Carnegie Mellon University
Research Showcase @ CMU

Machine Learning Department

School of Computer Science

6-2012

Robust Interactive Learning

Maria-Florina Balcan
Georgia Institute of Technology, ninamf@cs.cmu.edu

Steve Hanneke
Carnegie Mellon University

Follow this and additional works at: http://repository.cmu.edu/machine_learning

Part of the Computer Sciences Commons

Published In
Journal of Machine Learning Research, 23, 1-34.

This Article is brought to you for free and open access by the School of Computer Science at Research Showcase @ CMU. It has been accepted for inclusion in Machine Learning Department by an authorized administrator of Research Showcase @ CMU. For more information, please contact research-showcase@andrew.cmu.edu.

JMLR: Workshop and Conference Proceedings vol (2012) 1–34

Robust Interactive Learning

Maria Florina Balcan NINAMF@CC.GATECH.EDU
Georgia Institute of Technology, School of Computer Science

Steve Hanneke SHANNEKE@STAT.CMU.EDU
Carnegie Mellon University, Department of Statistics

Abstract

In this paper we propose and study a generalization of the standard active-learning model where a more general type of queries including class conditional queries and mistake queries are allowed. Such queries have been quite useful in applications, but have been lacking theoretical understanding. In this work, we characterize the power of such queries under several well-known noise models. We give nearly tight upper and lower bounds on the number of queries needed to learn both for the general agnostic setting and for the bounded noise model. We further show that our methods can be made adaptive to the (unknown) noise rate, with only negligible loss in query complexity.

Keywords: Statistical Learning Theory, Interactive Learning, Query Complexity, Active Learning

1. Introduction

The ever-expanding range of application areas for machine learning, together with huge increases in the volume of raw data available, has encouraged researchers to look beyond the classic paradigm of passive learning from labeled data only. Perhaps the most extensively used and studied technique in this context is Active Learning, where the algorithm is presented with a large pool of unlabeled examples (such as all images available on the web) and can interactively ask for the labels of examples of its own choosing from the pool. The aim is to use this interaction to drastically reduce the number of labels needed (which are often the most expensive part of the data collection process) in order to reach a low-error hypothesis.

Over the past fifteen years there has been a great deal of progress on understanding active learning and its underlying principles Freund et al. (1997); Balcan et al. (2006, 2007); Beygelzimer et al. (2009); Castro and Nowak (2007); Dasgupta et al. (2007, 2005); Hanneke (2007a); Balcan et al. (2008); Hanneke (2009); Koltchinskii (2010); Wang (2009); Beygelzimer et al. (2010). However, while useful in many applications McCallum and Nigam (1998); Tong and Koller (2001), requesting the labels of select examples is only one very specific type of interaction between the learning algorithm and the labeler. When analyzing many real world situations, it is desirable to consider learning algorithms that make use of other types of queries as well. For example, suppose we are actively learning a multiclass image classifier from examples. If at some point, the algorithm needs an image from one of the classes, say an example of "house", then an algorithm that can only make individual label requests may need to ask the expert to label a large number of unlabeled examples before it finally finds an example of a house for the expert to label as such. This problem could be averted by simply allowing the algorithm to display a list of around a hundred thumbnail images on the screen, and ask the expert to point to an image of a house if there is one. The expert can visually scan through those images looking for a house much more quickly than she can label every

© 2012 M.F. Balcan & S. Hanneke.

one of them. We call such queries class conditional queries. As another example of a different type of query, the algorithm could potentially select a subset of the unlabeled data and ask the expert to point to two examples of opposite labels within a specified distance of each other (for instance, by Euclidean distance after projecting the data to a 2-dimensional space) and provide back the labels of those examples. As a third example, based on the data and interaction so far, the algorithm could propose a labeling of a set of unlabeled images and ask for a few mistakes if any exist – we call these mistake queries or sample-based equivalence queries. Queries of this type are commonly used by commercial systems (*e.g.*, Faces in Apple-iPhoto makes use of mistake queries for face recognition and labeling), and have been studied in several papers Chang et al. (2005); Doyle et al. (2009), but unfortunately have been lacking a principled theoretical understanding.

In this work we expand the study of active learning by considering a model that allows us to analyze learning with types of queries motivated by such applications. For most of our analysis, we focus on class-conditional queries, where the algorithm is able to select a subset of a pool of unlabeled examples and request the oracle an example of a given label within that subset, if one exists. Our results additionally have immediate implications for mistake queries, in which the algorithm may instead ask for a mistake within the selected subset of unlabeled examples, for an arbitrary specified classifier.[1] In these cases, we provide nearly tight bounds on query complexity under several commonly studied noise conditions. We also discuss how our techniques could be adapted to a more general setting involving abstract families of queries.

Class Conditional Queries It is well known that if the target function resides in a known concept class and there is no classification noise (the so-called *realizable case*), then a simple approach based on the Halving algorithm Littlestone (1988) can learn a function ϵ-close to the target function using a number of class conditional queries dramatically smaller than the number of random labeled examples required for PAC learning Hanneke (2009).

In this paper, we provide the first results for the more realistic non-realizable setting. Specifically, we provide general and nearly tight results on the query complexity of class-conditional queries in a multiclass setting under some of the most widely studied noise models including random classification noise, bounded noise, as well as the purely agnostic setting.

In the purely agnostic case with noise rate η, we show that any interactive learning algorithm in this model seeking a classifier of error at most $\eta + \epsilon$ must make $\Omega(d\eta^2/\epsilon^2)$ queries, where d is the Natarajan dimension; we also provide a nearly matching upper bound of $\tilde{O}(d\eta^2/\epsilon^2)$, for a constant number of classes. This is smaller by a factor of η compared to the sample complexity of passive learning (see Lemma 10), and represents a reduction over the known results for the query complexity of active learning in many cases.

In the bounded noise model, we provide nearly tight upper and lower bounds on the query complexity of class conditional queries as a function of the query complexity of active learning. In particular, we find that the query complexity of the class conditional query model is essentially within a factor of the noise bound of the query complexity of active learning. Interestingly, both our upper and lower bounds are proven via reductions from active learning. In the case of the upper bound, we illustrate a technique for using the method developed for the purely agnostic case as a subroutine in batch-based active learning algorithms, using it to get the labels of all samples in a given batch of unlabeled data.

[1] We note that both class conditional queries and mistake queries strictly generalize the traditional model of active learning by label requests.

We additionally study learning in the one-sided noise model, and show that in the case of intersection-closed concept classes, it is possible to get around our lower bounds and recover the much-better realizable-case query complexity of $\tilde{O}(d\log(1/\epsilon))$. Our analysis of this scenario is based on recent analyses of the frequency of mistakes made by the Closure algorithm along a sequence of i.i.d examples.

We further show that our methods can be made adaptive to the (unknown) noise rate η, with only negligible loss in query complexity. Specifically, our method for the purely agnostic case has the property that it produces a correctly labeled pool of i.i.d. labeled examples. We are able to use this property in both the agnostic and bounded noise settings as a way to verify that the method is successful; combined with a guess-and-double trick, this allows us to adapt to the noise rate. The method we develop for one sided noise naturally adapts to the unknown noise rate.

Overall, we find that the reductions in query complexity for this model, compared to the traditional active learning model, [2] largely concerned with a factor relating to the noise rate of the learning problem, so that the closer to the realizable case we are, the greater the potential gains in query complexity. However, for larger noise rates, the benefits are more modest, a fact that sharply contrasts with the enormous benefits of using these types of queries in the realizable case; this is true even for very benign types of noise, such as bounded noise. On this, it is interesting to note that, for both active learning and for passive learning, the difference between the realizable case sample complexity and bounded-noise sample complexity is at most a logarithmic factor (considering the noise bound as a constant). As a result, bounded noise is typically considered quite benign in passive and active learning. What our work shows is that, quite surprisingly, this trend fails to hold for class-conditional queries. That is, comparing the query complexity for the realizable case to that of the bounded noise case, there is often a *dramatic* increase. Specifically, while in the realizable case, the query complexity is *always* $O(d\log(1/\epsilon))$, when we move to the bounded noise case (with constant noise bound), the query complexity jumps up to be essentially proportional to the label complexity of *active* learning. Interestingly, both our upper and lower bounds are proven via reductions from active learning.

Other General Queries We additionally generalize these techniques and results to apply in more general setting, making them available for many other types of queries. Specifically, we prove upper bounds on the query complexity for an abstract type of sample-dependent query, for both the general agnostic case and for the bounded noise case. The results are similar to those obtained for class-conditional queries, except that they are multiplied by a complexity measure defined in terms of the specific family of queries available to the algorithm. The methods achieving these bounds are themselves somewhat more involved than those presented for class-conditional queries. In contrast to the results on class-conditional queries, we do not establish corresponding lower bound or tightness results for these more general cases.

Related Work Early work in the the exact learning literature also considers more general type of queries Angluin (1998); Balcázar et al. (2002, 2001). Our results are different from those in several respects. First, following the active learning literature, we are concerned with the case where we can

[2] We are slightly overloading the meaning of reduction here since class conditional queries, mistake queries, and the more general type of queries we consider are technically incomparable with active learning queries (label requests). We note however that answering for example a class conditional query or a mistake query on a query set S could be significantly easier than labeling all the examples in S which can only be achieved by $|S|$ label requests. This is observed in practice and also demonstrated by the fact that such queries are incorporated in commercial applications such as Faces in Apple-iPhoto.

ask queries only on subsets of our large pool of unlabeled examples, rather than directly on subsets of the instance space of our choosing. Second, we are mainly concerned with achieving tight query complexity guarantees in the presence of *noise* (e.g., purely agnostic or bounded noise). By contrast, the earlier work on exact learning has been focused on noise-free learning (the *realizable* case). Both of these differences make our treatment more appropriate and realistic for the statistical learning setting. Technically, our methods blend and extend the techniques of the classical literature on Exact Learning with the more recent literature on active learning in the statistical learning setting. Some of our results also have novel implications for the traditional active learning setting; in particular, we present the first query complexity bounds under bounded noise in terms of the splitting index.

Due to lack of space, we only include proof sketches of our results for class conditional queries in the main body, with further details in the appendices. We provide our results about one-sides noise appear in Appendix D and our results for other types of queries appear in Appendix E.

2. Formal Setting

We consider an interactive learning setting defined as follows. There is an *instance space* $\mathcal{X}$, a *label space* $\mathcal{Y}$, and some fixed *target distribution* $\mathcal{D}_{XY}$ over $\mathcal{X} \times \mathcal{Y}$, with marginal $\mathcal{D}_X$ over $\mathcal{X}$. Focusing on multiclass classification, we assume that $\mathcal{Y} = \{1, 2, \ldots, k\}$, for some $k \in \mathbb{N}$. In the learning problem, there is an i.i.d. sequence of random variables $(x_1, y_1), (x_2, y_2), (x_3, y_3), \ldots$, each with distribution $\mathcal{D}_{XY}$. The learning algorithm is permitted direct access to the sequence of x_i values (unlabeled data points). However, information about the y_i values is obtainable only via interaction with an oracle, defined as follows.

At any time, the learning algorithm may propose a label $\ell \in \mathcal{Y}$ and a finite subsequence of unlabeled examples $S = \{x_{i_1}, ..., x_{i_m}\}$ (for any $m \in \mathbb{N}$); if $y_{i_j} \neq \ell$ for all $j \leq m$, the oracle returns "none." Otherwise, the oracle selects an arbitrary $x_{i_j} \in S$ for which $y_{i_j} = \ell$ and returns the pair (x_{i_j}, y_{i_j}). In the following we call this model the CCQ (class-conditional queries) interactive learning model. Technically, we implicitly suppose the set S also specifies the unique indices of the examples it contains, so that the oracle knows which y_i corresponds to which x_{i_j} in the sample S; however, we make this detail implicit below to simplify the presentation.

In the analysis below, we fix a set of classifiers $h : \mathcal{X} \to \mathcal{Y}$ called the *hypothesis class*, denoted $\mathbb{C}$. We will denote by d the Natarajan dimension of $\mathbb{C}$ Natarajan (1989); Haussler and Long (1995); Ben-David et al. (1995), defined as the largest $m \in \mathbb{N}$ such that $\exists (a_1, b_1, c_1), \ldots, (a_m, b_m, c_m) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Y}$ with $b_i \neq c_i$ for each i s. t. $\{b_1, c_1\} \times \cdots \times \{b_m, c_m\} \subseteq \{(h(a_1), \ldots, h(a_m)) : h \in \mathbb{C}\}$.[3] The Natarajan dimension has been calculated for a variety of hypothesis classes, and is known to be related to other commonly used dimensions, including the pseudo-dimension and graph dimension Haussler and Long (1995); Ben-David et al. (1995). For instance, for neural networks of n nodes with weights given by b-bit integers, the Natarajan dimension is at most $bn(n-1)$ Natarajan (1989).

For any $h : \mathcal{X} \to \mathcal{Y}$ and distribution P over $\mathcal{X} \times \mathcal{Y}$, define the *error rate* of h as $\mathrm{err}_P(h) = \mathbb{P}_{(X,Y)\sim P}\{h(X) \neq Y\}$; when $P = \mathcal{D}_{XY}$, we abbreviate this as $\mathrm{err}(h)$. For any finite sequence of labeled examples $L = \{(x_{i_1}, y_{i_1}), \ldots, (x_{i_m}, y_{i_m})\}$, we define the empirical error rate $\mathrm{err}_L(h) = |L|^{-1} \sum_{(x,y)\in L} \mathbb{I}[h(x) \neq y]$. In some contexts, we also refer to the empirical error rate on a finite sequence of *unlabeled* examples $U = \{x_{i_1}, \ldots, x_{i_m}\}$, in which case we simply define $\mathrm{err}_U(h) = |U|^{-1} \sum_{x_{i_j}\in U} \mathbb{I}[h(x_{i_j}) \neq y_{i_j}]$, where the y_{i_j} values are the actual labels of these examples.

[3] If there are only two classes the Natarajan dimension is equal to the VC dimension.

Let h^* be the classifier in $\mathbb{C}$ of smallest $\mathrm{err}(h^*)$ (for simplicity, we suppose the minimum is always realized), and let $\eta = \mathrm{err}(h^*)$, called the *noise rate*. The objective of the learning algorithm is to identify some h with $\mathrm{err}(h)$ close to η using only a small number of queries. In this context, a *learning algorithm* is simply any algorithm that makes some number of queries and then halts and returns a classifier. We are particularly interested in the following quantity.

Definition 1 *For any $\epsilon, \delta \in (0,1)$, any hypothesis class $\mathbb{C}$, and any family of distributions $\mathbb{D}$ on $\mathcal{X} \times \mathcal{Y}$, define the quantity $\mathrm{QC}_{\mathrm{CCQ}}(\epsilon, \delta, \mathbb{C}, \mathbb{D})$ as the minimum $q \in \mathbb{N}$ such that there exists a learning algorithm $\mathcal{A}$, which for any target distribution $\mathcal{D}_{XY} \in \mathbb{D}$, with probability at least $1-\delta$, makes at most q queries and then returns a classifier $\hat{h}$ with $\mathrm{err}(\hat{h}) \leq \eta + \epsilon$. We generally refer to the function $\mathrm{QC}_{\mathrm{CCQ}}(\cdot, \cdot, \mathbb{C}, \mathbb{D})$ as the* query complexity *of learning $\mathbb{C}$ under $\mathbb{D}$.*

The query complexity, as defined above, represents a kind of minimax statstical analysis, where we fix a family of possible target distributions $\mathbb{D}$, and calculate, for the best possible learning algorithm, how many queries it makes under its worst possible target distribution $\mathcal{D}_{XY}$ in $\mathbb{D}$. Specific families of target distributions we will be interested in include the random classification noise model, the bounded noise model, and the agnostic model which we define formally in the sections below.

3. The General Agnostic Case

We start by considering the most general, *agnostic* setting, where we consider arbitrary noise distributions subject to a constraint on the noise rate. This is particularly relevant to many practical scenarios, where we often do not know what type of noise we are faced with, potentially including stochastic labels or model misspecification, and would therefore like to refrain from making any specific assumptions about the nature of the noise. Formally, consider the family of distributions $\mathcal{A}\mathrm{gnostic}(\mathbb{C}, \alpha) = \{\mathcal{D}_{XY} : \inf_{h \in \mathbb{C}} \mathrm{err}(h) \leq \alpha\}$, $\alpha \in [0, 1/2)$. We prove nearly tight upper and lower bounds on the query complexity of our model. Specifically, supposing k is constant, we have:

Theorem 2 *For any hypothesis class $\mathbb{C}$ of Natarajan dimension d, for any $\eta \in [0, 1/32)$,*

$$\mathrm{QC}_{\mathrm{CCQ}}(\epsilon, \delta, \mathbb{C}, \mathcal{A}\mathrm{gnostic}(\mathbb{C}, \eta)) = \tilde{\Theta}\left(d\frac{\eta^2}{\epsilon^2}\right).$$

The first interesting thing is that our bound differs from the sample complexity of passive learning only in a factor of η (see Lemma 10). This contrasts with the realizable case, where it is possible to learn with a query complexity that is exponential smaller than the query complexity of passive learning. On the other hand, is also interesting that this factor of η is consistently available regardless of the structure of the concept space. This contrasts with active learning where the extra factor of η is only available in certain special cases Hanneke (2007a).

3.1. Proof of the Lower Bound

We first prove the lower bound. We specifically prove that for $0 < 2\epsilon \leq \eta < 1/4$,

$$\mathrm{QC}_{\mathrm{CCQ}}(\epsilon, 1/4, \mathbb{C}, \mathcal{A}\mathrm{gnostic}(\mathbb{C}, \eta)) = \Omega\left(d\eta^2/\epsilon^2\right).$$

Monotonicity in δ extends this to any $\delta \in (0, 1/4]$. In words, this says that there is no algorithm based on class-conditional queries that, in the worst case, with probability greater than $3/4$, makes fewer than $O(d\eta^2/\epsilon^2)$ queries and returns a classifier h with $\mathrm{err}(h) \leq \eta + \epsilon$.

Proof The key idea of the proof is to provide a reduction from the (binary) active learning model (label request queries) to our multiclass interactive learning model (general class-conditional queries) for the hard case known previously for the active learning model Beygelzimer et al. (2009).

In particular, consider a set of d points x_0, x_1, x_2,..., x_{d-1} shattered by $\mathbb{C}$, and let $(y_0, z_0), \ldots, (y_{d-1}, z_{d-1})$ be the label pairs that witness the shattering. Here is a distribution over $\mathcal{X} \times \mathcal{Y}$: point x_0 has probability $1 - \beta$, while each of the remaining x_i has probability $\beta/(d-1)$, where $\beta = 2(\eta + 2\epsilon)$. At x_0 the response is always $Y = y_0$. At x_i, $1 \le i \le d-1$, the response is $Y = z_i$ with probability $1/2 + \gamma b_i$ and $Y = y_i$ with probability $1/2 - \gamma b_i$, where b_i is either $+1$ or -1, and $\gamma = 2\epsilon/\beta = \epsilon/(\eta + 2\epsilon)$.

Beygelzimer et al. (2009) show that for any active learning algorithm, one can set $b_0 = 1$ and all the b_i, $i \in \{1, \ldots, d-1\}$ in a certain way so that the algorithm must make $\Omega(d\eta^2/\epsilon^2)$ label requests in order to output a classifier of error at most $\eta + \epsilon$ with probability at least $1/2$. Building on this, we can show any interactive learning algorithm seeking a classifier of error at most $\eta + \epsilon$ must make $\Omega(d\eta^2/\epsilon^2)$ queries to succeed with probability at least $3/4$, as follows.

Assume that we have an algorithm $\mathcal{A}$ that works for the CCQ model with query complexity $\mathrm{QC}_{\mathrm{CCQ}}(\epsilon, \delta, \mathbb{C}, \mathcal{A}\mathrm{gnostic}(\mathbb{C}, \eta))$. We show how to use $\mathcal{A}$ as a subroutine in an active learning algorithm that is specifically tailored to the above hard set of distributions.

In particular, we can simulate an oracle for the CCQ algorithm as follows. Suppose our CCQ algorithm queries with a set S_i for a label ℓ. If ℓ is not one of the $y_0, \ldots, y_{d-1}, z_0, \ldots, z_{d-1}$ labels, we may immediately return that none exist. If there exists $x_{i,j} \in S_i$ such that $x_{i,j} = x_0$ and $\ell = z_0$, then we may simply return to the algorithm this $(x_{i,j}, z_0)$. Otherwise, we need only make (in expectation) $\frac{1}{1/2-\gamma}$ active learning queries to respond to the class-conditional query, as follows. We consider the subset R_i of S_i of points $x_{i,j}$ among those x_j with $\ell \in \{y_j, z_j\}$. We pick an example $x_i^{(1)}$ at random in R_i and request its label $y_i^{(1)}$. If $x_i^{(1)}$ has label $y_i^{(1)} = \ell$, then we return to the algorithm $(x_i^{(1)}, y_i^{(1)})$; otherwise, we continue sampling random $x_i^{(2)}, x_i^{(3)}, \ldots$ points from R_i (whose labels have not yet been requested) and requesting their labels $y_i^{(2)}, y_i^{(3)}, \ldots$, until we find one with label ℓ, at which point we return to the algorithm that example. If we exhaust R_i without finding such an example, we return to the algorithm that no such point exists. Since each $x_{i,j} \in R_i$ has probability at least $1/2 - \gamma$ of having $y_{i,j} = \ell$, we can answer any query of $\mathcal{A}$ using in expectation no more than $\frac{1}{1/2-\gamma}$ label request queries.

In particular, we can upper bound this number of queries by a geometric random variable and apply concentration inequalities for geometric random variables to bound the total number of label requests, as follows. Let A_i be a random variable indicating the actual number of label requests we make to answer query number i in the reduction above, before returning a response. For $j \le A_i$, if $h^*(x_i^{(j)}) \neq \ell$, let $Z_j = I[y_i^{(j)} = \ell]$, and if $h^*(x_i^{(j)}) = \ell$, let C_j be an independent Bernoulli$((1/2 - \gamma)/(1/2 + \gamma))$ random variable, and let $Z_j = C_j I[y_i^{(j)} = \ell]$. For $j > A_i$, let Z_j be an independent Bernoulli$(1/2-\gamma)$ random variable. Let $B_i = \min\{j : Z_j = 1\}$. Since, $\forall j \le A_i$, $Z_j \le I[y_i^{(j)} = \ell]$, we clearly have $B_i \ge A_i$. Furthermore, note that the Z_j are independent Bernoulli$(1/2-\gamma)$ random variables, so that B_i is a Geometric$(1/2 - \gamma)$ random variable. By Lemma 9 in Appendix A, we obtain that with probability at least $3/4$ we have that if Q is any constant and $\mathcal{A}$ makes $\le Q$ queries, then with probability greater than $3/4$, $\sum_i A_i \le \sum_{i=1}^{Q} B_i \le \frac{2}{1/2-\gamma}[Q + 4\ln(4)]$.

Without loss, we can suppose $\mathcal{A}$ makes at most $Q = \mathrm{QC}_{\mathrm{CCQ}}(\epsilon, 1/4, \mathbb{C}, \mathcal{A}\mathrm{gnostic}(\mathbb{C}, \eta))$ queries (otherwise, simply halt the algorithm if it exceeds this, and it will still achieve this optimal query complexity). Since $\sum_i A_i$ represents the total number of label requests made by this algorithm, we have that if $\mathrm{QC}_{\mathrm{CCQ}}(\epsilon, 1/4, \mathbb{C}, \mathcal{A}\mathrm{gnostic}(\mathbb{C}, \eta)) < \frac{1/2-\gamma}{2} m - 4\ln(4)$, where $m = O(d\eta^2/\epsilon^2)$ is the Beygelzimer et al. (2009) lower bound, then with probability $> 3/4$, the number of label requests is $< m$. Since any algorithm making $< m$ queries fails with probability at least $1/2$, there is a greater than $1/4$ probability that the number of label request is $< m$ *and* the above active learning algorithm fails. But this active learning algorithm succeeds if and only if $\mathcal{A}$ succeeds, given these responses to its queries; thus, the probability $\mathcal{A}$ succeeds is less than $3/4$, contradicting the assumption that it achieves query complexity $\mathrm{QC}_{\mathrm{CCQ}}(\epsilon, 1/4, \mathbb{C}, \mathcal{A}\mathrm{gnostic}(\mathbb{C}, \eta))$. ■

3.2. Upper bound

In this section, we describe an algorithm whose query complexity is $\tilde{O}\left(kd\frac{\beta^2}{\epsilon^2}\right)$. For clarity, we start by considering in the case where we know an upper bound β on η. We will discuss how to remove the assumption of knowing an upper bound β on η, adapting to η, in Section 3.2. Our main procedure (Algorithm 1) has two phases: in Phase 1, it uses a robust version of the classic halving algorithm to produce a classifier whose error rate is at most $10(\beta + \epsilon)$ by only using $\tilde{O}\left(kd \log \frac{1}{\epsilon}\right)$ queries. In Phase 2, we run a simple refining algorithm that uses $\tilde{O}\left(kd\frac{\beta^2}{\epsilon^2}\right)$ queries to turn the classifier output in Phase 1 into a classifier of error $\eta + \epsilon$. To implement Phase 1, we use a robust version of the classic halving algorithm. The idea here is that rather than eliminating a hypothesis when making just one mistake (as in the classic halving algorithm), we will eliminate a hypothesis when it makes at least one mistake in some number out of several sets (of an appropriate size) chosen uniformly at random from the unlabeled pool. The key point is that if the set size is appropriate (say $1/(16\eta)$), then we will not eliminate the best hypothesis in the class since it does not make mistakes on too many sets. On the other hand, if the plurality vote function has a high error (at least 10η), then it will make mistakes on enough sets and we can show that this then implies that a constant fraction of the version space will make mistakes on more sets than the best classifier in the class does (so we will be able to eliminate a constant fraction of the version space).

We express these algorithms in terms of a useful subroutine (Subroutine 1, Find-Mistake), which identifies an example in a given set on which a given classifier makes a mistake. Also, given $V \subseteq \mathbb{C}$, define the plurality vote classifier as $\mathrm{plur}(V)(x) = \mathrm{argmax}_{y \in \mathcal{Y}} \sum_{h \in V} \mathbb{I}[h(x) = y]$. Also, for $\epsilon > 0$, we call a set $\mathcal{H}$ an ϵ-cover of $\mathbb{C}$ if, for every $h \in \mathbb{C}$, $\inf_{g \in \mathcal{H}} P_{X \sim \mathcal{D}}(g(X) \neq h(X)) < \epsilon$. An ϵ-cover is called "minimal" if it has minimal possible cardinality among all ϵ-covers. It is known that the size of a minimal ϵ-cover of a class $\mathbb{C}$ of Natarajan dimension d is at most $(ck^2/\epsilon)^d$ for an appropriate constant c van der Vaart and Wellner (1996); Haussler and Long (1995). Note that constructing an ϵ-cover only requires access to the distribution $\mathcal{D}$ of the *unlabeled* examples, and in particular, one can construct a cover of near-minimal size based on a sample of $\tilde{O}(d/\epsilon)$ random unlabeled examples. Below, for brevity, we simply suppose we have access to a minimal ϵ-cover; it is a simple exercise to extend these results to near-minimal covers constructed from random unlabeled examples.

Note that, if $\mathrm{err}_S(h) > 0$, then Find-Mistake returns a labeled example (x, y) with y the true label of x, such that $h(x) \neq y$, and otherwise it returns an indication that no such point exists.

Lemma 3 below characterizes the performance of Phase 1 and Lemma 4 characterizes the performance of Phase 2. Note that the budget parameter in these methods is only utilized in our later discussion of adaptation to the noise rate.

Subroutine 1 Find-Mistake

```
Input: The sequence $S = (x_1, x_2, \ldots, x_m)$; classifier $h$
1. For each $y \in \{1, \ldots, k\}$,
    (a) Query the set $\{x \in S : h(x) \neq y\}$ for label $y$
    (b) If received back an example $(x, y)$, return $(x, y)$
2. Return "none"
```

Algorithm 1 General Agnostic Interactive Algorithm

```
Input: The sequence $(x_1, x_2, ..., )$; values $u$, s, $\delta$; budget $n$ (optional; default value = $\infty$).
1. Let $V$ be a (minimal) $\epsilon$-cover of the space of classifiers $\mathbb{C}$ with respect to $\mathcal{D}_X$. Let $U$ be $\{x_1, ..., x_u\}$.
2. Run the Generalized Halving Algorithm (Phase 1) with input $U$; $V$, s, $c \ln \frac{4 \log_2 |V|}{\delta}$, $n/2$, and get $h$.
3. Run the Refining Algorithm (Phase 2) with input $U$, $h$, $n/2$, and get labeled sample $L$ returned.
4. Find an hypothesis $h' \in V$ of minimum $\mathrm{err}_L(h')$.
Output Hypothesis $h'$ (and $L$).
```

Phase 1 Generalized Halving Algorithm

```
Input: The sequence $U = (x_1, x_2, ..., x_{\mathrm{ps}})$; set of classifiers $V$; values s, $N$; budget $n$ ($n$ optional: default value = $\infty$).
1. Set $b = \mathrm{true}$, $t = 0$.
2. while ($b$ and $t \leq n - N$)
    (a) Draw $S_1$, $S_2$, ..., $S_N$ of size s uniformly without replacement from $U$.
    (b) For each $i$, call Find-Mistake with arguments $S_i$, and $\mathrm{plur}(V)$. If it returns a mistake, we record the mistake $(\tilde{x}_i, \tilde{y}_i)$ it returns.
    (c) If Find-Mistake finds a mistake in more than $N/3$ of the sets, remove from $V$ every $h \in V$ making mistakes on $> N/9$ examples $(\tilde{x}_i, \tilde{y}_i)$, and set $t \leftarrow t + N$; else $b \leftarrow 0$.
Output Hypothesis $\mathrm{plur}(V)$.
```

Phase 2 Refining Algorithm

```
Input: The sequence $U = (x_1, x_2, ..., x_{\mathrm{ps}})$; classifier $h$; budget $n$ ($n$ optional: default value = $\infty$).
1. Set $b = 1$, $t = 0$, $W = U$, $L = \emptyset$.
2. while ($b$ and $t < n$)
    (a) Call Find-Mistake with arguments $W$, and $h$.
    (b) If it returns a mistake $(\tilde{x}, \tilde{y})$, then set $L \leftarrow L \cup \{(\tilde{x}, \tilde{y})\}$, $W \leftarrow W \setminus \{\tilde{x}\}$, and $t \leftarrow t + 1$.
    (c) Else set $b = 0$ and $L \leftarrow L \cup \{(x, h(x)) : x \in W\}$.
Output Labeled sample $L$.
```

Lemma 3 *Assume that some $\hat{h} \in V$ has $\mathrm{err}_U(\hat{h}) \leq \beta$ for $\beta \in [0, 1/32]$. With probability $\geq 1-\delta/2$, running Phase 1 with U, and values $\mathrm{s} = \left\lfloor \frac{1}{16\beta} \right\rfloor$ and $N = c \ln \frac{4 \log_2 |V|}{\delta}$ (for an appropriate constant $c \in (0, \infty)$), we have that for every round of the loop of Step 2, the following hold.*

- *$\hat{h}$ makes mistakes on at most $N/9$ of the returned $(\tilde{x}_i, \tilde{y}_i)$ examples.*
- *If $\mathrm{err}_U(\mathrm{plur}(V)) \geq 10\beta$, then Find-Mistake returns a mistake for $\mathrm{plur}(V)$ on $> N/3$ of the sets.*
- *If Find-Mistake returns a mistake for $\mathrm{plur}(V)$ on $> N/3$ of the sets S_i, then the number of h in V making mistakes on $> N/9$ of the returned $(\tilde{x}_i, \tilde{y}_i)$ examples in Step 3(b) is at least $(1/4)|V|$.*

Proof Sketch: Phase 1 and Lemma 3 are inspired by the analysis of Hanneke (2007b). In the following, by a *noisy* example we mean any x_i such that $\hat{h}(x_i) \neq y_i$. The expected number of noisy points in any given set S_i is at most $1/16$, which (by Markov's inequality) implies the probability S_i contains a noisy point is at most $1/16$. Therefore, the expected number of sets S_i with a noisy point in them is at most $N/16$, so by a Chernoff bound, with probability at least $1 - \delta/(4 \log_2 |V|)$ we have that at most $N/9$ sets S_i contain any noisy point, establishing claim 1.

Assume that $\mathrm{err}_U(\mathrm{plur}(V)) \geq 10\beta$. The probability that there is a point $\tilde{x}_i$ in S_i such that $\mathrm{plur}(V)$ labels $\tilde{x}_i$ differently from $\tilde{y}_i$ is $\geq 1-(1-10\beta)^{\mathrm{s}} \geq .37$ (discovered by direct optimization). So (for an appropriate value of $c > 0$ in N) by a Chernoff bound, with probability at least $1 - \delta/(4 \log_2 |V|)$, at least $N/3$ of the sets S_i contain a point $\tilde{x}_i$ such that $\mathrm{plur}(V)(\tilde{x}_i) \neq \tilde{y}_i$, which establishes claim 2. Via a combinatorial argument, this then implies with probability at least $1 - \delta/(4 \log_2 |V|)$, at least $|V|/4$ of the hypotheses make mistakes on more than $N/9$ of the sets S_i.

A union bound over the above two events, as well as over the iterations of the loop (of which there are at most $\log_2 |V|$ due to the third claim) obtains the claimed overall $1 - \delta/2$ probability.

Lemma 4 *Suppose some $\hat{h}$ has $\mathrm{err}_U(\hat{h}) \leq \beta$, for some $\beta \in [0, 1/32]$. Running Phase 2 with parameters U, $\hat{h}$, and any budget n, if L is the returned sample, and $|L| = |U|$, then every $(x_i, y) \in L$ has $y = y_i$ (i.e., the labels are in agreement with the oracle's labels); furthermore, $|L| = |U|$ definitely happens for any $n \geq \beta|U| + 1$.*

Proof Sketch: Every call to Find-Mistake returns a new mistake for $\hat{h}$ from U, except the last call, and since there are only $\beta|U|$ such mistakes, the procedure requires only $\beta|U| + 1$ calls to Find-Mistake. Furthermore, every label was either given to us by the oracle, or was assigned at the end, and in this latter case the oracle has certified that they are correct.

We are now ready to present our main upper bounds for the agnostic noise model.

Theorem 5 *Suppose $\beta \geq \eta$, and $\beta + \epsilon \leq 1/32$. Running Algorithm 1 on the data sequence $x_1, x_2, \ldots$, with parameters $u = O(d((\beta + \epsilon)/\epsilon^2) \log(k/\epsilon\delta))$, $\mathrm{s} = \left\lfloor \frac{1}{16(\beta+\epsilon)} \right\rfloor$, and δ, with probability at least $1 - \delta$ it produces a classifier h' with $\mathrm{err}(h') \leq \eta + \epsilon$ using a number of queries $O\left(kd \frac{\beta^2}{\epsilon^2} \log \frac{1}{\epsilon\delta} + kd \log \frac{\log(1/\epsilon)}{\delta} \log \frac{1}{\epsilon}\right)$.*

Proof Sketch: We have chosen u large enough so that $\mathrm{err}_U(h^*) \leq \eta + \epsilon \leq \beta + \epsilon$, with probability at least $1 - \delta/4$, by a (multiplicative) Chernoff bound. By Lemma 3, we know that with probability $1-\delta/2$, h^* is never discarded in Step 2(c) in Phase 1, and as long as $\mathrm{err}_U(\mathrm{plur}(V)) \geq 10(\beta+\epsilon)$, then we cut the set $|V|$ by a constant factor. So, with probability $1 - 3\delta/4$, after at most $O(kN \log(|V|))$ queries, Phase 1 halts with the guarantee that $\mathrm{err}_U(\mathrm{plur}(V)) \leq 10(\beta + \epsilon)$. Thus, by Lemma 4, the execution of Phase 2 returns a set L with the true labels after at most $(10(\beta+\epsilon)u+1)k$ queries. Therefore, due to the aforementioned bound on the size of a minimal ϵ-cover, by Chernoff and union bounds, we have chosen u large enough so that the h' of minimal $\mathrm{err}_U(h')$ has $\mathrm{err}(h') \leq \eta + \epsilon$ with probability at least $1-\delta/4$. Combining the above events by a union bound, with probability $1-\delta$, the h' chosen at the conclusion of Algorithm 1 has $\mathrm{err}(h') \leq \eta + \epsilon$ and the total number of queries is at most $kN \log_{4/3}(|V|) + k(10(\beta+\epsilon)u+1)$, which is bounded by the claimed value.

In particular, if we take $\beta = \eta$, Theorem 5 implies the upper bound part of Theorem 2.

Note: It is sometimes desirable to restrict the size of the sample we make the query for, so that the oracle does not need to sort through an extremely large sample searching for a mistake. To this end, we can run Phase 2 on chunks of size $1/(\eta+\epsilon)$ from U, and then union the resulting labeled samples to form L. The number of queries required for this is still bounded by the desired quantity.

Note: We note that if $\eta = \Omega(\epsilon^{2/3})$, then we could replace the first phase with a much simpler method, such as running empirical risk minimization on a labeled sample of size $\tilde{O}(d/\eta)$, while still producing a classifier h with a similar $\mathrm{err}(h) = O(\eta)$ guarantee, which would then be suitable to use in the second phase; indeed, this would allow us to avoid the use of the ϵ-cover V, which can often be exponentially large in d. However, when $\eta \ll \epsilon^{2/3}$, the bound in Theorem 5 will generally be smaller than $\tilde{O}(d/\eta)$, so that the additional complexity of using our robust halving technique is warranted by an improved query complexity. Moreover, in the special case where we are only interested in finding a classifier h with $\mathrm{err}(h) = O(\eta)$, the query complexity bound in Theorem 5 is merely $\tilde{O}(kd\log(1/\eta))$, which is preferable to the sample complexity $\tilde{O}(d/\eta)$ for passive learning.

In practice, knowledge of an upper bound β reasonably close to η is typically not available. As such, it is important to design algorithms that adapt to the unknown value of η. The following theorem indicates this is possible in our setting, without significant loss in query complexity.

Theorem 6 *There exists an algorithm that is independent of η and $\forall \eta \in [0, 1/2)$ achieves query complexity* $\mathrm{QC}_{\mathrm{CCQ}}(\epsilon, \delta, \mathbb{C}, \mathcal{A}\mathrm{gnostic}(\mathbb{C}, \eta)) = \tilde{O}\left(kd\frac{\eta^2}{\epsilon^2}\right)$.

Proof Sketch: First, note that if we set the budget parameter n large enough (at roughly $1/k$ times the value of the query complexity bound of Theorem 2), then the largest value of β for which the algorithm (with parameters as in Theorem 5) produces L with $|L| = u$ has $\beta \geq \eta$, so that it produces h' with $\mathrm{err}(h') \leq \eta + \epsilon$. So for a given budget n, we can simply run the algorithm for each β value in a log-scale grid of $[\epsilon, 1]$, and take the h' for the largest such β with $|L| = u$ (if n is large enough that such a β exists). The second part of the problem then becomes determining an appropriately large budget n, so that this works. For this, we can simply search for such a value by a guess-and-double technique, where for each n we check whether it is large enough by evaluating a standard confidence bound on the excess error rate; the key that allows this to work is that, if $|L| = u$, then L is an iid $\mathcal{D}_{XY}$-distributed sequence of labeled examples, so that we can use known confidence bounds for working with iid labeled data.

4. Bounded Noise

In this section we study the *Bounded noise* model (also known as Massart noise), which has been extensively studied in the learning theory literature (Massart and Nedelec, 2006; Gine and Koltchinskii, 2006; Hanneke, 2011). This model represents a significantly stronger restriction on the type of noise. The motivation for bounded noise is that, in some scenarios, we do have an accurate representation of the target function within our hypothesis class (i.e., the model is correctly specified), but we allow for nature's labels to be slightly randomized. Formally, the we consider the family $\mathrm{BN}(\mathbb{C}, \alpha) = \{\mathcal{D}_{XY} : \exists h^* \in \mathbb{C} \text{ s.t. } \mathbb{P}_{\mathcal{D}_{XY}}(Y \neq h^*(X)|X) \leq \alpha\}$, for $\alpha \in [0, 1/2)$. We are sometimes interested in the special case of Random Classification Noise, defined as $\mathrm{RCN}(\mathbb{C}, \alpha) = \{\mathcal{D}_{XY} : \exists h^* \in \mathbb{C} \text{ s.t. } \forall \ell \neq h^*(x), \mathbb{P}_{\mathcal{D}_{XY}}(Y = \ell | X = x) = \alpha/(k-1)\}$. Also define $\mathrm{BN}(\mathbb{C}, \alpha; \mathcal{D}_X)$ and $\mathrm{RCN}(\mathbb{C}, \alpha; \mathcal{D}_X)$ as those $\mathcal{D}_{XY}$ in these respective classes with marginal $\mathcal{D}_X$ on $\mathcal{X}$.

In this section we show a lower bound on the query complexity of interactive learning with class-conditional queries as a function of the query complexity of active learning (label request queries). The proof follows via a reduction from the (multiclass) active learning model (label request queries) to our interactive learning model (general class-conditional queries), very similar in spirit to the reduction given in the proof of the lower bound in Theorem 2.

Theorem 7 *Consider any hypothesis class $\mathbb{C}$ of Natarajan dimension $d \in (0, \infty)$. For any $\alpha \in [0, 1/2)$, and any distribution $\mathcal{D}_X$ over $\mathcal{X}$, in the random classification noise model we have the following relationship between the query complexity of interactive learning in the class-conditional queries model and the the query complexity of active learning with label requests:*

$$\tfrac{\alpha}{2(k-1)} \mathrm{QC}_{\mathrm{AL}}(\epsilon, 2\delta, \mathbb{C}, \mathrm{RCN}(\mathbb{C}, \alpha; \mathcal{D}_X)) - 4 \ln\left(\tfrac{1}{\delta}\right) \leq \mathrm{QC}_{\mathrm{CCQ}}(\epsilon, \delta, \mathbb{C}, \mathrm{RCN}(\mathbb{C}, \alpha; \mathcal{D}_X))$$

To complement this lower bound, we prove a related upper bound via an analysis of an algorithm below, which operates by reducing to a kind of batch-based active learning algorithm. Specifically, assume we have an active learning algorithm $\mathcal{A}$ that proceeds in rounds, and in each round it interacts with an oracle by providing a region R of the instance space and a number m and and it expects in return m labeled examples from the conditional distribution given that x is in R. For example the A^2 algorithm Balcan et al. (2006) and the algorithm of Koltchinskii (2010) can be written to operate this way. We show in the following how we can use our algorithms from Section 3 in order to provide the desired labeled examples to such an active learning procedure while using fewer than m queries to our oracle. In the description below we assume that algorithm $\mathcal{A}$ returns its state, a region R of the instance space, a number m of desired samples, a boolean flag b for halting ($b = 0$) or not ($b = 1$), and a classifier h.

The value δ' in this algorithm should be set appropriately depending on the context, essentially as δ divided by a coarse bound on the total number of batches the algorithm $\mathcal{A}$ will request the labels of; for our purposes a value $\delta' = \mathrm{poly}(\epsilon\delta(1-2\alpha)/d)$ will suffice. To state an explicit bound on the number of queries, we first review the following definition of Hanneke (2007a, 2009). For $r > 0$, define $B(h, r) = \{g \in \mathbb{C} : \mathbb{P}_{\mathcal{D}_X}(h(X) \neq g(X)) \leq r\}$. For any $\mathcal{H} \subseteq \mathbb{C}$, define the region of disagreement: $\mathrm{DIS}(\mathcal{H}) = \{x \in \mathcal{X} : \exists h, g \in \mathcal{H} \text{ s.t. } h(x) \neq g(x)\}$. Define the *disagreement coefficient* for $h \in \mathbb{C}$: $\theta_h(\epsilon) = \sup_{r > \epsilon} \mathbb{P}_{\mathcal{D}_X}(\mathrm{DIS}(B(h, r)))/r$. Define the disagreement coefficient of the class $\mathbb{C}$ as $\theta(\epsilon) = \sup_{h \in \mathbb{C}} \theta_h(\epsilon)$.

Theorem 8 *For $\mathbb{C}$ of Natarajan dimension d, and $\alpha \in [0, 1/2)$, for any distribution $\mathcal{D}_X$ over $\mathcal{X}$,*

$$\mathrm{QC}_{\mathrm{CCQ}}(\epsilon, \delta, \mathbb{C}, \mathrm{BN}(\mathbb{C}, \alpha; \mathcal{D}_X)) = O\left(\left(1 + \tfrac{\alpha\theta(\epsilon)}{(1-2\alpha)^2}\right) dk \log^2\left(\tfrac{dk}{\epsilon\delta(1-2\alpha)}\right)\right).$$

The significance of this result is that $\theta(\epsilon)$ is multiplied by α, a feature not present in the known results for active learning. In a sense, this factor of $\theta(\epsilon)$ is a measure of how difficult the active learning problem is, as the other terms are inevitable (up to the log factors).

Algorithm 2 General Interactive Algorithm for Bounded Noise

```
Input: The sequence $(x_1, x_2, ...,)$; allowed error rate $\epsilon$, noise bound $\alpha$, algorithm $\mathcal{A}$.
1. Set $b = 1$, $t = 1$. Initialize $\mathcal{A}$ and let $\mathcal{S}(\mathcal{A})$, $R$, $m$, $b$ and $\hat{h}$ be the returned values.
2. Let $V$ be a minimal $\epsilon$-cover of $\mathbb{C}$ with respect to the distribution $\mathcal{D}_X$.
3. While $(b)$
   (a) Let $ps = \frac{cd}{\epsilon^2} \log \frac{k}{\epsilon\delta}$ and let $(x_{i_1}, x_{i_2}, \ldots, x_{i_{ps+m}})$ be the first $ps + m$ points in $(x_{t+1}, x_{t+2}, \ldots) \cap R$.
   (b) Run Phase 1 with parameters $\mathcal{U}_1 = (x_{i_1}, x_{i_2}, \ldots, x_{i_{ps}})$, $V$, $\left\lfloor \frac{1}{16(\alpha+\epsilon)} \right\rfloor$, $c \log \frac{4\log_2|V|}{\delta'}$
       Let $h$ be the returned classifier.
   (c) Run Phase 2 with parameters $\mathcal{U}_2 = (x_{i_{ps+1}}, x_{i_{ps+2}}, \ldots, x_{i_{ps+m}})$, $h$.
       Let $L$ be the returned labeled sequence.
   (d) Run $\mathcal{A}$ with parameters $L$ and $\mathcal{S}(\mathcal{A})$. Let $\mathcal{S}(\mathcal{A})$, $R$, $m$, $b$ and $\hat{h}$ be the returned values.
   (e) Let $t = i_{ps+m}$
Output Hypothesis $\hat{h}$.
```

By the same reasoning as in the above proof, plugging in a different kind of active learning algorithm $\mathcal{A}$ (which space limitations prevent description of here), one can prove an analogous bound based on the splitting index of Dasgupta (2005), rather than the disagreement coefficient. This is interesting, in that one can also prove a lower bound on $\mathrm{QC}_{\mathrm{AL}}$ in terms of the splitting index, so that composed with Theorem 7, we have a nearly tight characterization of $\mathrm{QC}_{\mathrm{CCQ}}(\epsilon, \delta, \mathcal{D}, \mathrm{BN}(\mathbb{C}, \alpha; \mathcal{D}_X))$. See Appendix C.2.

As before, since the value of the noise bound α is typically not known in practice, it is often desirable to have an algorithm capable of *adapting* to the value of α, while maintaining the query complexity guarantees of Algorithm 2. Fortunately, we can achieve this by a similar argument to that used above in Theorem 6. That is, starting with an initial guess of $\hat{\alpha} = \epsilon$ as the noise bound argument to Algorithm 2, we use the budget argument to Phase 2 to guarantee we never exceed the query complexity bound of Theorem 8 (with $\hat{\alpha}$ in place of α), halting early if ever Phase 2 fails to label the entire $\mathcal{U}_1$ set within its query budget. Then we repeatedly double $\hat{\alpha}$ until finally this modified Algorithm 2 runs to completion. Setting the budget sizes and δ' values appropriately, we can maintain the guarantee of Theorem 8 with only an extra log factor increase.

5. Discussion and Open Questions

A concrete open question is determining the query complexity of class conditional and mistake queries under Tsybakov noise. Another concrete open question is providing computationally efficient procedures that the meet a nearly optimal query complexity for such queries in the presence of certain types of noise. While our analysis provides an upper bound on query complexity for general classes of queries, it is not clear that we have yet identified the appropriate quantities to appear in a tight analysis in the query complexity in a general case.

Acknowledgments We thank Vladimir Koltchinskii, Pranjal Awasthi, and Frank Dellaert for useful discussions. This work was supported in part by NSF grants CCF-0953192 and CCF-1101215, AFOSR grant FA9550-09-1-0538, an MSR Faculty Fellowship, and a Google Research Award.

References

D. Angluin. Queries and concept learning. *Machine Learning*, 2:319–342, 1998.

P. Auer and R. Ortner. A new PAC bound for intersection-closed concept classes. *Machine Learning*, 66: 151–163, 2007.

M. F. Balcan, A. Beygelzimer, and J. Langford. Agnostic active learning. In *ICML*, 2006.

M.-F. Balcan, A. Broder, and T. Zhang. Margin based active learning. In *COLT*, 2007.

M.-F. Balcan, S. Hanneke, and J. Wortman. The true sample complexity of active learning. In *COLT*, 2008.

J. L. Balcázar, J. Castro, and D. Guijarro. A general dimension for exact learning. In *Proceedings of the* 14^{th} *Conference on Learning Theory*, 2001.

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

S. Ben-David, N. Cesa-Bianchi, D. Haussler, and P. M. Long. Characterizations of Learnability for Classes of {0, ..., n}-Valued Functions. *J. Comput. Syst. Sci.*, 1995.

A. Beygelzimer, S. Dasgupta, and J. Langford. Importance weighted active learning. In *Proceedings of the 26th International Conference on Machine Learning (ICML)*, 2009.

A. Beygelzimer, D. Hsu, J. Langford, and T. Zhang. Agnostic active learning without constraints. In *NIPS*, 2010.

R. Castro and R. Nowak. Minimax bounds for active learning. In *Proceedings of the 20th Annual Conference on Computational Learning Theory (COLT)*, 2007.

E. Chang, S. Tong, K. Goh, and C.-W. Chang. Support vector machine concept-dependent active learning for image retrieval. *IEEE Transactions on Multimedia*, 2005.

S. Dasgupta. Coarse sample complexity bounds for active learning. In *NIPS*, volume 18, 2005.

S. Dasgupta, A. Kalai, and C. Monteleoni. Analysis of perceptron-based active learning. In *COLT*, 2005.

S. Dasgupta, D.J. Hsu, and C. Monteleoni. A general agnostic active learning algorithm. *Advances in Neural Information Processing Systems*, 20, 2007.

S. Doyle, J. Monaco, M. Feldman, J. Tomaszewski, and A. Madabhushi. A class balanced active learning scheme that accounts for minority class problems: Applications to histopathology. In *MICCAI Workshop on Optical Tissue Image Analysis in Microsopy, Histopathology and Endoscopy*, 2009.

Y. Freund, H.S. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28(2-3):133–168, 1997.

E. Gine and V. Koltchinskii. Concentration inequalities and asymptotic results for ratio type empirical processes. *The Annals of Probability*, 34(3):1143–1216, 2006.

S. Hanneke. A bound on the label complexity of agnostic active learning. In *ICML*, 2007a.

S. Hanneke. Teaching dimension and the complexity of active learning. In *Proceedings of the 20th Annual Conference on Computational Learning Theory (COLT)*, 2007b.

S. Hanneke. *Theoretical Foundations of Active Learning*. PhD thesis, Machine Learning Department, School of Computer Science, Carnegie Mellon University, 2009.

S. Hanneke. Rates of convergence in active learning. *The Annals of Statistics*, 39(1):333–361, 2011.

D. Haussler and P. M. Long. A generalization of sauer's lemma. *Journal of Combinatorial Theory, Series A*, 71:219–240, 1995.

T. Hegedüs. Generalized teaching dimension and the query complexity of learning. In *The 8*$^{\text{th}}$ *Annual Conference on Computational Learning Theory*, 1995.

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

V. Koltchinskii. Rademacher complexities and bounding the excess risk in active learning. *Journal of Machine Learning*, 11:2457–2485, 2010.

N. Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 1988.

P. Massart and E. Nedelec. Risk bounds for statistical learning. *The Annals of Statistics*, 34(5):2326–2366, 2006.

A. McCallum and K. Nigam. Employing EM in pool-based active learning for text classification. In *Proceedings of the 15th International Conference on Machine Learning (ICML)*, pages 350–358, 1998.

B. K. Natarajan. On learning sets and functions. *Machine Learning*, 4:67–97, 1989.

H. Simon. PAC-learning in the presence of one-sided classification noise. In *International Symposium on Artificial Intelligence and Mathematics*, 2012.

S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 4:45–66, 2001.

A. W. van der Vaart and J. A. Wellner. *Weak Convergence and Empirical Processes*. Springer, 1996.

V. N. Vapnik. *Statistical Learning Theory*. John Wiley and Sons, 1998.

L. Wang. Sufficient conditions for agnostic active learnable. In *NIPS*, 2009.

Appendix A. Useful Facts

Lemma 9 *Let* $B_1, \ldots, B_k$ *be independent* Geometric(α) *random variables. With probability at least* $1 - \delta$*,*

$$\sum_{i=1}^{k} B_i \leq \frac{2}{\alpha}\left(k + 4\ln\left(\frac{1}{\delta}\right)\right).$$

Proof Let $m = \frac{2}{\alpha}\left(k + 4\ln\left(\frac{1}{\delta}\right)\right)$. Let $X_1, X_2, \ldots$ be i.i.d. Bernoulli(α) random variables. $\sum_{i=1}^{k} B_i$ is distributionally equivalent to a value N defined as the smallest value of n for which $\sum_{i=1}^{n} X_i = k$, so it suffices to show $\mathbb{P}(N \leq m) \geq 1 - \delta$.

Let $H = \sum_{i=1}^{m} X_i$. We have $\mathbb{E}[H] = \alpha m \geq 2k$. By a Chernoff bound, we have

$$\mathbb{P}\left(H \leq k\right) \leq \mathbb{P}\left(H \leq (1/2)\mathbb{E}[H]\right) \leq \exp\left\{-\mathbb{E}[H]/8\right\} \leq \exp\left\{-\ln\left(\frac{1}{\delta}\right)\right\} = \delta.$$

Therefore, with probability $1-\delta$, we have $N \leq m$, as claimed. ■

The following is a direct consequence of a result of Vapnik (1998) (except substituting the appropriate quantities for the multiclass case).

Lemma 10 *For L a finite sequence of i.i.d. $\mathcal{D}_{XY}$ labeled examples, and any $\delta \in (0,1)$, with probability at least $1-\delta$, for all $h \in \mathbb{C}$,*

$$\left|\left(\mathrm{err}_L(h) - \min_{g \in \mathbb{C}} \mathrm{err}_L(g)\right) - \left(\mathrm{err}(h) - \mathrm{err}(h^*)\right)\right| \leq \frac{8d}{|L|}\ln\left(\frac{3|L|}{\delta}\right) + \sqrt{\mathrm{err}_L(h)\frac{16d}{|L|}\ln\left(\frac{3|L|}{\delta}\right)}.$$

This follows from the fact that

$$|\mathrm{err}(h) - \mathrm{err}_L(h)| \leq O\left(\frac{d}{|L|}\ln\left(\frac{|L|}{\delta}\right) + \sqrt{\mathrm{err}(h)\frac{d}{|L|}\ln\left(\frac{|L|}{\delta}\right)}\right),$$

which in particular also implies that the sample complexity of passive learning (by empirical risk minimization) is at most $\tilde{O}\left(d\frac{\eta+\epsilon}{\epsilon^2}\right)$.

Appendix B. Class Conditional Queries. The Agnostic Case

Lemma 3 Assume that some $\hat{h} \in V$ has $\mathrm{err}_U(\hat{h}) \leq \beta$ for $\beta \in [0, 1/32]$. With probability $\geq 1-\delta/2$, running Phase 1 with U, and values $\mathrm{s} = \left\lfloor \frac{1}{16\beta} \right\rfloor$ and $N = c\ln\frac{4\log_2|V|}{\delta}$ (for an appropriate constant $c \in (0,\infty)$), we have that for every round of the loop of Step 2, the following hold.

- $\hat{h}$ makes mistakes on at most $N/9$ of the returned $(\tilde{x}_i, \tilde{y}_i)$ examples.
- If $\mathrm{err}_U(\mathrm{plur}(V)) \geq 10\beta$, then Find-Mistake returns a mistake for $\mathrm{plur}(V)$ on $> N/3$ of the sets.
- If Find-Mistake returns a mistake for $\mathrm{plur}(V)$ on $> N/3$ of the sets S_i, then the number of h in V making mistakes on $> N/9$ of the returned $(\tilde{x}_i, \tilde{y}_i)$ examples in Step 3(b) is at least $(1/4)|V|$.

Proof Phase 1 and Lemma 3 are inspired by the analysis of Hanneke (2007b). In the following, by a *noisy* example we mean any x_i such that $\hat{h}(x_i) \neq y_i$. The expected number of noisy points in any given set S_i is at most $1/16$, which (by Markov's inequality) implies the probability S_i contains a noisy point is at most $1/16$. Therefore, the expected number of sets S_i with a noisy point in them is at most $N/16$, so by a Chernoff bound, with probability at least $1-\delta/(4\log_2|V|)$ we have that at most $N/9$ sets S_i contain any noisy point, establishing claim 1.

Assume that $\mathrm{err}_U(\mathrm{plur}(V)) \geq 10\beta$. The probability that there is a point $\tilde{x}_i$ in S_i such that $\mathrm{plur}(V)$ labels $\tilde{x}_i$ differently from $\tilde{y}_i$ is $\geq 1-(1-10\beta)^{\mathrm{s}} \geq .37$ (discovered by direct optimization). So (for an appropriate value of $c > 0$ in N) by a Chernoff bound, with probability at least $1 - \delta/(4\log_2|V|)$, at least $N/3$ of the sets S_i contain a point $\tilde{x}_i$ such that $\mathrm{plur}(V)(\tilde{x}_i) \neq \tilde{y}_i$, which establishes claim 2. Via a combinatorial argument, this then implies with probability at least $1 - \delta/(4\log_2|V|)$, at least $|V|/4$ of the hypotheses will make mistakes on more than $N/9$ of the sets S_i. To see this consider the bipartite graph where on the left hand side we have all the classifiers in V and on the right hand side we have all the returned $(\tilde{x}_i, \tilde{y}_i)$ examples. Let us put an edge between a node i on the left and a node j on the right if the hypothesis h_i associated to node i makes a mistake on $(\tilde{x}_i, \tilde{y}_i)$. Let M be the number of vertices in the right hand side. Clearly, the total number of edges in the graph is at least $(1/2)|V||M|$, since at most $|V|/2$ classifiers label $\tilde{x}_i$ as $\tilde{y}_i$. Let $\alpha|V|$ be the number of classifiers in V that make mistakes on at most $N/9$ $(\tilde{x}_i, \tilde{y}_i)$ examples. The total number of edges in the graph is then upper bounded by $\alpha|V|N/9 + (1-\alpha)|V|M$. Therefore,

$$(1/2)|V||M| \leq \alpha|V|N/9 + (1-\alpha)|V|M,$$

which implies

$$|V||M|(\alpha - 1/2) \leq \alpha|V|N/9.$$

Applying the lower bound $M \geq N/3$, we get $(N/3)|V|(\alpha - 1/2) \leq \alpha|V|N/9$, so $\alpha \leq 3/4$. This establishes claim 3.

A union bound over the above two events, as well as over the iterations of the loop (of which there are at most $\log_2|V|$ due to the third claim of this lemma) obtains the claimed overall $1 - \delta/2$ probability. ■

Lemma 4 Suppose some $\hat{h}$ has $\mathrm{err}_U(\hat{h}) \leq \beta$, for some $\beta \in [0, 1/32]$. Running Phase 2 with parameters U, $\hat{h}$, and any budget n, if L is the returned sample, and $|L| = |U|$, then every $(x_i, y) \in L$ has $y = y_i$ (i.e., the labels are in agreement with the oracle's labels); furthermore, $|L| = |U|$ definitely happens for any $n \geq \beta|U| + 1$.

Proof Every call to Find-Mistake returns a new mistake for $\hat{h}$ from U, except the last call, and since there are only $\beta|U|$ such mistakes, the procedure requires only $\beta|U| + 1$ calls to Find-Mistake. Furthermore, every label was either given to us by the oracle, or was assigned at the end, and in this latter case the oracle has certified that they are correct.

Formally, if $|L| = |U|$, then either every $x \in U$ was returned as some $(\tilde{x}, \tilde{y})$ pair in Step 2.b, or we reached Step 2.c. In the former case, these $\tilde{y}$ labels are the oracle's actual responses, and thus correspond to the true labels. In the latter case, every element of L added prior to reaching 2.c was returned by the oracle, and is therefore the true label. Every element $(x_i, y) \in L$ added in Step 2.c has label $\hat{h}(x_i)$, which the oracle has just told us is correct in Find-Mistake (meaning we definitely have $\hat{h}(x_i) = y_i$). Thus, in either case, the labels are in agreement with the true labels. Finally, note that each call to Find-Mistake either returns a mistake for $\hat{h}$ we have not previously received, or is the final such call. Since there are at most $\beta|U|$ mistakes in total, we can have at most $\beta|U| + 1$ calls to Find-Mistake. ■

Theorem 5 Suppose $\beta \geq \eta$, and $\beta + \epsilon \leq 1/32$. Running Algorithm 1 with parameters $u = O(d((\beta+\epsilon)/\epsilon^2)\log(k/\epsilon\delta))$, $\mathrm{s} = \left\lfloor \frac{1}{16(\beta+\epsilon)} \right\rfloor$, and δ, with probability at least $1-\delta$ it produces a classifier h' with $\mathrm{err}(h') \leq \eta + \epsilon$ using a number of queries $O\left(kd\frac{\beta^2}{\epsilon^2}\log\frac{1}{\epsilon\delta} + kd\log\frac{\log(1/\epsilon)}{\delta}\log\frac{1}{\epsilon}\right)$.

Proof We have chosen u large enough so that $\mathrm{err}_U(h^*) \leq \eta + \epsilon \leq \beta + \epsilon$, with probability at least $1 - \delta/4$, by a (multiplicative) Chernoff bound. By Lemma 3, we know that with probability $1 - \delta/2$,

h^* is never discarded in Step 2(c) in Phase 1, and as long as $\mathrm{err}_U(\mathrm{plur}(V)) \geq 10(\beta + \epsilon)$, then we cut the set $|V|$ by a constant factor. So, with probability $1 - 3\delta/4$, after at most $O(kN \log(|V|))$ queries, Phase 1 halts with the guarantee that $\mathrm{err}_U(\mathrm{plur}(V)) \leq 10(\beta + \epsilon)$. Thus, by Lemma 4, the execution of Phase 2 returns a set L with the true labels after at most $(10(\beta + \epsilon)u + 1)k$ queries.

Furthermore, we can choose the ϵ-cover V so that $|V| \leq 4(ck^2/\epsilon)^d$ for an appropriate constant c (van der Vaart and Wellner, 1996; Haussler and Long, 1995).

Therefore, by Chernoff and union bounds, we have chosen u large enough so that the h' of minimal $\mathrm{err}_U(h')$ has $\mathrm{err}(h') \leq \eta + \epsilon$ with probability at least $1 - \delta/4$. Combining the above events by a union bound, with probability $1 - \delta$, the h' chosen at the conclusion of Algorithm 1 has $\mathrm{err}(h') \leq \eta + \epsilon$ and the total number of queries is at most

$$kN \log_{4/3}(|V|) + k(10(\beta + \epsilon)u + 1) = O\left(kd \log \frac{d \log(k/\epsilon)}{\delta} \log \frac{1}{\epsilon} + kd \frac{(\beta + \epsilon)^2}{\epsilon^2} \log \frac{k}{\epsilon\delta}\right).$$

■

Theorem 6 There exists an algorithm that is independent of η and $\forall \eta \in [0, 1/2)$ achieves query complexity $\mathrm{QC}_{\mathrm{CCQ}}(\epsilon, \delta, \mathbb{C}, \mathcal{A}\mathrm{gnostic}(\mathbb{C}, \alpha)) = \tilde{O}\left(kd\frac{\eta^2}{\epsilon^2}\right)$.

Proof We consider the proof of this theorem in two stages, with the following intuitive motivation. First, note that if we set the budget parameter n large enough (at roughly $1/k$ times the value of the query complexity bound of Theorem 2), then the largest value of β for which the algorithm (with parameters as in Theorem 5) produces L with $|L| = u$ has $\beta \geq \eta$, so that it produces h' with $\mathrm{err}(h') \leq \eta + \epsilon$. So for a given budget n, we can simply run the algorithm for each β value in a log-scale grid of $[\epsilon, 1]$, and take the h' for the largest such β with $|L| = u$. The second part of the problem then becomes determining an appropriately large budget n, so that this works. For this, we can simply search for such a value by a guess-and-double technique, where for each n we check whether it is large enough by evaluating a standard confidence bound on the excess error rate; the key that allows this to work is that, if $|L| = u$, then the set L is an i.i.d. $\mathcal{D}_{XY}$-distributed sequence of labeled examples, so that we can use known confidence bounds for working with sequences of random labeled examples. The details of this strategy follow.

Consider values $n_j = 2^j$ for $j \in \mathbb{N}$, and define the following procedure. We can consider a sequence of values $\eta_i = 2^{1-i}$ for $i \leq \log_2(1/\epsilon)$. For each $i = 1, 2, \ldots, \log_2(1/\epsilon)$, we run Algorithm 1 with parameters

$$u = u_i = O(d((\eta_i + \epsilon)/\epsilon^2) \log(k/\epsilon\delta)),$$

$$\mathrm{s} = \mathrm{s}_i = \frac{1}{16(\eta_i + \epsilon)}, \quad \delta_i = \delta/(8 \log_2(1/\epsilon))$$

and budget parameter $n_j/\log_2(1/\epsilon)$. Let h_{ji} and L_{ji} denote the return values from this execution of Algorithm 1, and let $\hat{h}_j$ and $\hat{L}_j$ denote the values h_{ji} and L_{ji}, respectively, for the smallest value of i for which $|L_{ji}| = u_i$ (if such an i exists): that is, for which the execution of Phase 2 ran to completion.

Note that for some j with $n_j = O\left(d\frac{\eta^2}{\epsilon^2} \log \frac{k \log_2(1/\epsilon)}{\epsilon\delta} + d \log \frac{\log^2(1/\epsilon)}{\delta} \log \frac{k}{\epsilon}\right) \log_2 \frac{1}{\epsilon}$, Theorem 5 implies that with probability $1 - \delta/4$, every $i \leq \lfloor \log_2(1/\eta) \rfloor$ with $|L_{ji}| = u_i$ has $\mathrm{err}(h_{ji}) \leq \eta + \epsilon/2$, and $|L_{ji}| = u_i$ for at least one such i value: namely, $i = \lfloor \log_2(1/\max\{\eta, \epsilon\}) \rfloor$. Thus, $\mathrm{err}(\hat{h}_j) \leq$

$\eta + \epsilon/2$ for this value of j. Let j^* denote this value of j, and for the remainder of this subsection we suppose this high-probability event occurs.

All that remains is to design a procedure for searching over n_j values to find one large enough to obtain this error rate guarantee, but not so large as to lose the query complexity guarantee. Toward this end, define

$$\mathcal{E}_j = \frac{8d}{|\hat{L}_j|} \ln\left(\frac{12|\hat{L}_j|j^2}{\delta}\right) + \sqrt{\mathrm{err}_{\hat{L}_j}(\hat{h}_j)\frac{16d}{|\hat{L}_j|} \ln\left(\frac{12|\hat{L}_j|j^2}{\delta}\right)}.$$

Lemma 10 implies that with probability at least $1 - \delta/2$, $\forall j$ for which $\hat{L}_j$ and $\hat{h}_j$ are defined,

$$\left|\left(\mathrm{err}_{\hat{L}_j}(\hat{h}_j) - \min_{h \in \mathbb{C}} \mathrm{err}_{\hat{L}_j}(h)\right) - \left(\mathrm{err}(\hat{h}_j) - \mathrm{err}(h^*)\right)\right| \leq \mathcal{E}_j.$$

Consider running the above procedure for $j = 1, 2, 3, \ldots$ in increasing order until we reach the first value of j for which $\hat{L}_j$ and $\hat{h}_j$ are defined, and

$$\mathrm{err}_{\hat{L}_j}(\hat{h}_j) - \min_{h \in \mathbb{C}} \mathrm{err}_{\hat{L}_j}(h) + \mathcal{E}_j \leq \epsilon.$$

Denote this first value of j as $\hat{j}$. Note that choosing $\hat{j}$ in this way guarantees $\mathrm{err}(\hat{h}_{\hat{j}}) \leq \eta + \epsilon$.

It remains only to bound the value of this $\hat{j}$, so that we may add up the total number of queries among the executions of our procedure for all values $j \leq \hat{j}$. By setting the constants in u_i appropriately, the sample size of $|\hat{L}_j|$ is large enough so that, for $j = j^*$, a Chernoff bound (to bound $\mathrm{err}_{\hat{L}_j}(h^*) \geq \mathrm{err}_{\hat{L}_j}(\hat{h}_j)$) guarantees that with probability $1 - \delta/4$, $\mathcal{E}_j \leq \epsilon/4$. Furthermore, we have

$$\mathrm{err}_{\hat{L}_j}(\hat{h}_j) - \min_{h \in \mathbb{C}} \mathrm{err}_{\hat{L}_j}(h) \leq \mathrm{err}(\hat{h}_j) - \mathrm{err}(h^*) + \mathcal{E}_j \leq \epsilon/2 + \epsilon/4 = (3/4)\epsilon,$$

so that in total $\mathrm{err}_{\hat{L}_j}(\hat{h}_j) - \min_{h \in \mathbb{C}} \mathrm{err}_{\hat{L}_j}(h) + \mathcal{E}_j \leq (3/4)\epsilon + \epsilon/4 = \epsilon$. Thus, we have $\hat{j} \leq j^*$, so that the total number of queries is less than $2n_{j^*}$.

Therefore, by a union bound over the above events, with probability $1 - \delta$, the selected $\hat{h}_{\hat{j}}$ has $\mathrm{err}(\hat{h}_{\hat{j}}) \leq \eta + \epsilon$, and the total number of queries is less than

$$2kn_{j^*} = O\left(dk\frac{\eta^2}{\epsilon^2} \log\frac{\log(1/\epsilon)}{\epsilon\delta} \log\frac{1}{\epsilon} + dk \log\frac{\log(1/\epsilon)}{\delta} \log^2\frac{1}{\epsilon}\right).$$

Thus, not having direct access to the noise rate only increases our query complexity by at most a logarithmic factor compared to the bound of Theorem 2. ■

Appendix C. Class Conditional Queries. Bounded Noise

Theorem 7 Consider any hypothesis class $\mathbb{C}$ of Natarajan dimension $d \in (0, \infty)$. For any $\alpha \in [0, 1/2)$, and any distribution $\mathcal{D}_X$ over $\mathcal{X}$, in the random classification noise model we have the following relationship between the query complexity of interactive learning in the class-conditional queries model and the the query complexity of active learning with label requests:

$$\frac{\alpha}{2(k-1)}\mathrm{QC}_{\mathrm{AL}}(\epsilon, 2\delta, \mathbb{C}, \mathrm{RCN}(\mathbb{C}, \alpha; \mathcal{D}_X)) - 4\ln\frac{1}{\delta} \leq \mathrm{QC}_{\mathrm{CCQ}}(\epsilon, \delta, \mathbb{C}, \mathrm{RCN}(\mathbb{C}, \alpha; \mathcal{D}_X))$$

Proof The proof follows via a reduction from the active learning model (label request queries) to our interactive learning model (general class-conditional queries). Assume that we have an algorithm that works for the CCQ model with query complexity $\mathrm{QC}_{\mathrm{CCQ}}(\epsilon, \delta, \mathbb{C}, \mathrm{RCN}(\mathbb{C}, \alpha; \mathcal{D}_X))$. We can convert this into an algorithm that works in the active learning model with a query complexity of $\mathrm{QC}_{\mathrm{AL}}(\epsilon, 2\delta, \mathbb{C}, \mathrm{RCN}(\mathbb{C}, \alpha; \mathcal{D}_X)) = \frac{2(k-1)}{\alpha}[\mathrm{QC}_{\mathrm{CCQ}}(\epsilon, \delta, \mathbb{C}, \mathrm{RCN}(\mathbb{C}, \alpha; \mathcal{D}_X)) + 4\ln\frac{1}{\delta}]$, as follows. When our CCQ algorithm queries the i^{th} time, say querying for a label y among a set S_i, we pick an example $x_{i,1}$ at random in S_i and (if the label of $x_{i,1}$ has never previously been requested), we request its label $y_{i,1}$. If $y = y_{i,1}$, then we return $(x_{i,1}, y_{i,1})$ to the algorithm, and otherwise we keep taking examples $(x_{i,2}, x_{i,3}, \ldots)$ at random in the set S_i and (if their label has not yet been requested) requesting their labels $(y_{i,2}, y_{i,3}, \ldots)$, until we find one with label y, at which point we return this labeled example to the algorithm. If we exhaust S_i and we find example of label y, we return to the algorithm that there are no examples in S_i with label y.

Let A_i be a random variable indicating the actual number of label requests we make in round i before getting either an example of label y or exhausting the set S_i. We also define a related random variable B_i as follows. For $j \leq A_i$, if $h^*(x_{i,j}) \neq y$, let $Z_j = I[y_{i,j} = y]$, and if $h^*(x_{i,j}) = y$, let C_j be an independent Bernoulli$((\alpha/(k-1))/(1-\alpha))$ random variable, and let $Z_j = C_j I[y_{i,j} = y]$. For $j > A_i$, let Z_j be an independent Bernoulli$(\alpha/(k-1))$ random variable. Let $B_i = \min\{j : Z_j = 1\}$. Since, $\forall j \leq A_i$, $Z_j \leq I[y_{i,j} = y]$, we clearly have $B_i \geq A_i$. Furthermore, note that the Z_j are independent Bernoulli$(\alpha/(k-1))$ random variables, so that B_i is a Geometric$(\alpha/(k-1))$ random variable. By Lemma 9 in Appendix A, we obtain that with probability at least $1-\delta$ we have

$$\sum_i A_i \leq \sum_i B_i \leq \frac{2(k-1)}{\alpha}[\mathrm{QC}_{\mathrm{CCQ}}(\epsilon, \delta, \mathbb{C}, \mathrm{RCN}(\mathbb{C}, \alpha; \mathcal{D}_X)) + 4\ln\frac{1}{\delta}].$$

This then implies

$$\mathrm{QC}_{\mathrm{AL}}(\epsilon, 2\delta, \mathbb{C}, \mathrm{RCN}(\mathbb{C}, \alpha; \mathcal{D}_X)) \leq \frac{2(k-1)}{\alpha}[\mathrm{QC}_{\mathrm{CCQ}}(\epsilon, \delta, \mathbb{C}, \mathrm{RCN}(\mathbb{C}, \alpha; \mathcal{D}_X)) + 4\ln\frac{1}{\delta}],$$

which implies the desired result. ■

Theorem 8 For any concept space $\mathbb{C}$ of Natarajan dimension d, and any $\alpha \in [0, 1/2)$, for any distribution $\mathcal{D}_X$ over $\mathcal{X}$,

$$\mathrm{QC}_{\mathrm{CCQ}}(\epsilon, \delta, \mathbb{C}, \mathrm{BN}(\mathbb{C}, \alpha; \mathcal{D}_X)) = O\left(\left(1 + \frac{\alpha\theta(\epsilon)}{(1-2\alpha)^2}\right) dk \log^2\left(\frac{dk}{\epsilon\delta(1-2\alpha)}\right)\right).$$

Proof We show that, for $\mathcal{D}_{XY} \in \mathrm{BN}(\mathbb{C}, \alpha)$, running Algorithm 2 with the algorithm $\mathcal{A}$ as the method from (Koltchinskii, 2010) returns a classifier $\hat{h}$ with $\mathrm{err}(\hat{h}) \leq \eta + \epsilon$ using a number of queries as in the claim.

For bounded noise, with noise bound α, on each round of Algorithm 2, we run Algorithm 1 on a set $\mathcal{U}_1$ that, by Hoeffding's inequality and the size of ps, with probability $1 - \delta / \log(1/\epsilon)$, has

$$\min_{h \in V} \mathrm{err}_{\mathcal{U}_1}(h) \leq \alpha + \epsilon.$$

Thus, by Lemma 3, the fraction of examples in each $\mathcal{U}_1 = (x_{i_1}, \ldots, x_{i_{ps}})$ on which the returned h makes a mistake is at most $10(\alpha + \epsilon)$. Then the size of ps and Hoeffding's inequality implies that $\mathrm{err}(h) \leq O(\alpha + \epsilon)$ with probability $1 - \delta / \log(1/\epsilon)$, and a Chernoff bound implies that Algorithm 2 is run on a set $\mathcal{U}_2$ with

$$\mathrm{err}_{\mathcal{U}_2}(h) \leq O(\alpha + \epsilon + \sqrt{(\alpha + \epsilon) \log(\log(1/\epsilon)/\delta)/m} + \log(\log(1/\epsilon)/\delta)/m).$$

Thus, by Lemmas 3 and 4, the number of queries per round is

$$O(k(\alpha + \epsilon)m + k\sqrt{(\alpha + \epsilon)m \log(\log(1/\epsilon)/\delta)} + kd \log(d/\epsilon\delta(1 - 2\alpha))).$$

In particular, for the algorithm of Koltchinskii (2010), it is known that with probability $1 - \delta/2$, every round has $m \leq O\left(\frac{\theta(\epsilon)d}{(1-2\alpha)^2} \log\left(\frac{1}{\epsilon\delta(1-2\alpha)}\right)\right)$, and there are at most $O(\log(1/\epsilon))$ rounds, so that the total number of queries is at most $O\left(k\left(\alpha\theta(\epsilon) + 1\right) \frac{d}{(1-2\alpha)^2} \log^2\left(\frac{d}{\epsilon\delta(1-2\alpha)}\right)\right)$. ■

C.1. Adapting to Unknown α

Algorithm 2 is based on having direct access to the noise bound α. As in Section 3.2, since this information is not typically available in practice, we would prefer a method that can obtain essentially the same query complexity bounds without direct access to α. Fortunately, we can achieve this by a similar argument to Section 3.2, merely by doubling our guess at the value of α until the algorithm behaves as expected, as follows.

Consider modifying Algorithm 2 as follows. In Step 6, we include the budget argument to Algorithm 2, with value $O((1 + \alpha m) \log(1/\delta'))$. Then, if the set L returned has $|L| < m$, we return Failure. Note that if this α is at least as large as the actual noise bound, then this bound is inconsequential, as it will be satisfied anyway (with probability $1 - \delta'$, by a Chernoff bound). Call this modified method Algorithm $2'$.

Now consider the sequences $\alpha_i = 2^{i-1}\epsilon$, for $1 \leq i \leq \log_2(1/\epsilon)$. For $i = 1, 2, \ldots, \log_2(1/\epsilon)$ in increasing order, we run Algorithm $2'$ with parameters $(x_1, x_2, \ldots)$, ϵ, α_i, $\mathcal{A}$. If the algorithm runs to completion, we halt and output the $\hat{h}$ returned by Algorithm $2'$. Otherwise, if the algorithm returns Failure, we increment i and repeat.

Since Algorithm $2'$ runs to completion for any $i \geq \lceil \log(\alpha/\epsilon) \rceil$, and since the number of queries Algorithm $2'$ makes is monotonic in its α argument, for an appropriate choice of $\delta' = O(\delta\epsilon^2/d)$ (based on a coarse bound on the total number of batches the algorithm will request labels for), we have a total number of queries at most $O\left((1 + \alpha\theta(\epsilon))\frac{d}{(1-2\alpha)^2} \log^2\left(\frac{d}{\epsilon\delta(1-2\alpha)}\right) \log\left(\frac{1}{\epsilon}\right)\right)$ for the method of Koltchinskii (2010), only a $O(\log(1/\epsilon))$ factor over the bound of Theorem 8; similarly, we lose at most a factor of $O(\log(1/\epsilon))$ for the splitting method, compared to the bound of Theorem 14.

C.2. Bounds Based on the Splitting Index

By the same reasoning as in the proof of Theorem 8, except running Algorithm 2 with Algorithm 3 instead, one can prove an analogous bound based on the splitting index of Dasgupta (2005), rather than the disagreement coefficient. This is interesting, in that one can also prove a lower bound on $\mathrm{QC}_{\mathrm{AL}}$ in terms of the splitting index, so that composed with Theorem 7, we have a nearly tight characterization of $\mathrm{QC}_{\mathrm{CCQ}}(\epsilon, \delta, \mathcal{D}, \mathrm{BN}(\mathbb{C}, \alpha; \mathcal{D}_X))$. Specifically, consider the following definitions due to Dasgupta (2005).

Let $Q \subseteq \{\{h, g\} : h, g \in \mathbb{C}\}$ be a finite set of unordered pairs of classifiers from $\mathbb{C}$. For $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, define $Q_x^y = \{\{h, g\} \in Q : h(x) = g(x) = y\}$. A point $x \in \mathcal{X}$ is said to *ρ-split* Q if

$$\max_{y \in \mathcal{Y}} |Q_x^y| \leq (1 - \rho)|Q|.$$

Fix any distribution $\mathcal{D}_X$ on $\mathcal{X}$. We say $\mathcal{H} \subseteq \mathbb{C}$ is (ρ, Δ, τ)-*splittable* if for all finite $Q \subseteq \{\{h, g\} \subseteq \mathbb{C} : \mathbb{P}_{\mathcal{D}_X}(x : h(x) \neq g(x)) > \Delta\}$,

$$\mathbb{P}_{\mathcal{D}_X}(x : x \ \rho\text{-splits } Q) \geq \tau.$$

A large value of ρ for a reasonably large τ indicates that there are highly informative examples that are not too rare. Following Dasgupta (2005), for each $h \in \mathbb{C}$, $\tau > 0$, $\epsilon > 0$, we define

$$\rho_{h,\tau}(\epsilon) = \sup\{\rho : \forall \Delta \geq \epsilon/2, B(h, 4\Delta) \text{ is } (\rho, \Delta, \tau)\text{-splittable}\}.$$

Here, $B(h, r) = \{g \in \mathbb{C} : \mathbb{P}_{\mathcal{D}_X}(x : h(x) \neq g(x)) \leq r\}$ for $r > 0$. Though Dasgupta (2005) explores results on the query complexity as a function of h^*, $\mathcal{D}_X$, for our purposes (minimax analysis) we will take a worst-case value of ρ. That is, define

$$\rho_\tau(\epsilon) = \inf_{h \in \mathbb{C}} \rho_{h,\tau}(\epsilon).$$

Theorem 7 (in the main body) relates the query complexity of CCQ to that of AL. There is much known about the latter, and in the interest of stating a concrete particularly tight result here, we provide a new particularly tight result, inspired by the analysis of Dasgupta (2005). For simplicity, we will only discuss the $k = 2$ case in this section.

Theorem 11 *Suppose $k = 2$. There exist universal constants $c_1, c_2 \in (0, \infty)$ such that, for any concept space $\mathbb{C}$ of VC dimension d, any $\alpha \in [0, 1/2)$, $\epsilon, \delta \in (0, 1/16)$, and distribution $\mathcal{D}_X$ over $\mathcal{X}$,*

$$\inf_{\tau > 0} \frac{c_1}{\rho_\tau(4\epsilon)} \leq \mathrm{QC}_{\mathrm{AL}}(\epsilon, \delta, \mathbb{C}, \mathrm{BN}(\mathbb{C}, \alpha; \mathcal{D}_X)) \leq \inf_{\tau > 0} \frac{c_2 d^3}{(1 - 2\alpha)^2 \rho_\tau(\epsilon)} \log^5\left(\frac{1}{\epsilon\delta\tau(1 - 2\alpha)}\right).$$

The proof of Theorem11 is included in Section C.2.1. The implication of the lower bound given by Theorem 7, combined with Theorem 11 is as follows.

Corollary 12 *Suppose $k = 2$. There exists a universal constant $c \in (0, \infty)$ such that, for any concept space $\mathbb{C}$ of Natarajan dimension d, any $\alpha \in [0, 1/2)$, $\epsilon, \delta \in (0, 1/32)$, and distribution $\mathcal{D}_X$ over $\mathcal{X}$,*

$$\mathrm{QC}_{\mathrm{CCQ}}(\epsilon, \delta, \mathbb{C}, \mathrm{BN}(\mathbb{C}, \alpha; \mathcal{D}_X)) \geq \frac{\alpha}{2} \cdot \inf_{\tau > 0} \frac{c}{\rho_\tau(4\epsilon)} - 4\ln(4).$$

In particular, this means that in some cases, the query complexity of CCQ learning is only smaller by a factor proportional to α compared to the number of random labeled examples required by passive learning, as indicated by the following example, which follows immediately from Corollary 12 and Dasgupta's analysis of the splitting index for interval classifiers (Dasgupta, 2005).

Corollary 13 *For $\mathcal{X} = [0,1]$ and $\mathbb{C} = \{2\mathbb{I}_{[a,b]} - 1 : a, b \in [0,1]\}$ the class of* interval *classifiers, there is a constant $c \in (0,1)$ such that, for any $\alpha \in [0,1/2)$ and sufficiently small $\epsilon > 0$,*

$$\mathrm{QC}_{\mathrm{CCQ}}(\epsilon, 1/32, \mathbb{C}, \mathrm{BN}(\mathbb{C},\alpha)) \geq c\frac{\alpha}{\epsilon}.$$

There is also a near-matching upper bound compared to Corollary 12. That is, running Algorithm 2 with Algorithm 3 of Appendix C.2.1, we have the following result in terms of the splitting index.

Theorem 14 *Suppose $k = 2$. For any concept space $\mathbb{C}$ of VC dimension d, and any $\alpha \in [0,1/2)$, for any distribution $\mathcal{D}_X$ over $\mathcal{X}$,*

$$\begin{aligned}&\mathrm{QC}_{\mathrm{CCQ}}(\epsilon, \delta, \mathbb{C}, \mathrm{BN}(\mathbb{C},\alpha;\mathcal{D}_X)) \\ &\qquad = O\left(d\log^2\left(\frac{d}{\epsilon\delta(1-2\alpha)}\right) + \inf_{\tau>0}\frac{\alpha d^3}{(1-2\alpha)^2\rho_\tau(\epsilon)}\log^5\left(\frac{1}{\epsilon\delta\tau(1-2\alpha)}\right)\right).\end{aligned}$$

Logarithmic factors and terms unrelated to ϵ and α aside, in spirit the combination of Corollary 12 with Theorem 14 imply that in the bounded noise model, the specific reduction in query complexity of using class-conditional queries instead of label request queries is essentially a factor of α.

C.2.1. PROOF OF THEOREM 11

We prove Theorem 11 in two parts. First, we establish the lower bound. The technique for this is quite similar to a result of Dasgupta (2005). Recall that $\mathrm{QC}_{\mathrm{AL}}(\epsilon, \delta, \mathbb{C}, \mathfrak{Realizable}(\mathbb{C};\mathcal{D}_X)) \leq \mathrm{QC}_{\mathrm{AL}}(\epsilon, \delta, \mathbb{C}, \mathrm{BN}(\mathbb{C},\alpha;\mathcal{D}_X))$. Thus, the following lemma implies the lower bound of Theorem 11.

Lemma 15 *For any hypothesis class $\mathbb{C}$ of Natarajan dimension d, for any distribution $\mathcal{D}_X$ over $\mathcal{X}$,*

$$\mathrm{QC}_{\mathrm{AL}}(\epsilon, 1/16, \mathbb{C}, \mathfrak{Realizable}(\mathbb{C};\mathcal{D}_X)) \geq \inf_{\tau>0}\frac{c}{\rho_\tau(4\epsilon)}.$$

Proof The proof is quite similar to that of a related result of Dasgupta (2005). Fix any $\tau \in (0,1/4)$, and suppose $\mathcal{A}$ is an active learning algorithm that considers at most the first $1/(4\tau)$ unlabeled examples, with probability greater than $7/8$. Let $h \in \mathbb{C}$ be such that $\rho_{h,\tau}(4\epsilon) \leq 2\rho_\tau(4\epsilon)$, and let $\Delta \geq 2\epsilon$ and $Q \subseteq \{\{f,g\} \subseteq B(h,4\Delta) : \mathbb{P}_{\mathcal{D}_X}(x : f(x) \neq g(x)) > \Delta\}$ be such that $\mathbb{P}_{\mathcal{D}_X}(x : x\ 2\rho_{h,\tau}(4\epsilon)\text{-splits } Q) < \tau$. In particular, with probability at least $(1-\tau)^{1/(4\tau)} \geq 3/4$, none of the first $1/(4\tau)$ unlabeled examples $2\rho_{h,\tau}(4\epsilon)$-splits Q. Fix any such data set, and denote $\rho = 2\rho_{h,\tau}(4\epsilon)$.

We proceed by the probabilistic method. We randomly select the target h^* as follows. First, choose a pair $\{f^*, g^*\} \in Q$ uniformly at random. Then choose h^* from among $\{f^*, g^*\}$ uniformly at random.

For each unlabeled example x among the first $1/(4\tau)$, call the label y with $|Q_x^y| > (1-\rho)|Q|$ the "bad" response. Given the initial $1/(4\tau)$ unlabeled examples, the algorithm $\mathcal{A}$ has some fixed (a priori known, though possibly randomized) behavior when the responses to all of its label requests are the bad responses. That is, it makes some number t of queries, and then returns some classifier $\hat{h}$.

For any one of those label requests, the probability that both f^* and g^* agree with the bad response is greater than $1-\rho$. Thus, by a union bound, the probability both f^* and g^* agree with the bad responses for the t queries of the algorithm is greater than $1-t\rho$. On this event, the algorithm returns $\hat{h}$, which is independent from the random choice of h^* from among f^* and g^*. Since $\mathbb{P}_{\mathcal{D}_X}(x : f^*(x) \neq g^*(x)) > \Delta \geq 2\epsilon$, $\hat{h}$ can be ϵ-close to at most one of them, so that there is at least a $1/2$ probability that $\operatorname{err}(\hat{h}) > \epsilon$.

Adding up the failure probabilities, by a union bound the probability the algorithm's returned classifier h' has $\operatorname{err}(h') > \epsilon$ is greater than $7/8 - 1/4 - t\rho - 1/2$. For any $t < 1/(16\rho)$, this is greater than $1/16$. Thus, there exists some deterministic $h^* \in \mathbb{C}$ for which $\mathcal{A}$ requires at least $1/(16\rho)$ queries, with probability greater than $1/16$.

As any active learning algorithm has a $7/8$-confidence upper bound M on the number of unlabeled examples it uses, letting $\tau \to 0$ in the above analysis allows $M \to \infty$, and thus covers all possible active learning algorithms. ■

We will establish the upper bound portion of Lemma 11 via the following algorithm. Here we write the algorithm in a closed form, but it is clear that we could rewrite the method in the batch-based style required by Algorithm 2 above, simply by including its state every time it makes a batch of label request queries. The value ϵ_0 in this method should be set appropriately for the result below; specifically, we will coarsely take $\epsilon_0 = O((1-2\alpha)^3\epsilon^2\tau^2\delta^2/d^3)$, based on the analysis of Dasgupta (2005) for the realizable case.

We have the following result for this method, with an appropriate setting of the constants in the "$O(\cdot)$" terms.

Lemma 16 *Suppose $k = 2$. There exists a constant $c \in (0,\infty)$ such that, for any hypothesis class $\mathbb{C}$ of VC dimension d, for any $\alpha \in [0, 1/2)$ and $\tau > 0$, for any distribution $\mathcal{D}_X$ over $\mathcal{X}$, for any $\mathcal{D}_{XY} \in \mathrm{BN}(\mathbb{C}, \alpha; \mathcal{D}_X)$, Algorithm 3 produces a classifier $\hat{h}$ with $\operatorname{err}(\hat{h}) \leq \eta + \epsilon$ using a number of label request queries at most*

$$O\left(\frac{d^3}{(1-2\alpha)^2 \rho_{h^*,\tau}(\epsilon)} \log^5\left(\frac{1}{(1-2\alpha)\epsilon\delta\tau}\right)\right).$$

Proof [Sketch] Since V is initially an ϵ_0-cover, the $\hat{h} \in V$ of minimal $\operatorname{err}(\hat{h})$ has $\operatorname{err}(\hat{h}) \leq \epsilon_0$. Furthermore, ϵ_0 was chosen so that, as long as the total number of unlabeled examples processed does not exceed $O(\frac{d^3}{(1-2\alpha)^3\epsilon^2\tau^2\delta})$, with probability $1 - O(\delta)$, we will have $\hat{h}$ agreeing with h^* on all of the unlabeled examples, and in particular on all of the examples whose labels the algorithm requests. This means that, for every example x we request the label of, $\mathbb{P}(\hat{h}(x) = y|x) \geq 1-\alpha$. By Chernoff and union bounds, with probability $1 - O(\delta)$, for every $g \in V$, we always have

$$M_{\hat{h}g} - M_{g\hat{h}} \leq O\left(\sqrt{\max\{M_{hg}, M_{gh}\} d \log\left(\frac{1}{\epsilon_0}\right)} + d \log\left(\frac{1}{\epsilon_0}\right)\right),$$

Algorithm 3 An active learning algorithm for learning with bounded noise, based on splitting.

```
Input: The sequence $U = (x_1, x_2, ...)$; allowed error rate $\epsilon$; value $\tau \in (0, 1)$; noise bound $\alpha \in [0, 1/2)$.

I. Let $V$ denote a minimal $\epsilon_0$-cover of $\mathbb{C}$

II. For each pair of classifier $h, g \in V$, initialize $M_{hg} = 0$

III. For $T = 1, 2, \ldots, \lceil \log_2(4/\epsilon) \rceil$
  1. Consider the set $Q \subseteq V^2$ of pairs $\{h, g\} \subseteq V$ with $\mathbb{P}_{\mathcal{D}_X}(x : h(x) \neq g(x)) > 2^{-T}$
  2. While ($|Q| > 0$)
     (a) Let $S = \emptyset$
     (b) Do $O\left(\frac{1}{(1-2\alpha)^2}\left(d \log\left(\frac{1}{\epsilon}\right) + \log\left(\frac{1}{\delta}\right)\right)\right)$ times
           i. Let $\tilde{Q} = Q$
          ii. While ($|\tilde{Q}| > 0$)
                A. From among the next $1/\tau$ unlabeled examples, select the one $\tilde{x}$ with minimum
                   $\max_{y \in \mathcal{Y}} |\tilde{Q}_x^y|$, and let $\tilde{y}$ denote the maximizing label
                B. $S \leftarrow S \cup \{\tilde{x}\}$
                C. $\tilde{Q} \leftarrow \tilde{Q}_{\tilde{x}}^{\tilde{y}}$
     (c) Request the labels for all examples in $S$, and let $L$ be the resulting labeled examples
     (d) For each $h, g \in V$, let $M_{hg} \leftarrow M_{hg} + |\{(x, y) \in L : h(x) \neq y = g(x)\}|$
     (e) Let $V \leftarrow \left\{h \in V : \forall g \in V, M_{hg} - M_{gh} \leq O\left(\sqrt{\max\{M_{hg}, M_{gh}\} d \log\left(\frac{1}{\epsilon_0}\right)} + d \log\left(\frac{1}{\epsilon_0}\right)\right)\right\}$
     (f) Let $Q \leftarrow \{\{h, g\} \in Q : h, g \in V\}$

Output Any hypothesis $h \in V$.
```

so that we never remove $\hat{h}$ from V. Thus, for each round T, the set $V \subseteq B(h^*, 4\Delta_T)$, where $\Delta_T = 2^{-T}$. In particular, this means the returned h is in $B(h^*, \epsilon)$, so that $\operatorname{err}(h) \leq \eta + \epsilon$.

Also by Chernoff and union bounds, with probability $1 - O(\delta)$, any $g \in V$ with $M_{\hat{h}g} + M_{g\hat{h}} > O\left(\frac{d}{(1-2\alpha)^2} \log \frac{1}{\epsilon_0}\right)$ has

$$M_{g\hat{h}} - M_{\hat{h}g} > O\left(\sqrt{\max\{M_{hg}, M_{gh}\} d \log\left(\frac{1}{\epsilon_0}\right)} + d \log\left(\frac{1}{\epsilon_0}\right)\right),$$

so that we remove it from V at the end of the round.

That $V \subseteq B(h^*, 4\Delta_T)$ also means V is (ρ, Δ_T, τ)-splittable, for $\rho = \rho_{h^*,\tau}(\epsilon)$. In particular, this means we get a ρ-splitting example for $\tilde{Q}$ every $\frac{1}{\tau}$ examples (in expectation). Thus, we always satisfy the $|\tilde{Q}| = 0$ condition after at most $O\left(\frac{d}{\rho} \log^2 \frac{1}{\epsilon_0}\right)$ rounds of the inner loop (by Chernoff and union bounds, and the definition of ρ). Furthermore, among the examples added to S during this period, regardless of their true labels we are guaranteed that at least $1/2$ of pairs $\{h, g\}$ in Q have at least one of $(M_{h\hat{h}} + M_{\hat{h}h})$ or $(M_{g\hat{h}} + M_{\hat{h}g})$ incremented as a result: that is, for at least $|Q|/2$ pairs, at least one of the two classifiers disagrees with $\hat{h}$ on at least one of these examples. This is because, if the y labels used in the algorithm to prune the $\tilde{Q}$ set are the actual labels, then *every* pair in Q has this property, whereas if any of these y labels are *not* the actual label, then for the first such instance, all the pairs already eliminated from $\tilde{Q}$ have that property, while at least $1/2$ of those remaining also have that property (since that y value minimizes $|\tilde{Q}_x^y|$). Thus, after executing this $O\left(\frac{1}{(1-2\alpha)^2} d \log\left(\frac{1}{\epsilon_0}\right)\right)$ times, we are guaranteed that at least half of the $\{h_1, h_2\}$ pairs in Q have (for some $i \in \{1,2\}$) $M_{\hat{h}h_i} + M_{h_i\hat{h}} > O\left(\frac{d}{(1-2\alpha)^2} \log \frac{1}{\epsilon_0}\right)$, thus reducing $|Q|$ by at least a factor of 2. Repeating this $\log|Q| = O(d\log(1/\epsilon_0))$ times satisfies the $|Q| = 0$ condition.

Thus, the total number of queries is at most $O\left(\frac{1}{(1-2\alpha)^2} \frac{d^3}{\rho} \log^5 \frac{1}{\epsilon_0}\right)$, as desired. ■

Appendix D. One-sided noise

Consider the special case of binary classification (i.e., $k = 2$). In this case, the Natarajan dimension is simply the well-known VC dimension Vapnik (1998). In this context, the one-sided noise model Simon (2012) is a special subclass of $\mathrm{BN}(\mathbb{C}, \alpha)$ characterized by the property that only one of the two labels gets corrupted by noise. Specifically, let $\mathrm{OSN}(\mathbb{C}, \alpha)$ denote the set of joint distributions $\mathcal{D}_{XY}$ for which $\exists h^* \in \mathbb{C}$ such that for every $x \in \mathcal{X}$ with $h^*(x) = 1$, $\mathbb{P}_{\mathcal{D}_{XY}}(Y = 1|X = x) = 1$, while for every $x \in \mathcal{X}$ with $h^*(x) = 2$, $\mathbb{P}_{\mathcal{D}_{XY}}(Y = 2|X = x) = 1 - \alpha$. In this context, a hypothesis class $\mathbb{C}$ is called *intersection-closed* if, for every $h, g \in \mathbb{C}$, there exists $f \in \mathbb{C}$ such that $\{x : f(x) = 2\} = \{x : h(x) = 2\} \cap \{x : g(x) = 2\}$ Helmbold et al. (1990); Auer and Ortner (2007). In this context, we have the following result, the proof of which is included below. This result is particularly interesting, as it shows that it is sometimes possible to circumvent the lower bounds prove above and obtain close to the realizable-case query complexity, even with certain types of bounded noise.

Theorem 17 *If $k = 2$, then for any intersection-closed concept space $\mathbb{C}$ of VC dimension d, and any $\alpha \in [0, 1)$, $\mathrm{QC}_{\mathrm{CCQ}}(\epsilon, \delta, \mathbb{C}, \mathrm{OSN}(\mathbb{C}, \alpha)) = \tilde{O}\left((d + \log(1/\delta)) \log(1/\delta) \log(1/\epsilon)\right)$.*

In the case of intersection-closed spaces, there is one quite natural learning strategy, based on choosing the minimum consistent hypothesis, called the *closure*. Specifically, define the closure hypothesis $\hat{h}_m$ by the property that $\{x : \hat{h}_m(x) = 2\} = \bigcap_{h \in V_m^+} \{x : h(x) = 2\}$, where $V_m^+ = \{h \in \mathbb{C} : \forall i \leq m \text{ s.t. } y_i = 2, h(x_i) = 2\}$. The following lemma concerns the sample complexity of passive learning with intersection-closed concept classes under one-sided noise.

Lemma 18 *If $k = 2$ and $\mathbb{C}$ is intersection-closed of VC dimension d, for any $\alpha \in [0, 1)$, and any $\mathcal{D}_{XY} \in \mathrm{OSN}(\mathbb{C}, \alpha)$, for a universal constant $c \in (0, \infty)$, for any $m \in \mathbb{N}$, with probability at least $1 - \delta$, the closure hypothesis $\hat{h}_m$ satisfies $P_{\mathcal{D}_{XY}}(\hat{h}_m(X) \neq h^*(X)) \leq \frac{c(d\log(d) + \log(1/\delta))}{(1-\alpha)m}$.*

Loosely speaking, Lemma 18 says that after observing m samples, the closure hypothesis is roughly d/m-close to h^*. We can use this observation to derive a result for learning with class-conditional queries via the following reasoning. Suppose we are able to determine the closure hypothesis $\hat{h}_m$ for some value of $m \in \mathbb{N}$. Then consider repeatedly asking for examples labeled 2 in the set $\{x_i : m < i \leq m(1 + 1/d), \hat{h}_m(x_i) = 1\}$, removing each returned example before the next query for a 2 label among the remaining examples. After we exhaust all of the examples labeled

2 among these points, we have all the information we need to calculate the closure hypothesis $\hat{h}_{m(1+1/d)}$. Proceeding inductively in this manner, we can arrive at $\hat{h}_n$ for a value of n roughly $\tilde{O}(d/\epsilon)$ after roughly $d\log(1/\epsilon)$ rounds (supposing the initial value of m is d), at which point Lemma 18 indicates $\mathrm{err}(\hat{h}_n) - \mathrm{err}(h^*)$ is roughly ϵ. To bound the number of queries made on each of these $d\log(1/\epsilon)$ rounds, note that Lemma 18 indicates we expect roughly $O(1)$ examples labeled 2 among $\{x_i : m < i \leq m(1+1/d), \hat{h}_m(x_i) = 1\}$, so that each round makes only $O(1)$ queries, for a total of $O(d\log(1/\epsilon))$ queries. This informal reasoning leads to the following result, which is only slighly larger to account for needing these claims to hold with high probability $1-\delta$.

Proof [Theorem 17] Consider Algorithm 4 (where c is from Lemma 18).

Algorithm 4 Algorithm for learning intersection-closed $\mathbb{C}$ under one-sided noise

```
Input: The sequence $(x_1, x_2, \ldots)$
1. Set $m \leftarrow \lceil c(d\log(d) + \log(1/\delta')) \rceil$
2. Request labels $y_1, \ldots, y_m$ individually, and set $\hat{h} \leftarrow \hat{h}_m$, the closure hypothesis
3. While $m < (c/\epsilon)(d\log(d) + \log(1/\delta'))$
    (a) Let $m \leftarrow \left\lceil m\left(1 + \frac{1}{c(d\log(d)+\log(1/\delta'))}\right)\right\rceil$
    (b) Let $\mathcal{U} \leftarrow \{x_i : i \leq m, \hat{h}(x_i) = 1\}$, $\mathcal{L} \leftarrow \{(x_i, 2) : i \leq m, \hat{h}(x_i) = 2\}$
    (c) Do
        i. Query $\mathcal{U}$ for label 2
        ii. If we receive $(x_i, y_i)$ returned from the query, let $\mathcal{U} \leftarrow \mathcal{U} \setminus \{x_i\}$, $\mathcal{L} \leftarrow \mathcal{L} \cup \{(x_i, 2)\}$
        iii. Else let $\hat{h} \leftarrow \hat{h}_m$, the closure hypothesis (which can be determined based solely on $\mathcal{L}$), and
             break out of loop (c)
Output Hypothesis $\hat{h}$
```

At the conclusion, we have $m \geq \frac{c}{\epsilon}(d\log(d) + \log(1/\delta'))$, while the number of rounds of the outer loop is $O((d\log(d) + \log(1/\delta'))\log(1/\epsilon))$. Furthermore, the closure hypothesis $\hat{h}$ at the end of each round is precisely the same as that for the true labeled data set $\{(x_1, y_1), \ldots, (x_m, y_m)\}$. By Lemma 18, with probability at least $1-\delta'$, $P_{\mathcal{D}_{XY}}(\hat{h}(X) \neq h^*(X)) \leq (c/m(1-\alpha))(d\log(d) + \log(1/\delta'))$, so that $\mathrm{err}(\hat{h}) - \mathrm{err}(h^*) \leq (c/m)(d\log(d) + \log(1/\delta'))$. Thus, if this is the final round of the algorithm, this guarantees $\mathrm{err}(\hat{h}) - \mathrm{err}(h^*) \leq P_{\mathcal{D}_{XY}}(\hat{h}(X) \neq h^*(X))(1-\alpha) \leq \epsilon$ with probability at least $1-\delta'$.

It remains only to bound the number of queries. Note that the responses to queries are always points (x_i, y_i) for which $\hat{h}(x_i) \neq h^*(x_i)$ and $y_i = 2$. Thus, if this is not the final round of the algorithm, but $P_{\mathcal{D}_{XY}}(\hat{h}(X) \neq h^*(X)) \leq (c/m(1-\alpha))(d\log(d) + \log(1/\delta'))$, then a Chernoff bound implies that with probability at least $1-\delta'$, the number of queries on the next round is at most $O(\log(1/\delta'))$.

We reach the final round of the algorithm after at most $c(d\log(d) + \log(1/\delta'))\log(1/\epsilon)$ rounds. So with probability at least $1 - \delta' 2c(d\log(d) + \log(1/\delta'))\log(1/\epsilon)$, the total number of queries is at most $O\left((d\log(d) + \log(1/\delta'))\log(1/\epsilon)\log(1/\delta')\right)$. Taking $\delta' = \frac{\delta}{4c(d\log(d)+\log(d\log(1/\epsilon)/\delta))\log(1/\epsilon)}$, we have that with probability at least $1-\delta$, the final $\hat{h}$ has $\mathrm{err}(\hat{h}) - \mathrm{err}(h^*) \leq \epsilon$, and the total number of queries is at most $O\left((d\log(d) + \log(d\log(1/\epsilon)/\delta))\log(1/\epsilon)\log(d\log(1/\epsilon)/\delta)\right)$. ■

Since the closure hypothesis can be computed efficiently for many intersection-closed spaces, such as intervals, conjunctions, and axis-aligned rectangles, Algorithm 4 can also be made efficient in these cases.

Appendix E. Other types of queries

Though the results of this paper above are all formulated for class conditional queries, similar arguments can be used to study the query complexity of other types of queries as well. For instance, as is evident from the fact that our methods interact with the oracle only via the Find-Mistake subroutine, all of the results in this work also apply (up to a factor of k) to a kind of sample-based *equivalence query* (or mistake query), in which we provide a sample of unlabeled examples to the oracle along with a classifier h, and the oracle returns an instance in the sample on which h makes a mistake, if one exists. However, many of the techniques and results also apply to a much broader family of queries. In much the same spirit as the general dimensions explored in quantifying the query complexity in the Exact Learning setting, we can work in our present setting with an abstract family of queries, and characterize the query complexity in terms of an abstract combinatorial complexity measure. The resulting query complexity bounds relate the complexity of learning to a measure of the complexity of teaching or verification. The formal details of this abstract characterization are provided below.

E.1. IA and AI dimensions

To present our results on this abstract setting, we adopt the notation of Hanneke (2009), which derives from earlier works in the Exact Learning literature Balcázar et al. (2002, 2001). A *query* is a function q mapping a function f to a nonempty collection of sets of functions $q(f)$ such that $\forall a \in q(f)$, $f \in a$, and $\forall g \in a$, we have $a \in q(g)$. We interpret the set $q(f)$ as the set of *valid answers* the teacher can give to the query q when the target function is f, and for each such answer $a \in q(f)$, we interpret the functions $g \in a$ as precisely those functions *consistent* with the answer a: that is, those functions g for which the teacher could have validly answered the query q in this way had g been the target function. Further define an *oracle* as any function T mapping a query q to a set of functions $T(q) \in \bigcup_f q(f)$; we denote by $\mathcal{T}^f$ the set of oracles T such that every query q has $T(q) \in q(f)$: that is, the oracle's answers are always consistent with f. We also overload this notation, defining for Q a *set* of queries, $T(Q) = \bigcap_{q \in Q} T(q)$.

For any $m \in \mathbb{N}$ and $\mathcal{U} \in \mathcal{X}^m$, define the set of *data-dependent queries* $\mathcal{Q}^{**}_{\mathcal{U}}$ to be those queries q such that, for any functions f and g with $f(x) = g(x)$ for every $x \in \mathcal{U}$, we have $q(f) = q(g)$. This corresponds to the set of queries *about* the labels of the examples in $\mathcal{U}$.

In the present work, we study a further restriction on the allowed types of queries. Specifically, for any $m \in \mathbb{N}$ and $\mathcal{U} = (z_1, \ldots, z_m) \in \mathcal{X}^m$, we suppose $\mathcal{Q}^*_{\mathcal{U}} \subseteq \mathcal{Q}^{**}_{\mathcal{U}}$ be the set of data-dependent queries q with the property that, for any function f, $\forall a \in q(f)$, $\exists \mathcal{Y}_1, \mathcal{Y}_2, \ldots, \mathcal{Y}_m \subseteq \{1, \ldots, k\}$ such that $a = \bigcap_{i=1}^m \{g | g(x_i) \in \mathcal{Y}_i\}$. Queries of this type actually return constraints on the labels of particular examples: so answers such as "$f(x_1) \neq 2$" are valid, but answers such as "$f(x_1) \neq f(x_2)$" are not. In our setting, the learning algorithm is only permitted to make queries from $\mathcal{Q}_{\mathcal{U}}$ for (finite) sets $\mathcal{U} \subseteq \{x_1, x_2, \ldots\}$.

For the remainder of this section, for every $m \in \mathbb{N}$ and $\mathcal{U} \in \mathcal{X}^m$, fix some arbitrary set of valid queries $\mathcal{Q}_{\mathcal{U}} \subseteq \mathcal{Q}^*_{\mathcal{U}}$, and let $\mathbb{Q} = \{\mathcal{Q}_{\mathcal{U}} : \mathcal{U} \in \bigcup_m \mathcal{X}^m\}$. In this setting, we define the *query* *complexity*, analogous to the above, as a minimal quantity $\mathrm{QC}_{\mathbb{Q}}(\epsilon, \delta, \mathbb{C}, \mathbb{D})$ such that there exists an algorithm $\mathcal{A}$ which, for any target distribution $\mathcal{D}_{XY} \in \mathbb{D}$, with probability at least $1-\delta$, makes at most $\mathrm{QC}_{\mathbb{Q}}(\epsilon, \delta, \mathbb{C}, \mathbb{D})$ queries from $\bigcup_{\text{finite } \mathcal{U} \subset \{x_1, x_2, \ldots\}} \mathcal{Q}_{\mathcal{U}}$ and then returns a classifier $\hat{h}$ with $\mathrm{err}(\hat{h}) \leq \eta + \epsilon$. Also denote by $V[\mathcal{U}]$ a subset of V such that $\forall h \in V$, $|\{g \in V[\mathcal{U}] : g(\mathcal{U}) = h(\mathcal{U})\}| = 1$.

Following analogous to Balcázar et al. (2002); Hanneke (2009, 2007b), define the *abstract identification dimension* of a function f with respect to $V \subseteq \mathbb{C}$ and $\mathcal{U} \in \mathcal{X}^m$ (for any $m \in \mathbb{N}$) as $\mathrm{AIdim}(f, V, \mathcal{U}) = \inf\{n | \forall T \in \mathcal{T}^f, \exists Q \subseteq \mathcal{Q}_{\mathcal{U}} \text{ s.t. } |Q| \leq n \text{ and } |V[\mathcal{U}] \cap T(Q)| \leq 1\}$. Then define $\mathrm{AIdim}(f, V, m, \delta) = \inf\{n : P_{\mathcal{U} \sim \mathcal{D}^m}(\mathrm{AIdim}(f, V, \mathcal{U}) \geq n) \leq \delta\}$, and finally $\mathrm{AIdim}(V, m, \delta) = \sup_f \mathrm{AIdim}(f, V, m, \delta)$, where f ranges over all classifiers. This notion of complexity is inspired by analogous notions (of the same name) defined for the Exact Learning model by Balcázar et al. (2002), where it tightly characterizes the query complexity. The extension of this complexity measure to this sample-based setting runs analogous to the extension of the extended teaching dimension by Hanneke (2007b) from the original notion of Hegedüs (1995), to study the query complexity of active learning with label requests; indeed, in the special case that the sets $\mathcal{Q}_{\mathcal{U}}$ correspond to label request queries, the above $\mathrm{AIdim}(f, V, \mathcal{U})$ quantity is equal to the generalization of the extended teaching dimension explored by Hanneke (2007b). In the case of class-conditional queries, we always have $\mathrm{AIdim}(V, m, \delta) \leq k$, while for sample-based equivalence queries (requesting a mistake for a given proposed labeling of the sample $S \subseteq \mathcal{U}$), $\mathrm{AIdim}(V, m, \delta) = 1$.

For our present purposes, rather than AIdim, we define a related quantity, which we call the IAdim, which reverses certain quantifiers. Specifically, let $\mathrm{IAdim}(f, V, \mathcal{U}) = \inf\{n | \exists Q \subseteq \mathcal{Q}_{\mathcal{U}} \text{ s.t. } |Q| \leq n \text{ and } \forall T \in \mathcal{T}^f, |V[\mathcal{U}] \cap T(Q)| \leq 1\}$. Then define $\mathrm{IAdim}(f, V, m, \delta) = \inf\{n : P_{\mathcal{U} \sim \mathcal{D}^m}(\mathrm{IAdim}(f, V, \mathcal{U}) \geq n) \leq \delta\}$, and finally $\mathrm{IAdim}(V, m, \delta) = \sup_f \mathrm{IAdim}(f, V, m, \delta)$.

In words, $\mathrm{IAdim}(f, V, \mathcal{U})$ is the smallest number of queries such that any valid answers consistent with f will leave at most one equivalence class in $V[\mathcal{U}]$ consistent with the answers: that is, there will be at most one labeling of $\mathcal{U}$ consistent with a classifier in V that is itself consistent with the answers to the queries. This contrasts with $\mathrm{AIdim}(f, V, \mathcal{U})$, in which we allow the choice of queries to adapt based on the oracle's choice of answers.

Examples In the special case where $\mathcal{Q}$ corresponds to *label requests*, we have $\mathrm{IAdim}(f, V, \mathcal{U}) = \mathrm{AIdim}(f, V, \mathcal{U})$, and both are equal to the *extended teaching dimension* quantity from Hanneke (2007b). For instance, when V is a set of threshold classifiers, we have $\mathrm{IAdim}(f, V, \mathcal{U}) = 2$, simply taking any two adjacent examples in $\mathcal{U}$ for which f has opposite labels. In fact, for several families of queries mentioned in various contexts above (class conditional queries, mistake queries, label requests, close examples labeled differently), the notions of AIdim and IAdim are actually identical. Indeed, one can show they will be equal in binary classification whenever the queries $\mathcal{Q}_{\mathcal{U}}$ have a certain *projective* property (where any query whose answer only constrains the labels of $\mathcal{U}' \subseteq \mathcal{U}$ has a query in $\mathcal{Q}_{\mathcal{U}'}$ that allows this same answer).

Focusing queries Formally, when $k = 2$, we say the family $\mathcal{Q}$ of queries is *focusing* if, for any finite set $\mathcal{U} \subseteq \mathcal{X}$, any query $q \in \mathcal{Q}_{\mathcal{U}}$, any classifier f, and any $a \in q(f)$, letting $\mathcal{U}' = \{x \in \mathcal{U} : \{h(x) : h \in a\} \neq \{1, 2\}\}$, there exists $q' \in \mathcal{Q}_{\mathcal{U}}$ such that $a \in q'(f) \subseteq \{a' \in q(f) : \forall x \in \mathcal{U} \setminus \mathcal{U}', \{h(x) : h \in a'\} = \{1, 2\}\}$. That is, by restricting the unlabeled sample to just those where the answer is informative, there is a query for that subsample such that the answer is still valid, and furthermore there are no answers for the query on the subset that were not valid for the original set. For instance, if q is a label request query for the label of a point $x \in \mathcal{U}$, then $q \in \mathcal{Q}_{\mathcal{U}}$, but also

$q \in \mathcal{Q}_{\{x\}}$, so label requests are a focusing query (where $q' = q$ in this case). As another example, if q requests a mistake for some classifier g from the sample $\mathcal{U}$, then for any point $x \in \mathcal{U}$ that the query could possibly indicate as a mistake, this remains a valid response to a mistake query q' for g from the subset $\{x\}$; thus, mistake queries are also focusing. We can show that, when $k = 2$ and $\mathcal{Q}$ is focusing, $\mathrm{AIdim}(f, V, \mathcal{U}) = \mathrm{IAdim}(f, V, \mathcal{U})$ for all f, V, and $\mathcal{U}$. Specifically, let $T \in \mathcal{T}^f$ be a maximizer of $\min\{|Q| : Q \subseteq \mathcal{Q}_{\mathcal{U}}$ s.t. $|V[\mathcal{U}] \cap T(Q)| \leq 1\}$, and without loss, we can suppose that for each $q \in \mathcal{Q}_{\mathcal{U}}$, there is no $T' \in \mathcal{T}^f$ with $T(q) \subset T'(q)$ (since changing $T(q)$ to $T'(q)$ would still result in a maximizer). Let $Q \subseteq \mathcal{Q}_{\mathcal{U}}$ be of minimal $|Q|$ such that $|V[\mathcal{U}] \cap T(Q)| \leq 1$. Then let $Q' \subseteq \mathcal{Q}_{\mathcal{U}}$ denote the set of queries q' guaranteed to exist by the definition of focusing, corresponding to the queries in Q. Now, for the purpose of obtaining a contradiction, suppose there exists $T' \in \mathcal{T}^f$ such that $|V[\mathcal{U}] \cap T'(Q')| > 1$. Then by the definition of focusing, there exists $T'' \in \mathcal{T}^f$ such that $T''(q) = T'(q')$ for each pair of corresponding $q \in Q$ and $q' \in Q'$, and for each such $q \in Q$, $\{x \in \mathcal{U} : \{h(x) : h \in T''(q)\} = \{1,2\}\} \supseteq \{x \in \mathcal{U} : \{h(x) : h \in T(q)\} = \{1,2\}\}$, with the inclusion being *strict* for at least one $q \in Q$, so that $T''(q) \supset T(q)$; but this contradicts the assumption that no such T'' exists. Thus, we must have $|V[\mathcal{U}] \cap T'(Q')| \leq 1$ for *every* $T' \in \mathcal{T}^f$, so that Q' satisfies the criterion in the definition of $\mathrm{IAdim}(f, V, \mathcal{U})$, with $|Q| = \mathrm{AIdim}(f, V, \mathcal{U})$, so that $\mathrm{IAdim}(f, V, \mathcal{U}) \leq \mathrm{AIdim}(f, V, \mathcal{U})$; the reverse inequality is obvious from the definitions, so that $\mathrm{IAdim}(f, V, \mathcal{U}) = \mathrm{AIdim}(f, V, \mathcal{U})$.

E.2. A bound on query complexity based on IAdim.

We present here a result a bound on query complexity in terms of IAdim. Specifically, using a technique essentially analogous to those used for class-conditional queries above, except with some additional work required in Phase 2 (analogous to the method of Hanneke (2007b)), we are able to prove the following results.

Theorem 19 *In the case of $k = 2$, for any $\mathbb{C}$ of VC dimension d, for $\eta \leq 1/64$, there are values $\mathrm{s} = \Theta\left(\frac{1}{\eta+\epsilon}\right)$ and $q = O\left(\left(\frac{\eta^2}{\epsilon^2} + 1\right)\left(d \log \frac{1}{\epsilon} + \log \frac{1}{\delta}\right)\left(\log \frac{d}{\epsilon\delta}\right)\right)$ such that, for $\mathrm{IA} = \mathrm{IAdim}\left(\mathbb{C}, \mathrm{s}, \delta/q\right)$,*

$$\mathrm{QC}_{\mathbb{Q}}(\epsilon, \delta, \mathbb{C}, \mathcal{A}\mathrm{gnostic}(\mathbb{C}, \eta)) \leq \mathrm{IA}q = O\left(\mathrm{IA}\left(\frac{\eta^2}{\epsilon^2} + 1\right)\left(d \log \frac{1}{\epsilon} + \log \frac{1}{\delta}\right)\left(\log \frac{d}{\epsilon\delta}\right)\right).$$

Moreover, this result also holds for $k > 2$ (with additional k-dependent constant factors) if $\epsilon \geq ck\eta$, for an appropriate constant $c > 0$.

A similar result should also hold for the bounded noise case, analogous to Theorems 8 and 14. We conjecture that these results remain valid in general when we replace IAdim by AIdim, even for non-focusing types of queries. However, a proof of such a result would require a somewhat new line of reasoning compared with that used here.

E.2.1. PROOF OF THEOREM 19

Intuition The primary tool used to obtain these results is a replacement for the Find-Mistake subroutine above, now based on queries from $\mathbb{Q}$. Our goal in constructing this new subroutine (which we call Simulated-Find-Mistake) is the following behavior: given a classifier h, a set of classifiers V, and a set of unlabeled examples S, *if* there exists a classifier $g \in V$ that correctly labels the points in S (in agreement with their true y_i labels), *then* the procedure either identifies a point in S where h makes a mistake, or if no such point exists it identifies the complete true labeling of S. Based on the definition of IAdim, for a given set S of unlabeled examples, a classifier h, and a set of classifiers V, we can find $\mathrm{IAdim}(h, V, S)$ queries $M \subseteq \mathcal{Q}_S$ such that, if h happens to be consistent with the oracle's responses to those queries (based on the true labels of points in S), then all of the classifiers in V that are consistent with the oracle's responses agree on the labels of all points in S: that is, there is at most one equivalence class in $V[S]$ consistent with the answers.[4] Note that this is not always the same as getting back the *true* labels of S when h is consistent with the answers, since some of the labels may be noisy (hence, it is possible that no $g \in V$ has $\mathrm{err}_S(g) = 0$). However, we *can* guarantee that *if* there is a classifier $g \in V$ that correctly labels all of the points in S, *and* h happens to be consistent with all of the answers given by the oracle (corresponding to the true labels) for the queries in M, *then* all of the classifiers in V consistent with the oracle's answers will correctly label all of the points in S.

For example, if $\mathbb{Q}$ corresponds to label request queries, and V is a set of *threshold* classifiers $x \mapsto 2I_{[t,\infty)}(x) - 1$ on $\mathbb{R}$, then for any S and any h, the queries could be for any two adjacent points in S that h labels differently (or the extremal points in S if h is homogeneous on S). If those two points happen to actually be labeled as in h, then there will be at most one labeling corresponding to a threshold classifier consistent with these labels. However, if one of these two labels corresponded to a noisy point, then h^* will not agree with this one consistent labeling.

Summarizing, for a given set S of random unlabeled samples, at a given time in the algorithm when the set of surviving classifiers so far is denoted V, there exist $\mathrm{IAdim}(\mathrm{plur}(V), V, S)$ queries such that, if any of the responses are not consistent with $\mathrm{plur}(V)$, we receive a label constraint contradicting at least a $1/k$ fraction of V, and if the responses are all consistent with $\mathrm{plur}(V)$, then there is at most one labeling of S by a classifier consistent with the answers.

Thus, in Phase 1, we can proceed as before, taking samples of size $\Theta(\frac{1}{\eta+\epsilon})$, so that most of them do not contain a point contradicting h^*, but for which $\mathrm{plur}(V)$ makes mistakes on a significantly larger fraction of them. By using the above tool to elicit responses that eitehr contradict $\mathrm{plur}(V)$ or contradict the vast majority of V, we can proceed as in Phase 1 by keeping a tally of how many contradicting answers each classifier in V suffers, and removing it if that tally exceeds the number of such contradictions we expect for h^*.

As before, Phase 1 will only work up to a certain point, at which the error rate of $\mathrm{plur}(V)$ is within a constant factor of the error rate of h^*. At this point, we would like something analogous to Phase 2 above. However, things are not quite as simple as they were for class-conditional queries, since our tool for finding contradictions does not necessarily give us the *true* labels but rather the labels of some classifier in V consistent with the responses. However, as long as h^* does not make any mistakes on that sample, those will be precisely the h^* labels. Since the error rates of both h^* and $\mathrm{plur}(V)$ are $\propto \eta + \epsilon$, taking a large enough number of random subsets of size $\Theta(\frac{1}{\eta+\epsilon})$ should guarantee that for most of those sets (a constant fraction greater than $1/2$), h^* and $\mathrm{plur}(V)$ are both correct (with respect to the true labels), so that the answers to our queries will be consistent with both $\mathrm{plur}(V)$ and h^*, and thus we can reliably *infer* the labels of the points in such sets. However, some fraction of such sets *will* have points inconsistent with $\mathrm{plur}(V)$ or h^*, and we may have no way to tell which ones. To resolve this, we make use of a trick from Hanneke (2007b): namely, we sample the sets of size $\Theta(\frac{1}{\eta+\epsilon})$ from a fixed pool $\mathcal{U}$ *with replacement*, and take enough

[4] Recall that, in this context, the classifiers consistent with each answer might have disagreements on the labels of points in S (possibly even all of the labels). But when combining all the answers, only (at most) one equivalence class will be consistent with *all* of the answers to these $\mathrm{IAdim}(h, V, S)$ queries.

of these sets so that, for each $x \in \mathcal{U}$, x appears in a large enough number of these small subsamples that we are guaranteed with high probability that most of them do not have any (other) points for which h^* or $\mathrm{plur}(V)$ make mistakes. Thus, assuming the answers to the queries do not reveal any information about the label of x itself, the answers to the queries will be consistent with $\mathrm{plur}(V)$, and the h^* labeling will be the one consistent with the answers, so that we get an accurate inference of $h^*(x)$ for the *majority* of the sets containing x: that is, the majority vote over the inferred labels for x will be $h^*(x)$ with high probability. On the other hand, if the answers to the queries directly reveal information about the label of x, then we can simply use that revealed label itself, rather than inferring the h^* label. Thus, in the end, we produce a label for each $x \in \mathcal{U}$, some the actual y labels, the others the $h^*(x)$ labels. All that remains is to show that a labeled data set of this type, in the contexts the labeled sample was used above for class-conditional queries, will serve the same (or better) purpose, so that the required guarantees remain valid.

Formal Description The formal details are analogous to Hanneke (2007b), and are specified as follows. Define the following methods, intended to replace their respective counterparts in the General Agnostic Interactive Algorithm above.

```
Subroutine 2 Simulated-Find-Mistake
Input: The sequence $S = (x_1, x_2, \ldots, x_m)$; classifier $h$; set of classifiers $V$
  1. Let $Q$ be the minimal set of queries from the definition of $\mathrm{IAdim}(h, V, S)$
  2. Make the queries in $Q$, and let $T(Q)$ denote the oracle's answers
Output: $T(Q)$
```

```
Algorithm 5 General Queries Agnostic Interactive Algorithm
Input: The sequence $(x_1, x_2, ..., )$; values $u$, $\mathrm{s}_1$, $\mathrm{s}_2$, $\delta$;
  1. Let $V$ be a (minimal) $\epsilon$-cover of the space of classifiers $\mathbb{C}$ with respect to $\mathcal{D}_X$. Let $U$ be $\{x_1, ..., x_u\}$.
  2. Run the General Queries Halving Algorithm (Subroutine 3) with input $U$; $V$, $\mathrm{s}_1$, $c \ln \frac{4 \log_2 |V|}{\delta}$, and get $h$.
  3. Run the General Queries Refining Algorithm (Subroutine 4) with input $U$, $V$, $h$, $\mathrm{s}_2$, $\left\lceil c \frac{u}{\mathrm{s}_2} \ln \frac{u}{\delta} \right\rceil$, and get labeled sample $L$ returned.
  4. Find an hypothesis $h' \in V$ of minimum $\mathrm{err}_L(h')$.
Output Hypothesis $h'$ (and $L$).
```

The only major changes compared to Algorithm 1 are in Find-Mistake and the Refining Algorithm. We have the following lemmas.

Lemma 20 *Suppose that some $\hat{h} \in V$ has $\mathrm{err}_U(\hat{h}) \leq \beta$, for $\beta \in [0, 1/(32k)]$. With probability $\geq 1 - \delta/4$, running Subroutine 3 with U, V, and values $s = \left\lfloor \frac{1}{16k\beta} \right\rfloor$ and $N = c \ln \frac{4 \log_2 |V|}{\delta}$ (for an appropriate constant $c \in (0, \infty)$), we have that for every round of the loop of Step 2, the following hold.*

- *There are at most $N/(9k)$ samples S_i containing a point x_j for which $\hat{h}(x_j) \neq y_j$; in particular, $\hat{h} \notin T_i$ for at most $N/(9k)$ of the returned T_i values.*

Subroutine 3 General Queries Halving Algorithm

```
Input: The sequence $U = (x_1, x_2, ..., x_{\mathrm{ps}})$; set of classifiers $V$; values s, $N$
1. Set $b = \mathrm{true}$, $t = 0$.
2. while $b$
    (a) Draw $S_1$, $S_2$, ..., $S_N$ of size s uniformly without replacement from $U$.
    (b) For each $i$, call Simulated-Find-Mistake with arguments $S_i$, $\mathrm{plur}(V)$, and $V$. Let $T_i$ be the return value.
    (c) If more than $N/(3k)$ of the sets have $\mathrm{plur}(V) \notin T_i$, remove from $V$ every $h \in V$ with $|\{i : h \notin T_i\}| > N/(9k)$
    (d) Else $b \leftarrow 0$
Output Hypothesis $\mathrm{plur}(V)$.
```

Subroutine 4 General Queries Refining Algorithm

```
Input: The sequence $U = (x_1, x_2, ..., x_{\mathrm{ps}})$; set of classifiers $V$; classifier $h$; values s, $M$;
2. Draw $S_1, S_2, \ldots, S_M$ for size s uniformly without replacement from $U$
3. For each $i$, call Simulated-Find-Mistake with arguments $S_i$, $h$, and $V$, and let $T_i$ denote the returned value
4. For each $j \leq \mathrm{ps}$, let $\hat{I}_j = \{i : x_j \in S_i, h \in T_i,$ and $T_i \cap V \neq \emptyset\}$
5. For each $i \in \bigcup_j \hat{I}_j$, let $h_i \in T_i \cap V$
6. For each $j \leq \mathrm{ps}$, let $\hat{y}_j$ be the plurality value of $\{h_i(x_j) : i \in \hat{I}_j\}$
7. Let $L = \{(x_1, \hat{y}_1), \ldots, (x_{\mathrm{ps}}, \hat{y}_{\mathrm{ps}})\}$
Output Labeled sample $L$
```

- *If* $\mathrm{err}_U(\mathrm{plur}(V)) \geq 11k\beta$*, then* $\mathrm{plur}(V) \notin T_i$ *for* $> (2/3 - 1/(9k))N$ *of the returned values.*
- *If* $\mathrm{plur}(V) \notin T_i$ *for* $> (2/3 - 1/(9k))N$ *of the returned values, then the number of* h *in* V *with* $h \notin T_i$ *for* $> N/(9k)$ *of the returned values* T_i *in Step* $3(c)$ *is at least* $\frac{(1-1/k)(1-1/(6k))}{(1-1/(3k))}|V| < |V|$.

Proof As before, a Chernoff bound implies the first claim holds with probability at least $1 - \delta/(c' \log_2 |V|)$. Similarly for the second claim, as before, a Chernoff bound implies that with probability at least $1 - \delta/(c' \log_2 |V|)$, at least $(2/3)N$ of the sets S_i contain a point x_j such that $\mathrm{plur}(V)(x_j) \neq y_j$. In particular, any such set S_i for which the labels are consistent with $\hat{h}$ necessarily has $|V \cap T_i| \geq |V|/k$. This happens for at least $(2/3 - 1/(9k))N$ of the sets. Following the combinatorial argument as before, now consider a bipartite graph where the left side has all the classifiers in V, while the right side has the returned T_i sets for those i with $\mathrm{plur}(V) \notin T_i$, and an edge connects a left vertex to a right vertex if the associated hypothesis is not in the associated T_i set. Let M be the number of right vertices. The total number of edges is at least $M|V|/k$. Let $\alpha|V|$ be the number of classifiers in V missing from at most $N/(9k)$ of the T_i sets. The total number of edges is then upper bounded by $\alpha|V|N/(9k) + (1 - \alpha)|V|M$. Therefore,

$$M|V|/k \leq \alpha|V|N/(9k) + (1 - \alpha)|V|M,$$

which implies

$$|V|M(\alpha - 1 + 1/k) \leq \alpha|V|N/(9k).$$

Applying the lower bound $M \geq (2/3 - 1/(9k))N$, we get

$$(2/3 - 1/(9k))(\alpha - 1 + 1/k) \leq \alpha/(9k),$$

so that $\alpha \leq \frac{(2/3-1/(9k))(1-1/k)}{(2/3-2/(9k))} = \frac{(1-1/(6k))(1-1/k)}{(1-1/(3k))}$. This establishes the third claim. Note that $\alpha < 1$, since $(1 - 1/k) < (1 - 1/(3k))$.

The full result then follows by a union bound, as before, where now the constant c' will depend on k due to a change in the base of the logarithm to be $\frac{(1-1/(3k))}{(1-1/k)(1-1/(6k))}$. ■

Lemma 21 *For this result, we suppose $k = 2$. Suppose some $\hat{h} \in V$ has $\mathrm{err}_U(\hat{h}) \leq \beta$, for $\beta \in [0, 1/64]$, and that h has $\mathrm{err}_U(h) \leq 22\beta$. Consider running Subroutine 4 with U, V, h, and values $\mathrm{s} = \left\lfloor \frac{1}{64\beta} \right\rfloor$ and $M = \left\lceil c\frac{u}{\mathrm{s}} \ln \frac{u}{\delta} \right\rceil$ (for an appropriate constant $c > 1$), where $u = |U|$, and let L be the returned sample. Then $|L| = |U|$, and for every j with $x_j \in U$, there is exactly one $y \in \mathcal{Y}$ with $(x_j, y) \in L$; also, with probability at least $1 - \delta/4$, every $(x_j, y) \in L$ has either $y = y_j$ or $y = \hat{h}(x_j)$.*

Proof This argument runs similar to that of Lemma 2 in Hanneke (2007b). First note that, for any $x_j \in U$ with $y_j \neq \hat{h}(x_j)$, the $(x_j, y) \in L$ trivially satisfies the requirement, regardless of which value y takes.

Let $A = \{i : \hat{h} \notin T_i\}$ and $B = \{i : h \notin T_i\}$. A (respectively B) represent the indices of subsamples S_i for which $\hat{h}$ (respectively, h) is contradicted by the answers. Since $A \subseteq \{i : \mathrm{err}_{S_i}(\hat{h}) > 0\}$ and $B \subseteq \{i : \mathrm{err}_{S_i}(h) > 0\}$, we have $\mathbb{E}[|A| + |B|] \leq \frac{23}{64}M$. By a Chernoff bound, $P\left(|A \cup B| > \frac{3}{8}M\right) < e^{-c'M}$, for an appropriate constant $c' \in (0, 1)$.

For each $x_j \in U$ with $y_j = \hat{h}(x_j)$, let $I_{x_j} = \{i : x_j \in S_i\}$. Note that if $|I_{x_j} \cap (A \cup B)^c| > \frac{1}{2}|I_{x_j}|$, then $\hat{y}_j = \hat{h}(x_j)$. The remainder of the proof bounds the probability this fails to happen. Toward this end, we note (by a union bound)

$$\begin{aligned}
&P\left(|I_{x_j} \cap (A \cup B)| \geq \frac{1}{2}|I_{x_j}|\right) \\
&\leq P\left(|I_{x_j}| < \frac{\mathrm{s}M}{2u}\right) + P\left(|A \cup B| > \frac{3}{8}M\right) \\
&+ P\left(|I_{x_j} \cap (A \cup B)| \geq \frac{1}{2}|I_{x_j}| \wedge |I_{x_j}| \geq \frac{\mathrm{s}M}{2u} \wedge |A \cup B| \leq \frac{3}{8}M\right).
\end{aligned}$$

As shown above, the second term is at most $e^{-c'M}$. By a Chernoff bound, the first term is at most $e^{-\frac{\mathrm{s}M}{8u}}$. Finally, by a Chernoff bound, the last term is at most $e^{-\frac{\mathrm{s}M}{144u}}$. By setting the constant c in M appropriately, we have $e^{-c'M} + e^{-\frac{\mathrm{s}M}{8u}} + e^{-\frac{\mathrm{s}M}{144u}} \leq \delta/(4u)$. A union bound over $x_j \in U$ with $y_j = \hat{h}(x_j)$ then implies this holds for all such x_j, with probability at least $1 - \delta/4$. ■

The difficulty in extending this to $k > 2$ is that, for the noisy points, every set they appear in will contain a noisy point (trivially). But that means there might not be a classifier in V that correctly labels that set, so that we do not predictably infer a correct label for that point. In fact, the behavior in these cases might be somewhat unpredictable, so that we may even infer a label that is neither the true y_j nor the $\hat{h}(x_j)$ label. But then there could potentially be a classifier $g \in V$ with $\mathrm{err}_U(g)$ slightly smaller than 2β such that, for the L output by this proceedure, $\mathrm{err}_L(g) < \beta$ and in particular $\mathrm{err}_L(g) < \mathrm{err}_L(\hat{h})$, where $\hat{h} = \mathrm{argmin}_{h' \in V} \mathrm{err}_U(h')$.

Note that this issue is not present if we are only interested in identifying a classifier h of $\mathrm{err}(h) = O(\eta)$, since then it suffices to use Subroutine 3, so that we can achieve this result even for $k > 2$.

Proof [Proof of Theorem 19 (Sketch)] Theorem 19 now follows from the above two lemmas, in the same way that Theorem 5 followed from Lemmas 3 and 4. The only two twists are that now some of the labels in the set labeled set L are *denoised*, in the sense that $(x_j, y) \in L$ has $y_j \neq y = h'(x_j)$, which does not change the fact that h' is still the minimizer of $\mathrm{err}_L(h)$ over $h \in V$; so the above two lemmas, combined with the reasoning from the proof of Theorem 5 regarding the sufficiency of taking $u = O\left(\left(\frac{\eta+\epsilon}{\epsilon^2}\right)\left(d \log \frac{1}{\epsilon} + \log \frac{1}{\delta}\right)\right)$ random unlabeled examples U to guarantee $\mathrm{err}_U(h^*) \leq \eta + \epsilon$ and that the empirical risk minimizer h' has $\mathrm{err}(h') \leq \eta + \epsilon$, with probability at least $1 - \delta/4$, the above two lemmas (with $\beta = \eta + \epsilon$ in each) imply that Algorithm 5 (with u as above, $\mathrm{s}_1 = \lfloor 1/(32\beta) \rfloor$, and $\mathrm{s}_2 = \lfloor 1/(64\beta) \rfloor$) returns a classifier with $\mathrm{err}(h') \leq \eta + \epsilon$ with probability at least $1 - 3\delta/4$.

Additionally, the total number of calls to Simulated-Find-Mistake is $c \ln \frac{4 \log_2 |V|}{\delta} + \left\lceil c \frac{u}{\mathrm{s}_2} \ln \frac{u}{\delta} \right\rceil = O\left(\left(\frac{\eta^2}{\epsilon^2} + 1\right)\left(d \log \frac{1}{\epsilon} + \log \frac{1}{\delta}\right)\left(\log \frac{d}{\epsilon\delta}\right)\right)$; in the theorem, suppose n is 4 times this value. Since each call to Simulated-Find-Mistake uses at most $\mathrm{IA}(\mathbb{C}, \mathrm{s}, \delta/n)$ queries with probability at least $1 - \delta/n$ (where s is either $\lfloor 1/(32\beta) \rfloor$ or $\lfloor 1/(64\beta) \rfloor$, which ever gives the larger IA), a union bound implies that every call to Simulated-Find-Mistake will use at most IA queries, with probability at least $1 - \delta/4$. Composing this with the results from above via a union bound gives the result. ■