligned rectangle. In this case, notice that we may simply select a subsample T of size at most $2d$ which contains points with the maximum and minimal value at every coordinate. Since it is clear that T infers all points inside the rectangle $R(T)$ it spans, it is enough to argue that S cannot infer any point outside $R(T)$. This follows from the fact that both $R(T)$ and the all 1's function are consistent with queries on S.

For the case of a negatively labeled sample, notice that if any two $x, x' \in X$ have the same "odd-one-out" response (b, i), one must infer the response of the other. Informally, this is simply because the odd-one-out query measures whether some feature is too large or too small. If two points are

too large in some coordinate, say, then the point with the smaller feature infers this response in the other point. More formally, assume without loss of generality that $x_i \leq x'_i$. Then if the query response is $(0, i)$, $q(S)|_{x'} \to q(S)|_x$. Likewise, if the response is $(1, i)$, then $q(S)|_x \to q(S)|_{x'}$. Thus for each of the $2d$ query types we need only a single point to recover all query information for the original sample. Since no information is lost, this compressed set clearly satisfies the conditions of the desired LCS. ■

As an immediate corollary, we get that axis-aligned rectangles are actively RPU-learnable with near-optimal query complexity and bounded memory. Since the compression scheme and inference rule are efficiently computable, the algorithm is additionally computationally efficient.

Corollary 17 *The class of axis-aligned rectangles in $\mathbb{R}^d$ is RPU-learnable in only*

$$q(\varepsilon, \delta) = O\left(d \log\left(\frac{1}{\varepsilon\delta}\right)\right)$$

queries, $O(d)$ memory, and time $O\left(\frac{d^2 \log(d/(\varepsilon\delta)) \log(1/(\varepsilon\delta))}{\varepsilon}\right)$ when the learner has access to $\mathcal{O}_{odd}$.

Proof The first two statements follow immediately from Theorem 15 and noting that $b(n) \leq O(n)$. The runtime guarantee is slightly more subtle, and follows from noting that compression time $T_C(n) \leq O(dn)$, and that with a suitable representation of the compressed set, inference for a point $x \in \mathbb{R}^d$ only requires checking the value of each coordinate (in essence, we may act as though $T_I(d) \leq O(d)$). ■

Fixing parameters other than ε, we note that the optimal query complexity for axis-aligned rectangles (and all following examples) is $\Omega(\log(1/\varepsilon))$. This follows from standard arguments (Kulkarni et al., 1993), and can be seen simply by noting that there exists a distribution with at least $O(1/\varepsilon)$ ε-pairwise-separated concepts—the bound comes from noting that each query only provides $O(1)$ bits of information.

B.2. Decision Trees

While axis-aligned rectangles provide a fundamental example of a classifier for which enriched queries break the standard barriers of active RPU-learning and bounded memory, they are too simple a class in practice to model many situations of interest. In this section we consider a broad generalization of axis-aligned rectangles that is not only studied extensively in the learning literature (Ehrenfeucht and Haussler, 1989; Kushilevitz and Mansour, 1993; O'Donnell and Servedio, 2007; Kalai and Teng, 2008; Blanc et al., 2020), but also used frequently in practice (Quinlan, 1986; Umanol et al., 1994; Hssina et al., 2014; Singh and Gupta, 2014): decision trees. In this setting we consider a natural enriched query which roughly falls into a paradigm known as *same-cluster* queries (Ashtiani et al., 2016), which determine whether a given set of points lie in a "cluster" of some sort. Variants of this query have seen substantial study in the past few years in both clustering and learning Verroios et al. (2017); Mazumdar and Saha (2017); Ailon et al. (2018); Firmani et al. (2018); Saha and Subramanian (2019) after their recent formal introduction in Ashtiani et al. (2016).

More formally, recall that a decision tree over $\mathbb{R}^d$ is a binary tree where each node in the tree corresponds to an inequality:

$$x_i \overset{?}{\geq} b \text{ or } x_i \overset{?}{\leq} b,$$

measuring the i-th feature (coordinate) of any $x \in \mathbb{R}^d$. Each leaf in the tree is assigned a label, and the label of any $x \in X$ is uniquely determined by the leaf resulting from following the decision tree from root to leaf, always taking the path determined by the inequality at each node. We measure the size of a decision tree by counting its leaves. We introduce a strong enriched query for decision trees in the same-cluster paradigm called the *same-leaf* oracle $\mathcal{O}_{\text{leaf}}$. In particular, given a decision tree h and two points x, x', the same-leaf query on x and x' simply determines whether x and x' lie in the same leaf of h. It is worth noting that the same-leaf oracle can be seen as a strengthening of Dasgupta et al. (2018)'s "discriminative features," a method of dividing a decision tree into clusters, each of which should have some discriminating feature. Same-leaf queries take this idea to its logical extreme where each leaf should be thought of as a separate cluster.

We will show that same-leaf queries have a small and efficiently computable lossless compression scheme with respect to an efficient inference rule R_{rect}, and therefore have query-optimal and computationally efficient bounded-memory RPU-learners. The inference rule R_{rect} is a simple restriction where $x \in I_{R_{\text{rect}}}(q(S))$ if and only if x lies inside a rectangle spanned by a mono-leaf subset $T \subseteq S$, that is a subset T for which $\forall x, x' \in T, \mathcal{O}_{\text{leaf}}(h, x, x') = 1$. In other words, we infer information independently for each leaf.

Proposition 18 *The class of size s decision trees over $\mathbb{R}^d$ has an LCS of size at most $2ds$ with respect to $\mathcal{O}_{leaf}$ and R_{rect}.*

Proof Notice that any leaf of a decision tree corresponds to an axis-aligned rectangle in $\mathbb{R}^d$. Further, for any $S \subset \mathbb{R}^d$, $q(S)$ allows us to partition S into subsamples sharing the same leaf. Fix one such subsample and denote it by S_L. By the same argument as Proposition 16, there exists $T \subseteq S_L$ of size at most $2d$ such that S_L lies entirely within the rectangle spanned by T. Since these are exactly the points inferred by S in that leaf under inference rule R_{rect}, repeating this process over all s leaves gives the desired LCS. ■

Corollary 19 *The class of size s decision trees is RPU-learnable in only*

$$q(\varepsilon, \delta) = O\left(ds^2 \log\left(\frac{1}{\varepsilon\delta}\right)\right)$$

queries, $O(ds)$ memory, and time $O\left(\frac{d^2 s^2 \log(ds/(\varepsilon\delta)) \log(1/(\varepsilon\delta))}{\varepsilon}\right)$ when the learner has access to $\mathcal{O}_{leaf}$.

Proof The proof follows similarly to Corollary 17. It is not hard to see that $T_C(n)$ is at worst quadratic. Determining queries on a set of size n reduces to grouping points into s buckets (each standing for some leaf), which can be done in $b(n) \leq O(ns)$ queries. While $T_{I_{R_{\text{rect}}}}(n)$ would technically require finding the rectangles implied by the given sample and checking x against each one, the former can be thought of pre-processing since it need only be done once per round. With this process removed, checking x takes only $O(d)$ time, which gives the desired result. ■

While it is not possible to learn decision trees of arbitrary size in the standard learning models (Hancock et al., 1996), one might hope that Corollary 19 can be used to build a learner for this class with nice guarantees. Indeed this is possible if we weaken our memory and computational complexity measures to be expected rather than worst-case. We will show that in this regime,

Corollary 19 can be bootstrapped to build an *attribute-efficient* algorithm for RPU-learning decision trees, that is one in which the expected memory, query, and computational complexity scale with the (unknown) size of the underlying decision tree.

Corollary 20 *There exists an algorithm for RPU-learning decision trees over $\mathbb{R}^d$ which in expectation:*

1. *Makes $O\left(ds^2 \log\left(\frac{s}{\varepsilon\delta}\right)\right)$ queries*
2. *Runs in time poly$(s, d, \varepsilon^{-1}, \log(\delta^{-1}))$*
3. *Uses $O(ds)$ memory,*

where s is the size of the underlying decision tree.

Proof For simplicity, we use Corollary 19 as a blackbox for an increasing "guess" s' on the size of the decision tree. After each application of Corollary 19, we test its coverage empirically. If too many points go un-inferred, we double our guess for s' and continue the process. More formally, we start by setting our guess s' to 2. The learner then applies Corollary 19 with accuracy parameters $\delta' = (\delta/s')$ and $\varepsilon' = O(\varepsilon/(s' \log(s'/\delta)))$. For ease of analysis, we assume that each application of Corollary 19 uses some fixed number of samples $N_{s'}$ (given by its corresponding sample complexity). This can be done by arbitrarily inflating the number of samples drawn if the query cutoff is reached before the sample cutoff (and likewise for the query counter if the sample cutoff is reached first). Fixing this quantity allows the algorithm to "know" what step it is in by checking the sample and query counters.[8]

After each application of Corollary 19, we draw $M_{s'} = O(\log(s'/\delta)/\varepsilon)$ points. If a single point in this sample is un-inferred by the output of Corollary 19, we double s' and continue the process. Otherwise, we output the classifier given by Corollary 19 in that round. Notice that tracking the existence of an un-inferred point in this sample takes only a single bit, and moreover that since the sample size of each round is fixed to be $N_{s'} + M_{s'}$, the algorithm can still use the counters to track its position at any step. Finally, we note that for a given s', if the algorithm does not have coverage at least $1-\varepsilon$ it aborts with probability at most $\frac{\delta}{s'}$ by our choice of $M_{s'}$. Since s' starts at 2 and doubles each round, a union bound gives that the probability the algorithm ever aborts with coverage worse than $1-\varepsilon$ is at most δ as desired.

It is left to compute the expected memory usage, query complexity, and computational complexity of our algorithm. Notice that at any given step with guess s', our algorithm runs in time

$$O\left(\frac{d^2 s'^2 \log(ds'/(\varepsilon\delta)) \log(s'/(\varepsilon\delta))}{\varepsilon}\right),$$

uses $O(ds')$ memory, and makes at most $O(ds'^2 \log(s'/\varepsilon\delta))$ queries. Further, for every round in which $s' \geq s$, the samples drawn are sufficiently large that Corollary 19 is guaranteed to succeed with probability at least $1-O(1/s')$ by our chosen parameters. Since the coverage test also succeeds with probability at least $1 - O(1/s')$ and each round is independent, for any $s' \geq 16s$ the failure probability at that step is at most $O(1/s'^4)$ (as the algorithm has at this point run at least four rounds

8. More formally, this just means there is a well-defined transition function for the algorithm which relies on the counters to move between potential tree sizes.

with $s' \geq s$). Since all our complexity measures scale like $o(s'^3)$, the expected contribution to time, query complexity, and memory usage at least half in each step for which $s' > 16s$. It is not hard to observe that these final steps then add no additional asymptotic complexity, which gives the desired result. ■

B.3. Halfspaces in Two Dimensions

Much of the prior work on active learning with enriched queries centers around the class of halfspaces (Kane et al., 2017; Xu et al., 2017; Hopkins et al., 2020b,a; Cui and Sato, 2020). While it has long been known the class cannot be efficiently active learned in the standard model (Dasgupta, 2005), Kane et al. (2017) showed that adding a natural enriched query called a *comparison* resolves this issue. In particular, recall that a 2D-halfspace is given by the sign of the inner product with some normal vector $v \in \mathbb{R}^2$ plus some bias $b \in \mathbb{R}$. A comparison query on two points $x, x' \in \mathbb{R}^2$ measures which point is further from the hyperplane defined by h, that is:

$$\langle x, v \rangle \overset{?}{\geq} \langle x', v \rangle.$$

Kane et al. (2017) proved that halfspaces in two dimensions have finite inference dimension, a combinatorial parameter slightly weaker than perfect sample compression that characterizes query-optimal active learning. In fact, we show that 2D-halfspaces with comparisons have substantially stronger structure—namely an LCS of size 5.

Proposition 21 *The class of halfspaces over $\mathbb{R}^2$ has a size 5 lossless compression scheme with respect to the comparison oracle.*

Proof Given some (unknown) hyperplane $h = \langle v, \cdot \rangle + b$, monochromatic set S, and labels and comparisons $q_h(S)$, we must prove the existence of a subset T of size at most 5 such that $I(q_h(T)) = I(q_h(S))$. The idea behind our construction is to consider positive rays based on S and $q_h(S)$. In other words, notice that for any point $s \in S$ and pair (x_1, y_1) such that $h(x_1) \geq h(y_1)$, the ray $r_1 = s + t(x_1 - y_1)$ is *increasing* with respect to h. Furthermore, any two such rays with the same base point $r_1 = s + t(x_1 - y_1)$ and $r_2 = s + t(x_2 - y_2)$ form a *cone*

$$C(r_1, r_2) = \left\{ y \in \mathbb{R}^2 : \exists \alpha_1, \alpha_2 > 0 \text{ s.t. } y = s + \sum_{i=1}^{2} \alpha_i (x_i - y_i) \right\}$$

such that for any $y \in C(r_1, r_2)$, labels and comparisons on the set $\{s, x_1, x_2, y_1, y_2\}$ infer that $h(y) \geq 0$:

$$\begin{aligned} h(y) &= \left\langle v, s + \sum_{i=1}^{2} \alpha_i (x_i - y_i) \right\rangle + b \\ &= (\langle v, s \rangle + b) + \sum_{i=1}^{2} \langle v, \alpha_i (x_i - y_i) \rangle \\ &= h(s) + \alpha_1 (h(x_1) - h(y_1)) + \alpha_2 (h(x_2) - h(y_2)) \\ &\geq 0. \end{aligned}$$

Notice that the union of all such cones is itself a cone where the base point s is some minimal element in S (in the sense that for all $s' \in S, h(s) \leq h(s')$), and r_1 and r_2 are rays stemming from s that make the greatest angle. We argue that $y \in I(q_h(S))$ if and only if y lies inside this cone. Since the cone may be defined by queries on the 5 points making up r_1 and r_2, this gives the desired compression scheme. See Figure 1 for a pictorial description of this process.

Denote the cone given by this process by C. We have already proved that $y \in I(q_h(S))$ if $y \in C$, so it is sufficient to show that any $y \notin C$ cannot be inferred by queries on S. To see this, we examine the bounding hyperplanes of C: $s + t(x_1 - y_1)$, and $s + t(x_2 - y_2)$. Let H_1 and H_2 denote the corresponding halfspaces (defined such that $C = H_1 \cap H_2$), and define $h_i = \langle v_i, \cdot \rangle + b_i$ such that $H_i = \text{sign}(h_i)$. We first note that the labels and comparisons on S corresponding to each h_i are consistent with those on the true hypothesis h. This is obvious for label queries since S is assumed to be entirely positive, and C contains S by definition. For comparisons, assume for the sake of contradiction that there exists a query inconsistent with some h_i, that is a pair $x, y \in S$ such that $h(x) \geq h(y)$ but $h_i(x) < h_i(y)$. If this is the case, notice that the ray extending from y through x is decreasing with respect h_i, and therefore must eventually cross it. It follows that replacing (x_i, y_i) in the construction of T with (x, y) results in a wider angle, which gives the desired contradiction.

Given that h_1 and h_2 are both consistent with queries on S under the true hypothesis, consider a point y lying outside of C. By definition, either $\text{sign}(h_1(y))$ or $\text{sign}(h_2(y))$ is negative. However, notice that since $h_i = \langle v_i, \cdot \rangle + b_i$ is consistent with queries on S, it must also be the case that any non-negative shift $h_{i,b'} = \langle v_i, \cdot \rangle + b_i + b'$ for $b' \geq 0$ is consistent as well. Since for sufficiently large b', $\text{sign}(h_{i,b'}(y))$ is positive, the true label $\text{sign}(h(y))$ cannot be inferred as desired. ■

Corollary 22 *The class of halfspaces over $\mathbb{R}^2$ is actively RPU-learnable in only*

$$q(\varepsilon, \delta) \leq O\left(\log\left(\frac{1}{\varepsilon\delta}\right)\right)$$

queries, $O(1)$ memory, and time $O\left(\frac{\log^2(1/(\delta\varepsilon))}{\varepsilon}\right)$.

Proof Inference is done through a linear program as in Kane et al. (2017). All computational parameters are $O(1)$ due to being in $O(1)$ dimensions, which gives the desired result. ■

Appendix C. Further Directions

We end with a brief discussion of two natural directions suggested by our work.

C.1. Characterizing Bounded Memory Active Learning

In this work we prove that lossless sample compression is a sufficient condition for efficient, bounded memory active learning in the enriched query regime. Kane et al. (2017) prove that inference dimension, a strictly weaker combinatorial parameter for enriched queries, is necessary for efficient bounded memory active learning. Closing the gap between these two conditions remains an open problem—it is currently unknown whether inference dimension is sufficient or lossless sample compression is necessary. Similarly, the relation between inference dimension and lossless sample compression themselves remains unknown. Over finite spaces it is clear from arguments of Kane et al.

(2017) that inference dimension and lossless sample compression are equivalent up to a factor of $\log(|X|)$. However, since in the finite regime we are likely interested in learning all points in $|X|$ rather than to some accuracy parameter, the relevant memory bound depends crucially on $\log(|X|)$, making the tightness of the above relation an important question as well.

On a related note, though halfspaces in dimensions greater than two have infinite inference dimension, Kane et al. (2017) do show that halfspaces with certain restricted structure (e.g. bounded bit complexity, margin) have finite inference dimension. Whether such classes have lossless compression schemes, or indeed are even learnable in bounded memory at all remains an interesting open problem and a concrete step towards understanding the above.

C.2. Learning with Noise

In this work we study only *realizable* case learning. It is reasonable to wonder to what extent our results hold in the *agnostic* model, where the adversary may choose any function (rather than being restricted to one from the concept class H), or various models of noise in which the adversary may corrupt queries. While inference-based learning is difficult in such regimes, previous work has seen some success with particular classes of enriched queries such as comparisons (Xu et al., 2017; Hopkins et al., 2020b). Proving the existence of *bounded memory* active learners even for simple noise and query regimes such as random classification noise with comparisons remains an important problem for bringing bounded memory active learning closer to practice.