Statistical Active Learning Algorithms for Noise Tolerance and Differential Privacy

Maria Florina Balcan
Carnegie Mellon University
ninamf@cs.cmu.edu

Vitaly Feldman
IBM Research - Almaden
vitaly@post.harvard.edu

Abstract

We describe a framework for designing efficient active learning algorithms that are tolerant to random classification noise and are differentially-private. The framework is based on active learning algorithms that are *statistical* in the sense that they rely on estimates of expectations of functions of filtered random examples. It builds on the powerful statistical query framework of Kearns [Kea98].

We show that any efficient active statistical learning algorithm can be automatically converted to an efficient active learning algorithm which is tolerant to random classification noise as well as other forms of "uncorrelated" noise. The complexity of the resulting algorithms has information-theoretically optimal quadratic dependence on $1/(1-2\eta)$, where η is the noise rate.

We show that commonly studied concept classes including thresholds, rectangles, and linear separators can be efficiently actively learned in our framework. These results combined with our generic conversion lead to the first computationally-efficient algorithms for actively learning some of these concept classes in the presence of random classification noise that provide exponential improvement in the dependence on the error ϵ over their passive counterparts. In addition, we show that our algorithms can be automatically converted to efficient active differentially-private algorithms. This leads to the first differentially-private active learning algorithms with exponential label savings over the passive case.

1 Introduction

Most classic machine learning methods depend on the assumption that humans can annotate all the data available for training. However, many modern machine learning applications have massive amounts of unannotated or unlabeled data. As a consequence, there has been tremendous interest both in machine learning and its application areas in designing algorithms that most efficiently utilize the available data, while minimizing the need for human intervention. An extensively used and studied technique is active learning, where the algorithm is presented with a large pool of unlabeled examples and can interactively ask for the labels of examples of its own choosing from the pool, with the goal to drastically reduce labeling effort. This has been a major area of machine learning research in the past decade [Das11, Han], with several exciting developments on understanding its underlying statistical principles [FSST97, Das05, BBL06, BBZ07, Han07, DHM07, CN07, BHW08, Kol10, BHLZ10, Wan11, RR11, BH12]. In particular, several general characterizations have been developed for describing when active learning can in principle have an advantage over the classic passive supervised learning paradigm, and by how much. While the label complexity aspect of active learning has been intensively studied and is currently well understood, the question of providing computationally efficient noise tolerant active learning algorithms has remained largely open. In particular, prior to this work, there were no known efficient active algorithms for concept classes of super-constant VC-dimension that are provably robust to random and independent noise while giving improvements over the passive case.

1.1 Our Results

We propose a framework for designing efficient (polynomial time) active learning algorithms which is based on restricting the way in which examples (both labeled and unlabeled) are accessed by the algorithm. These restricted algorithms can be easily simulated using active sampling and, in addition, possess a number of other useful properties. The main property we will consider is tolerance to random classification noise of rate η (each label is flipped randomly and independently with probability η [AL88]). Further, as we will show, the algorithms are tolerant to other forms of noise and can be simulated in a differentially-private way.

In our restriction, instead of access to random examples from some distribution P over $X \times Y$ the learning algorithm only gets "active" estimates of the statistical properties of P in the following sense. The algorithm can choose any *filter* function $\chi(x) : X \to [0,1]$ and a query function $\phi : X \times Y \to [-1,1]$ for any χ and ϕ. For simplicity we can think of χ as an indicator function of some set $\chi_S \subseteq X$ of "informative" points and of ϕ as some useful property of the target function. For this pair of functions the learning algorithm can get an estimate of $\mathbf{E}_{(x,y)\sim P}[\phi(x,y) \mid x \in \chi_S]$. For τ and τ_0 chosen by the algorithm the estimate is provided to within *tolerance* τ as long as $\mathbf{E}_{(x,y)\sim P}[x \in \chi_S] \geq \tau_0$ (nothing is guaranteed otherwise). The key point it that when we simulate this query from random examples, the inverse of τ corresponds to the label complexity of the algorithm and the inverse of τ_0 corresponds to its unlabeled sample complexity. Such a query is referred to as *active statistical query (SQ)* and algorithms using active SQs are referred to as *active statistical algorithms*.

Our framework builds on the classic statistical query (SQ) learning framework of Kearns [Kea98] defined in the context of PAC learning model [Val84]. The SQ model is based on estimates of expectations of functions of examples (but without the additional filter function) and was defined in order to design efficient noise tolerant algorithms in the PAC model. Despite the restrictive form, most of the learning algorithms in the PAC model and other standard techniques in machine learning and statistics used for problems over distributions have SQ analogues [Kea98, BFKV97, BDMN05, CKL$^+$06, FGR$^+$13][1]. Further, statistical algorithms enjoy additional properties: they can be simulated in a differentially-private way [BDMN05], automatically parallelized on multi-core architectures [CKL$^+$06] and have known information-theoretic characterizations of query complexity [BFJ$^+$94, Fel12]. As we show, our framework inherits the strengths of the SQ model while, as we will argue, capturing the power of active learning.

At a first glance being active and statistical appear to be incompatible requirements on the algorithm. Active algorithms typically make label query decisions on the basis of examining individual samples (for example as in binary search for learning a threshold or the algorithms in [FSST97, DHM07, DKM09]). At the same time statistical algorithms can only examine properties of the underlying distribution. But there also exist a number of active learning algorithms that can be seen as applying passive learning techniques to batches of examples that are obtained from querying labels of samples that satisfy the same filter. These include the general A^2 algorithm

[1] The sample complexity of the SQ analogues might be polynomially larger though.

[BBL06] and, for example, algorithms in [BBZ07, DH08, BDL09, BL13]. As we show, we can build on these techniques to provide algorithms that fit our framework.

We start by presenting a general reduction showing that any efficient active statistical learning algorithm can be automatically converted to an efficient active learning algorithm which is tolerant to random classification noise as well as other forms of "uncorrelated" noise. The sample complexity of the resulting algorithms depends just quadratically on $1/(1-2\eta)$, where η is the noise rate.

We then demonstrate the generality of our framework by showing that the most commonly studied concept classes including thresholds, balanced rectangles, and homogenous linear separators can be efficiently actively learned via active statistical algorithms. For these concept classes, we design efficient active learning algorithms that are statistical and provide the same exponential improvements in the dependence on the error ϵ over passive learning as their non-statistical counterparts.

The primary problem we consider is active learning of homogeneous halfspaces a problem that has attracted a lot of interest in the theory of active learning [FSST97, Das05, BBZ07, BDL09, DKM09, CGZ10, DGS12, BL13, GSSS13]. We describe two algorithms for the problem. First, building on insights from margin based analysis of active learning [BBZ07, BL13], we give an active statistical learning algorithm for homogeneous halfspaces over all isotropic log-concave distributions, a wide class of distributions that includes many well-studied density functions and has played an important role in several areas including sampling, optimization, integration, and learning [LV07]. Our algorithm for this setting proceeds in rounds; in round t we build a better approximation w_t to the target function by using a passive SQ learning algorithm (e.g., the one of [DV04]) over a distribution D_t that is a mixture of distributions in which each component is the original distribution conditioned on being within a certain distance from the hyperplane defined by previous approximations w_i. To perform passive statistical queries relative to D_t we use active SQs with a corresponding real valued filter. This algorithm is computationally efficient and uses only $\mathrm{poly}(d, \log(1/\epsilon))$ active statistical queries of tolerance inverse-polynomial in the dimension d and $\log(1/\epsilon)$.

For the special case of the uniform distribution over the unit ball we give a new, simpler and substantially more efficient active statistical learning algorithm. Our algorithm is based on measuring the error of a halfspace conditioned on being within some margin of that halfspace. We show that such measurements performed on the perturbations of the current hypothesis along the d basis vectors can be combined to derive a better hypothesis. This approach differs substantially from the previous algorithms for this problem [BBZ07, DKM09]. The algorithm is computationally efficient and uses $d\log(1/\epsilon)$ active SQs with tolerance of $\Omega(1/\sqrt{d})$ and filter tolerance of $\Omega(\epsilon)$.

These results, combined with our generic simulation of active statistical algorithms in the presence of random classification noise (RCN) lead to the first known computationally efficient algorithms for actively learning halfspaces which are RCN tolerant and give provable label savings over the passive case. For the uniform distribution case this leads to an algorithm with sample complexity of $O((1-2\eta)^{-2} \cdot d^2 \log(1/\epsilon) \log(d \log(1/\epsilon)))$ and for the general isotropic log-concave case we get sample complexity of $\mathrm{poly}(d, \log(1/\epsilon), 1/(1-2\eta))$. This is worse than the sample complexity in the noiseless case which is just $O((d + \log\log(1/\epsilon)) \log(1/\epsilon))$ [BL13]. However, compared to passive learning in the presence of RCN, our algorithms have exponentially better dependence on ϵ and essentially the same dependence on d and $1/(1-2\eta)$. One issue with the generic simulation is that it requires knowledge of η (or an almost precise estimate). Standard approach to dealing with this issue does not always work in the active setting and for our log-concave and the uniform distribution algorithms we give a specialized argument that preserves the exponential improvement in the dependence on ϵ.

Differentially-private active learning: In many application of machine learning such as medical and financial record analysis, data is both sensitive and expensive to label. However, to the best of our knowledge, there are no formal results addressing both of these constraints. We address the problem by defining a natural model of differentially-private active learning. In our model we assume that a learner has full access to unlabeled portion of some database of n examples $S \subseteq X \times Y$ which correspond to records of individual participants in the database. In addition, for every element of the database S the learner can request the label of that element. As usual, the goal is to minimize the number of label requests (such setup is referred to as *pool-based* active learning [MN98]). In addition, we would like to preserve the *differential privacy* of the participants in the database, a now-standard notion of privacy introduced in [DMNS06]. Informally speaking, an algorithm is differentially private if adding any record to S (or removing a record from S) does not affect the probability that any specific hypothesis will be output by the algorithm significantly.

As first shown by Blum *et al.* [BDMN05], SQ algorithms can be automatically translated into differentially-private algorithms by using the so-called Laplace mechanism (see also [KLN$^+$11]). Using a similar approach, we show that active SQ learning algorithms can be automatically transformed into differentially-private active learning algorithms. As a consequence, for all the classes for which we provide statistical active learning algorithms that can be simulated by using only $\mathrm{poly}(d, \log(1/\epsilon))$ labeled examples (including thresholds and halfspaces), we can learn and preserve privacy with much fewer label requests than those required by even non-private classic passive learning algorithms, and can do so even when in our model the privacy parameter is very small. Note that while we focus on the number of label requests, the algorithms also preserve the differential privacy of the unlabeled points.

1.2 Additional Related Work

As we have mentioned, most prior theoretical work on active learning focuses on either sample complexity bounds (without regard for efficiency) or the noiseless case. For random classification noise in particular, [BH12] provides a sample complexity analysis based on the notion of splitting index that is optimal up to polylog factors and works for general concept classes and distributions, but it is not computationally efficient. In addition, several works give active learning algorithms with empirical evidence of robustness to certain types of noise [BDL09, GSSS13];

In [CGZ10, DGS12] online learning algorithms in the selective sampling framework are presented, where labels must be actively queried before they are revealed. Under the assumption that the label conditional distribution is a linear function determined by a fixed target vector, they provide bounds on the regret of the algorithm and on the number of labels it queries when faced with an adaptive adversarial strategy of generating the instances. As pointed out in [DGS12], these results can also be converted to a distributional PAC setting where instances x_t are drawn i.i.d. In this setting they obtain exponential improvement in label complexity over passive learning. These interesting results and techniques are not directly comparable to ours. Our framework is not restricted to halfspaces. Another important difference is that (as pointed out in [GSSS13]) the exponential improvement they give is not possible in the noiseless version of their setting. In other words, the addition of linear noise defined by the target makes the problem easier for active sampling. By contrast RCN can only make the classification task harder than in the realizable case.

Among the so called disagreement-based algorithms that provably work under very general noise models (adversarial label noise) and for general concept classes [BBL06, Kol10, DHM07, BHLZ10, Wan11, RR11, BH12, Han], those of Dasgupta, Hsu, and Monteleoni [DHM07] and Beygelzimer, Hsu, Langford, and Zhang [BHLZ10] are most amenable to implementation. While more amenable to implementation than other disagreement-based techniques, these algorithms assume the existence of a computationally efficient passive learning algorithm (for the concept class at hand) that can minimize the empirical error in the adversarial label noise — however, such algorithms are not known to exist for most concept classes, including linear separators.

Following the original publication of our work, Awasthi *et al.* [ABL14] give a polynomial-time active learning algorithm for learning linear separators in the presence of adversarial forms of noise. Their algorithm is the first one that can tolerate both adversarial label noise and malicious noise (where the adversary can corrupt both the instance part and the label part of the examples) as long as the rate of noise $\eta = O(\epsilon)$. We note that these results are not comparable to ours as we need the noise to be "uncorrelated" but can deal with noise of any rate (with complexity growing with $1/(1-2\eta)$).

Organization: Our model, its properties and several illustrative examples (including threshold functions and balanced rectangles) are given in Section 2. Our algorithm for learning homogeneous halfspaces over log-concave and uniform distributions are given in Section 3 and Section 4 respectively. The formal statement of differentially-private simulation is given in Section 5.

2 Active Statistical Algorithms

Let X be a domain and P be a distribution over labeled examples on X. We represent such a distribution by a pair (D, ψ) where D is the marginal distribution of P on X and $\psi : X \to [-1,1]$ is a function defined as $\psi(z) = \mathbf{E}_{(x,\ell)\sim P}[\ell \mid x = z]$. We will be considering learning in the PAC model (realizable case) where ψ is a boolean function, possibly corrupted by random noise.

When learning with respect to a distribution $P = (D, \psi)$, an active statistical learner has access to *active statistical queries*. A query of this type is a pair of functions (χ, ϕ), where $\chi : X \to [0,1]$ is the *filter* function which for a point x, specifies the probability with which the label of x should be queried. The function $\phi : X \times \{-1,1\} \to [-1,1]$ is the query function and depends on both point and the label. The filter function χ defines the distribution D conditioned on χ as follows: for each x the density function $D_{|\chi}(x)$ is defined as $D_{|\chi}(x) = D(x)\chi(x)/\mathbf{E}_D[\chi(x)]$. Note that if χ is an indicator function of some set S then $D_{|\chi}$ is exactly D conditioned on x being in S. Let $P_{|\chi}$ denote the conditioned distribution $(D_{|\chi}, \psi)$. In addition, a query has two tolerance parameters: filter tolerance τ_0 and query tolerance τ. In response to such a query the algorithm obtains a value μ such that if $\mathbf{E}_D[\chi(x)] \geq \tau_0$ then

$$\left|\mu - \mathop{\mathbf{E}}_{P_{|\chi}}[\phi(x,\ell)]\right| \leq \tau$$

(and nothing is guaranteed when $\mathbf{E}_D[\chi(x)] < \tau_0$).

An active statistical learning algorithm can also ask *target-independent* queries with tolerance τ which are just queries over unlabeled samples. That is for a query $\varphi : X \to [-1,1]$ the algorithm obtains a value μ, such that $|\mu - \mathbf{E}_D[\varphi(x)]| \leq \tau$. Such queries are not necessary when D is known to the learner.

For the purposes of obtaining noise tolerant algorithms one can relax the requirements of model and give the learning algorithm access to unlabelled samples. A similar variant of the model was considered in the context of SQ model [Kea98, BFKV97]. We refer to this variant as *label-statistical.* Label-statistical algorithms do not need access to target-independent queries access as they can simulate those using unlabelled samples.

Our definition generalizes the statistical query framework of Kearns [Kea98] which does not include filtering function, in other words a query is just a function $\phi : X \times \{-1,1\} \to [-1,1]$ and it has a single tolerance parameter τ. By definition, an active SQ (χ, ϕ) with tolerance τ relative to P is the same as a passive statistical query ϕ with tolerance τ relative to the distribution $P_{|\chi}$. In particular, a (passive) SQ is equivalent to an active SQ with filter $\chi \equiv 1$ and filter tolerance 1.

Finally we note that from the definition of active SQ we can see that

$$\mathop{\mathbf{E}}_{P_{|\chi}}[\phi(x,\ell)] = \mathop{\mathbf{E}}_{P}[\phi(x,\ell)\cdot\chi(x)]/\mathop{\mathbf{E}}_{P}[\chi(x)].$$

This implies that an active statistical query can be estimated using two passive statistical queries. However to estimate $\mathbf{E}_{P_{|\chi}}[\phi(x,\ell)]$ with tolerance τ one needs to estimate $\mathbf{E}_P[\phi(x,\ell)\cdot\chi(x)]$ with tolerance $\tau \cdot \mathbf{E}_P[\chi(x)]$ which can be much lower than τ. Tolerance of a SQ directly corresponds to the number of examples needed to evaluate it and therefore simulating active SQs passively might require many more examples.

2.1 Simulating Active Statistical Queries

In our model, the algorithm operates via statistical queries. In this section we describe how the answers to these queries can be simulated from random examples, which immediately implies that our algorithms can be transformed into active learning algorithms in the usual model [Das11].

We first note that a valid response to a target-independent query with tolerance τ can be obtained, with probability at least $1-\delta$, using $O(\tau^{-2}\log(1/\delta))$ unlabeled samples.

A natural way of simulating an active SQ is by filtering points drawn randomly from D: draw a random point x, let B be drawn from Bernoulli distribution with probability of 1 being $\chi(x)$; ask for the label of x when $B = 1$. The points for which we ask for a label are distributed according to $D_{|\chi}$. This implies that the empirical average of $\phi(x,\ell)$ on $O(\tau^{-2}\log(1/\delta))$ labeled examples will then give μ. Formally we get the following theorem.

Theorem 2.1. *Let $P = (D, \psi)$ be a distribution over $X \times \{-1,1\}$. There exists an active sampling algorithm that given functions $\chi : X \to [0,1]$, $\phi : X \times \{-1,1\} \to [-1,1]$, values $\tau_0 > 0$, $\tau > 0$, $\delta > 0$, and access to samples from P, with probability at least $1-\delta$, outputs a valid response to active statistical query (χ, ϕ) with tolerance parameters (τ_0, τ). The algorithm uses $O(\tau^{-2}\log(1/\delta))$ labeled examples from P and $O(\tau_0^{-1}\tau^{-2}\log(1/\delta))$ unlabeled samples from D.*

Proof. The Chernoff-Hoeffding bounds imply that for some $t = O(\tau^{-2}\log(1/\delta))$, the empirical mean of ϕ on t examples that are drawn randomly from $P_{|\chi}$ will, with probability at least $1-\delta/2$, be within τ of $\mathbf{E}_{P_{|\chi}}[\phi(x,\ell)]$. We can also assume that $\mathbf{E}_D[\chi(x)] \geq \tau_0$ since any value would be a valid response to the query when this assumption does not hold. By the standard multiplicative form of the Chernoff bound we also know that given $t_0 = O(\tau_0^{-1}t\cdot\log(1/\delta)) = O(\tau_0^{-1}\tau^{-2}\log(1/\delta)^2)$ random samples from D, with probability at least $1-\delta/2$, at least t of the samples will pass the filter χ. Therefore with, probability at least $1-\delta$, we will obtain at least t samples from D filtered using $\chi(x)$ and labeled examples on these points will give an estimate of $\mathbf{E}_{P_{|\chi}}[\phi(x,\ell)]$ with tolerance τ.

This procedure gives $\log(1/\delta)^2$ dependence on confidence (and not the claimed $\log(1/\delta)$). To get the claimed dependence we can use a standard confidence boosting technique. We run the above procedure with $\delta' = 1/3$, k times and let $\mu_1, \mu_2, \ldots, \mu_k$ denote the results. The simulation returns the median of μ_i's. The Chernoff bound implies that for $k = O(\log(1/\delta))$, with probability at least $1-\delta$, at least half of the μ_i's satisfy the condition $\left|\mu_i - \mathbf{E}_{P_{|\chi}}[\phi(x,\ell)]\right| \le \tau$. In particular, the median satisfies this condition. The dependence on δ of sample complexity is now as claimed. □

We remark that in some cases better sample complexity bounds can be obtained using multiplicative forms of the Chernoff-Hoeffding bounds (e.g. [AD98]).

A direct way to simulate all the queries of an active SQ algorithm is to estimate the response to each query using fresh samples and use the union bound to ensure that, with probability at least $1-\delta$, all queries are answered correctly. Such direct simulation of an algorithm that uses at most q queries can be done using $O(q\tau^{-2}\log(q/\delta))$ labeled examples and $O(q\tau_0^{-1}\tau^{-2}\log(q/\delta))$ unlabeled samples. However, in many cases a more careful analysis can be used to reduce the sample complexity of simulation. Labeled examples can be shared to simulate queries that use the same filter χ and do not depend on each other. This implies that the sample size sufficient for simulating q non-adaptive queries with the same filter scales logarithmically with q. More generally, given a set of q query functions (possibly chosen adaptively) which belong to some set Q of low complexity (such as VC dimension) one can reduce the sample complexity of estimating the answers to all q queries (with the same filter) by invoking the standard bounds based on uniform convergence (e.g. [BEHW89, Vap98]).

2.2 Noise tolerance

An important property of the simulation described in Theorem 2.1 is that it can be easily adapted to the case when the labels are corrupted by random classification noise [AL88]. For a distribution $P = (D, \psi)$ let P^η denote the distribution P with the label flipped with probability η randomly and independently of an example. It is easy to see that $P^\eta = (D, (1-2\eta)\psi)$. We now show that, as in the SQ model [Kea98], active statistical queries can be simulated given examples from P^η.

Theorem 2.2. *Let $P = (D, \psi)$ be a distribution over examples and let $\eta \in [0, 1/2)$ be a noise rate. There exists an active sampling algorithm that given functions $\chi : X \to [0,1]$, $\phi : X \times \{-1,1\} \to [-1,1]$, values η, $\tau_0 > 0$, $\tau > 0$, $\delta > 0$, and access to samples from P^η, with probability at least $1-\delta$, outputs a valid response to active statistical query (χ, ϕ) with tolerance parameters (τ_0, τ). The algorithm uses $O(\tau^{-2}(1-2\eta)^{-2}\log(1/\delta))$ labeled examples from P^η and $O(\tau_0^{-1}\tau^{-2}(1-2\eta)^{-2}\log(1/\delta))$ unlabeled samples from D.*

Proof. Using a simple observation from [BF02], we first decompose the statistical query ϕ into two parts: one that computes a correlation with the label and the other that does not depend on the label altogether. Namely,

$$\phi(x,\ell) = \phi(x,1)\frac{1+\ell}{2} + \phi(x,-1)\frac{1-\ell}{2} = \frac{\phi(x,1)-\phi(x,-1)}{2} \cdot \ell + \frac{\phi(x,1)+\phi(x,-1)}{2}\ . \quad (1)$$

Clearly, to estimate the value of $\mathbf{E}_{P_{|\mathcal{X}}}[\phi(x,\ell)]$ with tolerance τ it is sufficient to estimate the values of $\mathbf{E}_{P_{|\mathcal{X}}}[\frac{1}{2}(\phi(x,1)-\phi(x,-1))\cdot\ell]$ and $\mathbf{E}_{P_{|\mathcal{X}}}[\frac{1}{2}(\phi(x,1)+\phi(x,-1))]$ with tolerance $\tau/2$. The latter expression does not depend on the label and, in particular, is not affected by noise. Therefore it can be estimated as before using P^η in place of P. At the same time we can use the independence of noise to conclude[2],

$$\mathop{\mathbf{E}}_{P^{\eta}_{|\mathcal{X}}}\left[\frac{1}{2}(\phi(x,1)-\phi(x,-1))\cdot\ell\right]=(1-2\eta)\mathop{\mathbf{E}}_{P_{|\mathcal{X}}}\left[\frac{1}{2}(\phi(x,1)-\phi(x,-1))\cdot\ell\right].$$

This means that we can estimate $\mathbf{E}_{P_{|\mathcal{X}}}[\frac{1}{2}(\phi(x,1)-\phi(x,-1))\cdot\ell]$ with tolerance $\tau/2$ by estimating $\mathbf{E}_{P^{\eta}_{|\mathcal{X}}}[\frac{1}{2}(\phi(x,1)-\phi(x,-1))\cdot\ell]$ with tolerance $(1-2\eta)\tau/2$ and then multiplying the result by $1/(1-2\eta)$. The estimation of $\mathbf{E}_{P^{\eta}_{|\mathcal{X}}}[\frac{1}{2}(\phi(x,1)-\phi(x,-1))\cdot\ell]$ with tolerance $(1-2\eta)\tau/2$ can be done exactly as in Theorem 2.1. □

Note that the sample complexity of the resulting active sampling algorithm has information-theoretically optimal quadratic dependence on $1/(1-2\eta)$, where η is the noise rate. Note that RCN does not affect the unlabelled samples so algorithms which are only label-statistical algorithms can also be simulated in the presence of RCN.

Remark 2.3. *This simulation assumes that η is given to the algorithm exactly. It is easy to see from the proof, that any value η' such that $\frac{1-2\eta}{1-2\eta'}\in[1-\tau/4,1+\tau/4]$ can be used in place of η (with the tolerance of estimating $\mathbf{E}_{P^{\eta}_{|\mathcal{X}}}[\frac{1}{2}(\phi(x,1)-\phi(x,-1))\cdot\ell]$ set to $(1-2\eta)\tau/4$). In some learning scenarios even an approximate value of η is not known but it is known that $\eta\leq\eta_0<1/2$. To address this issue one can construct a sequence $\eta_1,\ldots,\eta_k$ of guesses of η, run the learning algorithm with each of those guesses in place of the true η and let $h_1,\ldots,h_k$ be the resulting hypotheses [Kea98]. One can then return the hypothesis h_i among those that has the best agreement with a suitably large sample. It is not hard to see that $k=O(\tau^{-1}\cdot\log(1/(1-2\eta_0)))$ guesses will suffice for this strategy to work [AD98].*

Passive hypothesis testing requires $\Omega(1/\epsilon)$ labeled examples and might be too expensive to be used with active learning algorithms. It is unclear if there exists a general approach for dealing with unknown η in the active learning setting that does not increase substantially the labeled example complexity. However, as we will demonstrate, in the context of specific active learning algorithms variants of this approach can be used to solve the problem.

We now show that more general types of noise can be tolerated as long as they are "uncorrelated" with the queries and the target function. Namely, we represent label noise using a function $\Lambda : X\to[0,1]$, where $\Lambda(x)$ gives the probability that the label of x is flipped. The rate of Λ when learning with respect to marginal distribution D over X is $\mathbf{E}_D[\Lambda(x)]$. For a distribution $P=(D,\psi)$ over examples, we denote by P^Λ the distribution P corrupted by label noise Λ. It is easy to see that $P^\Lambda=(D,\psi\cdot(1-2\Lambda))$. Intuitively, Λ is "uncorrelated" with a query if the way that Λ deviates from its rate is almost orthogonal to the query on the target distribution.

[2] For any function $f(x)$ that does not depend on the label, we have: $\mathbf{E}_{P^{\eta}_{|\mathcal{X}}}[f(x)\cdot\ell]=(1-\eta)\mathbf{E}_{P_{|\mathcal{X}}}[f(x)\cdot\ell]+\eta\cdot\mathbf{E}_{P_{|\mathcal{X}}}[f(x)\cdot(-\ell)]=(1-2\cdot\eta)\mathbf{E}_{P_{|\mathcal{X}}}[f(x)\cdot\ell]$. The first equality follows from the fact that under $P^{\eta}_{|\mathcal{X}}$, for any given x, there is a $(1-\eta)$ chance that the label is the same as under $P_{|\mathcal{X}}$, and an η chance that the label is the negation of the label obtained from $P_{|\mathcal{X}}$.

Definition 2.4. *Let $P = (D, \psi)$ be a distribution over examples and $\tau' > 0$. For functions $\chi : X \to [0,1]$, $\phi : X \times \{-1,1\} \to [-1,1]$, we say that a noise function $\Lambda : X \to [0,1]$ is (η, τ')-uncorrelated with ϕ and χ over P if,*

$$\left| \mathop{\mathbf{E}}_{D_{|\chi}} \left[\frac{\phi(x,1) - \phi(x,-1)}{2} \psi(x) \cdot (1 - 2(\Lambda(x) - \eta)) \right] \right| \leq \tau' \ .$$

In this definition $(1 - 2(\Lambda(x) - \eta))$ is the expectation of $\{-1,1\}$ coin that is flipped with probability $\Lambda(x) - \eta$, whereas $(\phi(x,1) - \phi(x,-1))\psi(x)$ is the part of the query which measures the correlation with the label. We now give an analogue of Theorem 2.2 for this more general setting.

Theorem 2.5. *Let $P = (D, \psi)$ be a distribution over examples, $\chi : X \to [0,1]$, $\phi : X \times \{-1,1\} \to [-1,1]$ be a query and a filter functions, $\eta \in [0, 1/2), \tau > 0$ and Λ be a noise function that is $(\eta, (1-2\eta)\tau/4)$-uncorrelated with ϕ and χ over P. There exists an active sampling algorithm that given functions χ and ϕ, values η, $\tau_0 > 0$, $\tau > 0$, $\delta > 0$, and access to samples from P^Λ, with probability at least $1 - \delta$, outputs a valid response to active statistical query (χ, ϕ) with tolerance parameters (τ_0, τ). The algorithm uses $O(\tau^{-2}(1-2\eta)^{-2}\log(1/\delta))$ labeled examples from P^Λ and $O(\tau_0^{-1}\tau^{-2}(1-2\eta)^{-2}\log(1/\delta))$ unlabeled samples from D.*

Proof. As in the proof of Theorem 2.2, we note that it is sufficient to estimate the value of

$$\lambda \triangleq \mathop{\mathbf{E}}_{P_{|\chi}} \left[\frac{1}{2}(\phi(x,1) - \phi(x,-1)) \cdot \ell \right] = \mathop{\mathbf{E}}_{D_{|\chi}} \left[\frac{\phi(x,1) - \phi(x,-1)}{2} \psi(x) \right]$$

within tolerance $\tau/2$ (since $\mathbf{E}_{P_{|\chi}}[\frac{1}{2}(\phi(x,1) + \phi(x,-1))]$ does not depend on the label and can be estimated as before). Now

$$\begin{aligned}
\mathop{\mathbf{E}}_{P^\Lambda_{|\chi}} \left[\frac{\phi(x,1) - \phi(x,-1)}{2} \cdot \ell \right] &= \mathop{\mathbf{E}}_{D_{|\chi}} \left[\frac{\phi(x,1) - \phi(x,-1)}{2} \cdot \psi(x) \cdot (1 - 2\Lambda(x)) \right] \\
&= (1-2\eta) \mathop{\mathbf{E}}_{D_{|\chi}} \left[\frac{\phi(x,1) - \phi(x,-1)}{2} \psi(x) \right] + \mathop{\mathbf{E}}_{D_{|\chi}} \left[\frac{\phi(x,1) - \phi(x,-1)}{2} \psi(x)(1 - 2(\Lambda(x) - \eta)) \right] \\
&= (1-2\eta) \mathop{\mathbf{E}}_{D_{|\chi}} \left[\frac{\phi(x,1) - \phi(x,-1)}{2} \psi(x) \right] + \tau' = (1-2\eta)\lambda + \tau',
\end{aligned}$$

where $|\tau'| \leq (1-2\eta)\tau/4$, since Λ is $(\eta, (1-2\eta)\tau/4)$-uncorrelated with ϕ and χ over P.

This means that we can estimate $\mathbf{E}_{P_{|\chi}}[\frac{1}{2}(\phi(x,1) - \phi(x,-1)) \cdot \ell]$ with tolerance $\tau/2$ by estimating $\mathbf{E}_{P^\Lambda_{|\chi}}[\frac{1}{2}(\phi(x,1) - \phi(x,-1)) \cdot \ell]$ with tolerance $(1-2\eta)\tau/4$ and then multiplying the result by $1/(1-2\eta)$. The estimation of $\mathbf{E}_{P^\Lambda_{|\chi}}[\frac{1}{2}(\phi(x,1) - \phi(x,-1)) \cdot \ell]$ with tolerance $(1-2\eta)\tau/4$ can be done exactly as in Theorem 2.1. □

An immediate implication of Theorem 2.5 is that one can simulate an active SQ algorithm A using examples corrupted by noise Λ as long as Λ is $(\eta, (1-2\eta)\tau/4)$-uncorrelated with all A's queries of tolerance τ for some fixed η.

Clearly, random classification noise of rate η has function $\Lambda(x) = \eta$ for all $x \in X$. It is therefore $(\eta, 0)$-uncorrelated with any query over any distribution. Another simple type of noise that is uncorrelated with most queries over most distributions is the one where noise function is chosen randomly so that for every point x the noise rate $\Lambda(x)$ is chosen randomly and independently from some distribution with expectation η (not necessarily the same for all points). For any fixed query and target distribution, the expected correlation is 0. If the probability mass of every single point of the domain is small enough compared to (the inverse of the logarithm of) the size of space of queries and target distributions then standard concentration inequalities will imply that the correlation will be small with high probability.

We would like to note that the noise models considered here are not directly comparable to the well-studied Tsybakov's and Massart's noise conditions [BBL05]. However, it appears that from a computational point of view our noise model is significantly more benign than these conditions as they do not impose any structure on the noise and only limit the rate.

2.3 Simple examples

Thresholds: We show that a classic example of active learning a threshold function on an interval can be easily expressed using active SQs. For simplicity and without loss of generality we can assume that the interval is $[0, 1]$ and the distribution is uniform over it. [3] Assume that we know that the threshold θ belongs to the interval $[a, b] \subseteq [0, 1]$. We ask a query $\phi(x, \ell) = (\ell + 1)/2$ with filter $\chi(x)$ which is the indicator function of the interval $[a, b]$ with tolerance $1/4$ and filter tolerance $b - a$. Let v be the response to the query. By definition, $\mathbf{E}[\chi(x)] = b - a$ and therefore we have that $|v - \mathbf{E}[\phi(x, \ell) \mid x \in [a, b]]| \leq 1/4$. Note that,

$$\mathbf{E}[\phi(x, \ell) \mid x \in [a, b]] = (b - \theta)/(b - a) .$$

We can therefore conclude that $(b - \theta)/(b - a) \in [v - 1/4, v + 1/4]$ which means that $\theta \in [b - (v + 1/4)(b - a), b - (v - 1/4)(b - a)] \cap [a, b]$. Note that the length of this interval is at most $(b - a)/2$. This means that after at most $\log_2(1/\epsilon) + 1$ iterations we will reach an interval $[a, b]$ of length at most ϵ. In each iteration only constant $1/4$ tolerance is necessary and filter tolerance is never below ϵ. A direct simulation of this algorithm can be done using $\log(1/\epsilon) \cdot \log(\log(1/\epsilon)/\delta)$ labeled examples and $\tilde{O}(1/\epsilon) \cdot \log(1/\delta)$ unlabeled samples.

Axis-aligned rectangles: Next we show that learning of thresholds can be used to obtain a simple algorithm for learning axis-aligned rectangles whose weight under the target distribution is not too small. Namely, we assume that the target function satisfies that $\mathbf{E}_D[f(x)] \geq \beta$. In the one dimensional case, we just need to learn an interval. After scaling the distribution to be uniform on $[0, 1]$ we know that the target interval $[\theta_1, \theta_2]$ has length at least β. We first need to find a point inside that interval. To do this we consider the $2/\beta$ intervals $[(i - 1)\beta/2, i\beta/2]$ for $1 \leq i \leq 2/\beta$. At least one of these intervals in fully included in $[\theta_1, \theta_2]$. Hence using an active statistical query with query function $\phi(x, \ell) = (\ell + 1)/2$ conditioned on being in interval $[(i - 1)\beta/2, i\beta/2]$ for each $1 \leq i \leq 2/\beta$ and with tolerance $1/4$ we are guaranteed to find an interval for which the answer is at least $3/4$. The midpoint of any interval for which the answer to the query is at least $3/4$ must be inside the target interval. Let the midpoint be a. We can now use two binary searches with accuracy $\epsilon/2$ to find the lower and upper endpoints of the target interval in the intervals $[0, a]$ and $[a, 1]$, respectively. This will require $2/\beta + \log_2(2/\epsilon)$ active SQs of tolerance $1/4$. As usual, the d-dimensional axis-aligned rectangles can be reduced to d interval learning problems

[3] As usual, we can bring the distribution to be close enough to this form using unlabeled samples or $O(b/\epsilon)$ target-independent queries, where b is the number of bits needed to represent our examples.

with error ϵ/d [KV94]. This gives an active statistical algorithm using $2d/\beta + \log_2(2d/\epsilon)$ active SQs of tolerance $1/4$ and filter tolerance $\geq \min\{\beta/2, \epsilon/2\}$.

$\mathbf{A^2}$: We now note that the general and well-studied A^2 algorithm of [BBL06] falls naturally into our framework. At a high level, the A^2 algorithm is an iterative, *disagreement-based* active learning algorithm. It maintains a set of surviving classifiers $C_i \subseteq C$, and in each round the algorithm asks for the labels of a few random points that fall in the current region of disagreement of the surviving classifiers. Formally, the region of disagreement $\mathrm{DIS}(\mathcal{C}_i)$ of a set of classifiers $\mathcal{C}_i$ is the of set of instances x such that for each $x \in \mathrm{DIS}(\mathcal{C}_i)$ there exist two classifiers $f, g \in \mathcal{C}_i$ that disagree about the label of x. Based on the queried labels, the algorithm then eliminates hypotheses that were still under consideration, but only if it is *statistically confident* (given the labels queried in the last round) that they are suboptimal. In essence, in each round A^2 only needs to estimate the error rates (of hypotheses still under consideration) under the conditional distribution of being in the region of disagreement. The key point is that this can be easily done via active statistical queries. Note that while the number of active statistical queries needed to do this could be large, the number of labeled examples needed to simulate these queries is essentially the same as the number of labeled examples needed by the known A^2 analyses [Han07, Han]. While in general the required computation of the disagreement region and manipulations of the hypothesis space cannot be done efficiently, efficient implementation is possible in a number of simple cases such as when the VC dimension of the concept class is a constant. It is not hard to see that in these cases the implementation can also be done using a statistical algorithm.

3 Learning halfspaces with respect to log-concave distributions

In this section we present a reduction from active learning to passive learning of homogeneous linear separators under log-concave distributions. Combining it with the SQ algorithm for learning halfspaces in the passive learning setting due to Dunagan and Vempala [DV04], we obtain the first efficient noise-tolerant active learning of homogeneous halfspaces for any isotropic log-concave distribution.

Our reduction proceeds in rounds; in round t we build a better approximation w_t to the target function by using the passive SQ learning algorithm [DV04] over a distribution D_t that is a mixture of distributions in which each component is the original distribution conditioned on being within a certain distance from the hyperplane defined by previous approximations w_i. To perform passive statistical queries relative to D_t we use active SQs with a corresponding real valued filter. Our analysis builds on the analysis of the margin-based algorithms due to [BBZ07, BL13]. However, note that in the standard margin-based analysis only points close to the current hypothesis w_t are queried in round t. As a result the analysis of our algorithm is somewhat different from that in earlier work [BBZ07, BL13].

3.1 Preliminaries

For a unit vector $v \in \mathbb{R}^d$ we denote by $h_v(x)$ the function defined by the homogenous hyperplane orthogonal to v, that is $h_v(x) = \mathsf{sign}(\langle v, x \rangle)$. Let $\mathcal{H}_d$ denote the concept class of all homogeneous halfspaces.

Definition 3.1. *A distribution over $\mathbb{R}^d$ is log-concave if $\log f(\cdot)$ is concave, where f is its associated density function. It is isotropic if its mean is the origin and its covariance matrix is the identity.*

Log-concave distributions form a broad class of distributions: for example, the Gaussian, Logistic, Exponential, and uniform distribution over any convex set are log-concave distributions.

Next, we state several simple properties of log-concave densities from [LV07].

Lemma 3.2. *There exists a constant c_m such that for any isotropic log-concave distribution D on $\mathbb{R}^d$, every unit vector v and $a \in [0,1]$,*

$$c_m a \leq \mathop{\mathbf{Pr}}_{D}[x \cdot v \in [-a,a]] \leq 2a.$$

Lemma 3.3. *There exists a constant c such that for any isotropic log-concave D on $\mathbb{R}^d$ and any two unit vectors u and v in $\mathbb{R}^d$ we have $c\theta(u,v) \leq \mathbf{E}_D[h_u(x) \neq h_v(x)]$, where $\theta(u,v)$ denotes the angle between u and v.*

For our applications the key property of log-concave densities proved in [BL13] is given in the following lemma.

Lemma 3.4. *For any constant $c_1 > 0$, there exists a constant $c_2 > 0$ such that the following holds. Let u and v be two unit vectors in $\mathbb{R}^d$, and assume that $\theta(u,v) = \alpha < \pi/2$. Assume that D is isotropic log-concave in $\mathbb{R}^d$. Then*

$$\mathop{\mathbf{Pr}}_{D}[h_u(x) \neq h_v(x) \text{ and } |v \cdot x| \geq c_2\alpha] \leq c_1\alpha. \tag{2}$$

We now state the passive SQ algorithm for learning halfspaces which will be the basis of our active SQ algorithm.

Theorem 3.5. *There exists a SQ algorithm* `LearnHS` *that learns $\mathcal{H}_d$ to accuracy $1-\epsilon$ over any distribution $D_{|\chi}$, where D is an isotropic log-concave distribution and $\chi : \mathbb{R}^d \to [0,1]$ is a filter function. Further* `LearnHS` *outputs a homogeneous halfspace, runs in time polynomial in d,$1/\epsilon$ and $\log(1/\lambda)$ and uses SQs of tolerance $\geq 1/poly(d, 1/\epsilon, \log(1/\lambda))$, where $\lambda = \mathbf{E}_D[\chi(x)]$.*

We use the Dunagan-Vempala algorithm for learning halfspaces to prove this algorithm [DV04]. The bounds on the complexity of the algorithm follow easily from the properties of log-concave distributions. Further details of the analysis and related discussion appear in Appendix A.

3.2 Active learning algorithm

Theorem 3.6. *There exists an active SQ algorithm* `ActiveLearnHS-LogC` *(Algorithm 1) that for any isotropic log-concave distribution D on $\mathbb{R}^d$, learns $\mathcal{H}_d$ over D to accuracy $1-\epsilon$ in time $poly(d, \log(1/\epsilon))$ and using active SQs of tolerance $\geq 1/poly(d, \log(1/\epsilon))$ and filter tolerance $\Omega(\epsilon)$.*

Proof. Let c be the constant given by Lemma 3.3 and let C_1 be the constant c_2 given by Lemma 3.4 when $c_1 = c/16$. Let $C_2 = c/(8C_1)$ and $C_3 = c_m \cdot C_2 \cdot c$. For every $k \leq s = \lceil \log_2(1/(c\epsilon)) \rceil$ define $b_k = C_1/2^k$. Let h_w denote the target halfspace and for any unit vector v and distribution D' we define $\mathsf{err}_{D'}(v) = \mathbf{Pr}_{D'}[h_w(x) \neq h_v(x)]$.

We define $w_0, w_1, \ldots, w_s$ via the iterative process described in Algorithm 1. Note that active SQs are used to allow us to execute `LearnHS` on D_k. That is a SQ ϕ of tolerance τ asked by `LearnHS`

Algorithm 1 `ActiveLearnHS-LogC`: Active SQ learning of homogeneous halfspaces over isotropic log-concave densities

```
%% Constants $c$, $C_1$, $C_2$ and $C_3$ are determined by the analysis.
Run LearnHS with error $C_2$ to obtain $w_0$.
for $k = 1$ to $s = \lceil \log_2(1/(c\epsilon)) \rceil$ do
    Let $b_{k-1} = C_1/2^{k-1}$
    Let $\mu_k$ equal the indicator function of being within margin $b_{k-1}$ of $w_{k-1}$
    Let $\chi_k = (\sum_{i \le k} \mu_i)/k$
    Run LearnHS over $D_k = D_{|\chi_k}$ with error $C_2/k$ by using active queries with filter $\chi_k$ and filter tolerance $C_3\epsilon$ to obtain $w_k$
end for
return $w_s$
```

(relative to D_k) is replaced with an active SQ (χ_k, ϕ) of tolerance $(C_3\epsilon, \tau)$. The response to the active SQ is a valid response to the query of `LearnHS` as long as $\mathbf{E}_D[\chi_k] \ge C_3\epsilon$. We will prove that this condition indeed holds later. We now prove by induction on k that after $k \le s$ iterations, we have that every $\hat{w}$ such that $\mathsf{err}_{D_{|\mu_i}}(\hat{w}) \le C_2$ for all $i \le k$ satisfies $\mathsf{err}_D(\hat{w}) \le c/2^k$. In addition, w_k satisfies this condition.

The case $k = 0$ follows from the properties of `LearnHS`(without loss of generality $C_2 \le c$). Assume now that the claim is true for $k-1$ ($k \ge 1$). Let $S_k^1 = \{x : |w_{k-1} \cdot x| \le b_{k-1}\}$ and $S_k^2 = \{x : |w_{k-1} \cdot x| > b_{k-1}\}$. Note that μ_{k-1} is defined to be the indicator function of S_k^1. By the inductive hypothesis we know that $\mathsf{err}_D(w_{k-1}) \le c/2^{k-1}$.

Consider an arbitrary separator $\hat{w}$ that satisfies $\mathsf{err}_{D_{|\mu_i}}(\hat{w}) \le C_2$ for all $i \le k$. By the inductive hypothesis, we know that $\mathsf{err}_D(\hat{w}) \le c/2^{k-1}$. By Lemma 3.3 we have $\theta(\hat{w}, w) \le 2^{-k+1}$ and $\theta(w_{k-1}, w) \le 2^{-k+1}$. This implies $\theta(w_{k-1}, \hat{w}) \le 2^{-k+2}$. By our choice of C_1 and Lemma 3.4, we obtain:

$$\mathop{\mathbf{Pr}}_D \left[\mathsf{sign}(w_{k-1} \cdot x) \ne \mathsf{sign}(\hat{w} \cdot x),\ x \in S_k^2\right] \le c2^{-k}/4$$

$$\mathop{\mathbf{Pr}}_D \left[\mathsf{sign}(w_{k-1} \cdot x) \ne \mathsf{sign}(w \cdot x),\ x \in S_k^2\right] \le c2^{-k}/4.$$

Therefore,

$$\mathop{\mathbf{Pr}}_D \left[\mathsf{sign}(\hat{w} \cdot x) \ne \mathsf{sign}(w \cdot x),\ x \in S_k^2\right] \le c2^{-k}/2. \tag{3}$$

By the inductive hypothesis, we also have:

$$\mathsf{err}_{D_{|\mu_k}}(\hat{w}) = \mathop{\mathbf{Pr}}_D \left[\mathsf{sign}(\hat{w} \cdot x) \ne \mathsf{sign}(w \cdot x) \mid x \in S_k^1\right] \le C_2.$$

The set S_k^1 consists of points x such that $x \cdot w_{k-1}$ fall into interval $[-b_{k-1}, b_{k-1}]$. By Lemma 3.2, this implies that $\mathbf{Pr}_D[x \in S_k^1] \le 2b_{k-1}$ and therefore,

$$\begin{aligned}\mathop{\mathbf{Pr}}_D \left[\mathsf{sign}(\hat{w} \cdot x) \ne \mathsf{sign}(w \cdot x),\ x \in S_k^1\right] &= \mathop{\mathbf{Pr}}_D \left[\mathsf{sign}(\hat{w} \cdot x) \ne \mathsf{sign}(w \cdot x) \mid x \in S_k^1\right] \cdot \mathop{\mathbf{Pr}}_D \left[x \in S_k^1\right] \\ &\le 2C_2 \cdot b_{k-1} = c2^{-k}/2. \end{aligned} \tag{4}$$

Now by combining eq. (3) and eq. (4) we get that $\mathsf{err}_D(\hat{w}) \le c/2^k$ as necessary to establish the first part of the inductive hypothesis. By the properties of `LearnHS`, $\mathsf{err}_{D_k}(w_k) \le C_2/k$. By the definition of χ_k,

$$\mathsf{err}_{D_k}(w_k) = \frac{1}{k} \sum_{i \le k} \mathsf{err}_{D_{|\mu_i}}(w_k).$$

This implies that for every $i \le k$, $\mathsf{err}_{D_{|\mu_i}}(w_k) \le C_2$, establishing the second part of the inductive hypothesis.

Inductive hypothesis immediately implies that $\mathsf{err}_D(w_s) \le \epsilon$. Therefore to finish the proof we only need to establish the bound on running time and query complexity of the algorithm. To establish the lower bound on filter tolerance we observe that by Lemma 3.2, for every $k \le s$,

$$\mathop{\mathbf{E}}_{D}[\mu_k] \ge c_m \cdot b_{k-1} = c_m \cdot C_2/2^{k-1} \ge c_m \cdot C_2 \cdot c \cdot \epsilon.$$

This implies that for every $k \le s$,

$$\mathop{\mathbf{E}}_{D}[\chi_k] = \frac{1}{k} \sum_{i \le k} \mathop{\mathbf{E}}_{D}[\mu_i] = \Omega(\epsilon).$$

Each execution of `LearnHS` is with error $C_2/k = \Omega(1/\log(1/\epsilon))$ and there are at most $O(\log(1/\epsilon))$ such executions. Now by Theorem A this implies that the total running time, number of queries and the inverse of query tolerance are upper-bounded by a polynomial in d and $\log(1/\epsilon)$. □

We remark that, as usual, we can first bring the distribution to an isotropic position by using target independent queries to estimate the mean and the covariance matrix of the distribution [LV07]. Therefore our algorithm can be used to learn halfspaces over general log-concave densities as long as the target halfspace passes through the mean of the density.

We can now apply Theorem 2.2 (or more generally Theorem 2.5) to obtain an efficient active learning algorithm for homogeneous halfspaces over log-concave densities in the presence of random classification noise of known rate. Further since our algorithm relies on `LearnHS` which can also be simulated when the noise rate is unknown (see Remark 2.3) we obtain an active algorithm which does not require the knowledge of the noise rate.

Corollary 3.7. *There exists a polynomial-time active learning algorithm that for any $\eta \in [0, 1/2)$, learns $\mathcal{H}_d$ over any log-concave distributions with random classification noise of rate η to error ϵ using poly$(d, \log(1/\epsilon), 1/(1-2\eta))$ labeled examples and a polynomial number of unlabeled samples.*

4 Learning halfspaces over the uniform distribution

The algorithm presented in Section 3 relies on the relatively involved and computationally costly algorithm of Dunagan and Vempala [DV04] for learning halfspaces over general distributions. Similarly, other active learning algorithms for halfspaces often rely on the computationally costly linear program solving [BBZ07, BL13]. For the special case of the uniform distribution on the unit sphere we now give a substantially simpler and more efficient algorithm in terms of both sample and computational complexity. This setting was studied in [BBZ07, DKM09].

We remark that the uniform distribution over the unit sphere is not log-concave and therefore, in general, an algorithm for the isotropic log-concave case might not imply an algorithm for the uniform distribution over the unit sphere. However a more careful look at the known active algorithms for the isotropic log-concave case [BBZ07, BL13] and at the algorithms in this work shows that minimization of error is performed over homogeneous halfspaces. For any homogeneous halfspace h_v, any $x \in \mathbb{R}^d$ and $\alpha > 0$, $h_v(x) = h_v(\alpha x)$. This implies that for algorithms optimizing the error over homogenous halfspaces any two spherically symmetric distributions are equivalent. In particular, the uniform distribution over the sphere is equivalent to the uniform distribution over the unit ball – an isotropic and log-concave distribution.

For a dimension d let $X = S_{d-1}$ or the unit sphere in d dimensions. Let U_d denote the uniform distribution over S_{d-1}. Unless specified otherwise, in this section all probabilities and expectations are relative to U_d. We would also like to mention explicitly the following trivial lemma relating the accuracy of an estimate of $f(\alpha)$ to the accuracy of an estimate of α.

Lemma 4.1. *Let $f : \mathbb{R} \to \mathbb{R}$ be a differentiable function and let α and $\tilde{\alpha}$ be any values in some interval $[a, b]$. Then*

$$|f(\alpha) - f(\tilde{\alpha})| \leq |\alpha - \tilde{\alpha}| \cdot \sup_{\beta \in [a,b]} |f'(\beta)| \ .$$

The lemma follows directly from the mean value theorem. Also note that given an estimate $\tilde{\alpha}$ for $\alpha \in [a, b]$ we can always assume that $\tilde{\alpha} \in [a, b]$ since otherwise $\tilde{\alpha}$ can be replaced with the closest point in $[a, b]$ which will be at least as close to α as $\tilde{\alpha}$.

We start with an outline of a non-active and simpler version of the algorithm that demonstrates one of the ideas of the active SQ algorithm. To the best of our knowledge the algorithm we present is also the simplest and most efficient (passive) SQ algorithm for the problem. A less efficient algorithm is given in [KVV10].

4.1 Learning using (passive) SQs

Let w denote the normal vector of the target hyperplane and let v be any unit vector. Instead of arguing about the disagreement between h_w and h_v directly we will use the (Euclidean) distance between v and w as a proxy for disagreement. It is easy to see that, up to a small constant factor, this distance behaves like disagreement.

Lemma 4.2. *For any unit vectors v and w,*

1. *Error is upper bounded by half the distance: $\mathbf{Pr}[h_v(x) \neq h_w(x)] \leq \|w - v\|/2$;*

2. *To estimate distance it is sufficient to estimate error: for every value $\alpha \in [0, 1]$,*

$$|\|w - v\| - 2\sin(\pi\alpha/2)| \leq \pi \left|\mathbf{Pr}[h_v(x) \neq h_w(x)] - \alpha\right| \ .$$

Proof. The angle between v and w equals $\gamma = \pi\,\mathbf{Pr}[h_v(x) \neq h_w(x)]$. Hence

$$\|w - v\| = 2\sin(\pi\gamma/2) = 2\sin(\pi\,\mathbf{Pr}[h_v(x) \neq h_w(x)]/2)$$

and $\mathbf{Pr}[h_v(x) \neq h_w(x)] = 2\arcsin(\|w-v\|/2)/\pi$. The first claim follows by observing that $\frac{2\arcsin(x/2)}{\pi x}$ is a monotone function in $[0, 2]$ and equals $1/2$ when $x = 2$.

The derivative of $2\sin(\pi x/2)$ equals at most π in absolute value and therefore the second claim follows from Lemma 4.1. □

The main idea of our algorithm is as follows. Given a current hypothesis represented by its normal vector v, we estimate the distance from the target vector w to v perturbed in the direction of each the d basis vectors. By combining the distance measurements in these directions we can find an estimate of w. Specifically, let $\{x^1, x^2, \ldots, x^d\}$ be the unit vectors of the standard basis. Let $v^i = (v + \beta x^i)/\|v + \beta x^i\|$ for some $\beta \in (0, 1/2]$. Then the distance from w to v^i can be used to (approximately) find $w \cdot x^i$. Namely, we rely on the following simple lemma.

Lemma 4.3. *Let u, v and w be any unit vectors and $\beta \in (0, 1/2]$. Then for $v' = (v+\beta u)/\|v+\beta u\|$ it holds that*

$$\langle u, w\rangle = \frac{\|v + \beta u\|(2 - \|v' - w\|^2) - 2 + \|v - w\|^2}{2\beta} \ .$$

Proof. By definition, $\langle v', w\rangle = (\langle v, w\rangle + \beta\langle u, w\rangle)/\|v + \beta u\|$ and therefore

$$\langle u, w\rangle = \frac{\|v + \beta u\|\langle v', w\rangle - \langle v, w\rangle}{\beta} \ . \tag{5}$$

For every pair of unit vectors u and u', $\langle u, u'\rangle = \frac{\|u\|^2+\|u'\|^2-\|u-u'\|^2}{2} = 1 - \|u - u'\|^2/2$ and therefore, we get that $\langle v, w\rangle = 1 - \|w - v\|^2/2$ and $\langle v', w\rangle = 1 - \|w - v'\|^2/2$. Substituting those into eq. (5) gives the claim. □

Using approximate values of $\langle x^i, w\rangle$ for all $i \in [d]$ one can easily approximate w. Our (non-active) SQ learning algorithm provides a simple example of how such reconstruction can be used for learning.

Theorem 4.4. *There exists a polynomial time SQ algorithm* `LearnHS-U` *that learns $\mathcal{H}_d$ over U_d using $d + 1$ statistical queries each of tolerance $\Omega(\epsilon/\sqrt{d})$.*

Proof. Let v be any unit vector (e.g. x^1) and for $i \in [d]$, define $v^i = (v + x^i/2)/\|v + x^i/2\|$ (that is $\beta = 1/2$). Let h_w denote the unknown target halfspace. For every v^i, we ask a statistical query with tolerance $\epsilon/(10 \cdot \pi\sqrt{d})$ to obtain α_i such that $|\mathbf{Pr}[h_{v^i} \neq h_w] - \alpha_i| \leq \epsilon/(10 \cdot \pi\sqrt{d})$ and similarly get α such that $|\mathbf{Pr}[h_v \neq h_w] - \alpha| \leq \epsilon/(10 \cdot \pi\sqrt{d})$.

We define $\gamma = 1 - (2\sin(\pi\alpha/2))^2/2$ and for every $i \in [d]$,

$$\gamma_i = \|v + x^i/2\|(2 - (2\sin(\pi\alpha_i/2))^2) - 2\gamma \ .$$

By Lemma 4.2(2), we get that $|\|v - w\| - 2\sin(\pi\alpha/2)| \leq \epsilon/(10\sqrt{d})$. Clearly $\|v - w\| \leq 2$ and therefore, by Lemma 4.1,

$$|\gamma - \langle v, w\rangle| = |(2\sin(\pi\alpha/2))^2 - \|v - w\|^2|/2 \leq 4 \cdot |(2\sin(\pi\alpha/2)) - \|v - w\||/2 \leq \epsilon/(5\sqrt{d}).$$

Note that $\|v + x^i/2\| \leq 3/2$ and therefore, by Lemmas 4.3 and 4.1

$$\left|\gamma_i - \langle x^i, w\rangle\right| \leq \frac{3}{2} \cdot 4 \cdot \epsilon/(10\sqrt{d}) + 2\epsilon/(5\sqrt{d}) = \epsilon/\sqrt{d} \ .$$

Now let

$$w' = \sum_{i \in [d]} \gamma_i x^i \ .$$

Parseval's identity implies that

$$\|w - w'\|^2 = \sum_{i \in [d]} |\gamma_i - \langle x^i, w \rangle|^2 \leq d \cdot \frac{\epsilon^2}{d} = \epsilon^2 \ .$$

Let $w^* = w'/\|w'\|$. Clearly, $\|w^* - w'\| \leq \|w - w'\| \leq \epsilon$ and therefore, by triangle inequality, $\|w^* - w\| \leq 2\epsilon$. By Lemma 4.2(1) this implies that $\mathbf{Pr}[h_{w^*} \neq h_w] \leq \epsilon$. It is easy to see that this algorithm uses $d+1$ statistical queries of tolerance $\Omega(\epsilon/\sqrt{d})$ and runs in time linear in d. □

Remark 4.5. *It is not hard to see that an even simpler way to find each of the coordinates of w is by measuring the error of each of the standard basis vectors themselves and using the fact that $\langle w, x^i \rangle = \cos(\pi \cdot \mathbf{Pr}[h_{x^i} \neq h_w])$. The variant we presented in Theorem 4.4 is more useful as a warm-up for the analysis of the active version of the algorithm.*

4.2 Active Learning of Halfspaces over U_d

Our active SQ learning algorithm is based on two main ideas. First, as in Theorem 4.4, we rely on measuring the error of hypotheses which are perturbations of the current hypothesis in the direction of each of the basis vectors. We then combine the measurements to obtain a new hypothesis. Second, as in previous active learning algorithms for the problem [DKM09, BBZ07], we only use labeled examples which are within a certain margin of the current hypothesis. The margin we use to filter the examples is a function of the current error rate. It is implicit in previous work [DKM09, BBZ07, BL13] that for an appropriate choice of margin, a constant fraction of the error region is within the margin while the total probability of a point being within the margin is linear in the error of the current hypothesis. Together these conditions allow approximating the error of a hypothesis using tolerance that has no dependence on ϵ.

We start by computing the error of a hypothesis v whose distance from the target w is Δ conditioned on being within margin γ of v. Let A_{d-1} denote the surface area of S_{d-1}. First, the surface area within margin γ of any homogenous halfspace v is

$$2\int_0^\gamma A_{d-2}(1-r^2)^{(d-2)/2} \cdot \frac{1}{\sqrt{1-r^2}} dr = 2 \cdot A_{d-2} \int_0^\gamma (1-r^2)^{(d-3)/2} dr. \qquad (6)$$

We now observe that for any v and w such that $\|v - w\| = \Delta$, $\mathbf{Pr}[h_v(x) \neq h_w(x) \mid |\langle v, x \rangle| \leq \gamma]$ is a function that depends only on Δ and γ.

Lemma 4.6. *For any $v, w \in S_{d-1}$ such that $\|v - w\| = \Delta \leq \sqrt{2}$ and $\gamma > 0$,*

$$\mathbf{Pr}[h_v(x) \neq h_w(x) \mid |\langle v, x \rangle| \leq \gamma] = \frac{A_{d-3} \int_0^\gamma (1-r^2)^{(d-3)/2} \int_{\frac{r \cdot \sqrt{2-\Delta^2}}{\Delta \cdot \sqrt{1-r^2}}}^{1} (1-s^2)^{(d-4)/2} ds \cdot dr}{A_{d-2} \int_0^\gamma (1-r^2)^{(d-3)/2} dr}.$$

We denote the probability by $\mathsf{cp}_d(\gamma, \Delta)$.

The proof of this lemma can be found in Appendix B.

The analysis of our algorithm is based on relating the effect that the change in distance of a hypothesis has on the conditional probability of error. This effect can be easily expressed using the derivative of the conditional probability as a function of the distance. Specifically we prove the following lower bound on the derivative (the proof can be found in Appendix B).

Lemma 4.7. *For $\Delta \leq \sqrt{2}$, any $d \geq 4$, and $\gamma \geq \Delta/(2\sqrt{d})$, $\partial_\Delta \mathsf{cp}_d(\gamma, \Delta) \geq 1/(56\gamma \cdot \sqrt{d})$.*

An important corollary of Lemma 4.7 is that given a hypothesis h_v and Δ, such that $\|v-w\| \leq \Delta$ we can estimate $\|v-w\|$ to accuracy ρ using an active statistical query with query tolerance $\Omega(\rho/\Delta)$. Specifically:

Lemma 4.8. *Let h_w be the target hypothesis. There is an algorithm* `MeasureDistance`(v, ρ, Δ') *that given a unit vector v and $\rho > 0$ and Δ' such that $\|v - w\| \leq \Delta'$, outputs a value $\tilde{\Delta}$ satisfying $|\|v-w\| - \tilde{\Delta}| \leq \rho$. The algorithm asks a single active SQ with filter tolerance $\tau_0 = \Omega(\Delta')$ and query tolerance of $\tau = \Omega(\rho/\Delta')$ and runs in time poly*$(\log(1/(\Delta\rho)))$.

Proof. Let $\gamma = \Delta'/(2\sqrt{d})$. This implies that $\gamma \geq \|v - w\|/(2\sqrt{d})$. Lemma 4.7 together with the mean value theorem imply that if $\Delta_1, \Delta_2 \leq \Delta'$ and $\Delta_1 - \Delta_2 \geq \rho$ then for some $\hat{\Delta} \in [\Delta_1, \Delta_2]$,

$$\mathsf{cp}_d(\gamma, \Delta_1) - \mathsf{cp}_d(\gamma, \Delta_2) = \rho \cdot \partial_\Delta \mathsf{cp}_d(\gamma, \hat{\Delta}) \geq \rho/(56\gamma\sqrt{d}) = \rho/(28\Delta') \ . \tag{7}$$

This implies that in order to estimate $\|v - w\|$ to within tolerance ρ it is sufficient to estimate $\mathsf{cp}_d(\gamma, \|v - w\|)$ to within $\rho/(28\Delta')$. To see this note that an estimate of $\mathsf{cp}_d(\gamma, \|v - w\|)$ within $\rho/(28\Delta')$ is a value μ such that $|\mu - \mathsf{cp}_d(\gamma, \|v-w\|)| \leq \rho/(28\Delta')$. Let $\tilde{\Delta}$ be such that $\mathsf{cp}_d(\gamma, \tilde{\Delta}) = \mu$. Note that Lemma 4.6 does not give an explicit mapping from $\mathsf{cp}_d(\gamma, \Delta)$ to Δ. But $\mathsf{cp}_d(\gamma, \Delta)$ is a monotone function of Δ and can be computed efficiently given Δ. Therefore we can efficiently invert $\mathsf{cp}_d(\gamma, \Delta)$ using a simple binary search. This computation will give us a value $\tilde{\Delta}$ such that $|\mathsf{cp}_d(\gamma, \tilde{\Delta}) - \mathsf{cp}_d(\gamma, \|v - w\|)| \leq \rho/(28\Delta')$. Using this together with eq. (7), we obtain that $|\tilde{\Delta} - \|v - w\|| \leq \rho$.

Let "$|\langle v, x\rangle| \leq \gamma$" denote the function (of x) that outputs 1 when the condition is satisfied and 0 otherwise. By definition, $\mathsf{cp}_d(\gamma, \|v - w\|)$ can be estimated to within $\rho/(28\Delta')$ using an active SQ ("$|\langle v, x\rangle| \leq \gamma$"$; h_v(x) \cdot \ell)$ with query tolerance of $\tau = \rho/(28\Delta')$ and filter tolerance of $\tau_0 = \Delta'/8 \leq \mathbf{Pr}_{U_d}[|\langle v, x\rangle| \leq \gamma]$ (for example see [DKM09]). □

We can now use the estimates of distance of a vector to w (the normal vector of the target hyperplane) and Lemma 4.3 to obtain a vector which is close to w. We perform this iteratively until we obtain a vector v giving a hypothesis with error of at most ϵ.

Theorem 4.9. *There exists an active statistical algorithm* `ActiveLearnHS-U` *that learns $\mathcal{H}_d$ over U_d to accuracy $1-\epsilon$, uses $(d+1)\log(1/\epsilon)$ active SQs with tolerance of $\Omega(1/\sqrt{d})$ and filter tolerance of $\Omega(1/\epsilon)$ and runs in time $d \cdot$ poly$(\log(d/\epsilon))$.*

Proof. Our algorithm works by finding a vector v that is at distance of at most 2ϵ from the normal vector of the target hyperplane which be denote by w. We do this via an iterative process such that at step t we construct a vector u^t, satisfying $\|w - u^t\| \leq 2^{-t}$. In step 1 we construct a vector u^1 such that $\|u^1 - w\| \leq 1/2$ by using the (non-active) algorithm `LearnHS-U` (Theorem 4.4) with error parameter of $\epsilon' = 1/(2\pi)$. By Lemma 4.2(2) we get that $\mathbf{Pr}[h_w \neq h_{u^1}] \leq 1/(2\pi)$ implies that $\|u^1 - w\| \leq 1/2$.

Now given a vector $v = u^t$ such that $\|w - v\| \leq 2^{-t}$ we construct a unit vector v^* such that $\|w - v^*\| \leq 2^{-t-1}$ and set $u^{t+1} = v^*$. Clearly, for $w^* = u^{\lceil \log(1/\epsilon) \rceil - 1}$ we will get $\|w - w^*\| \leq 2\epsilon$ and hence, by Lemma 4.2, $\mathbf{Pr}[h_w \neq h_{w^*}] \leq \epsilon$.

Let $\Delta' = 2^{-t}$, $\beta = 2^{-t}$ and define $v^i = (v + \beta x^i)/\|v + \beta x^i\|$. For every v^i, we know that $\|v - v^i\| \leq \beta = \Delta'$, this means $\|w - v^i\| \leq 2\Delta'$. We use `MeasureDistance` (Lemma 4.8) for v, distance

Algorithm 2 `ActiveLearnHS-U`: *Active* SQ Learning of Homogeneous Halfspaces over the Uniform Distribution

```
Run LearnHS-U with parameter error $\epsilon' = 1/(2\pi)$ to obtain $u^1$
for $t = 1$ to $\lceil \log(1/\epsilon) \rceil - 2$ do
    Set $\alpha =$MeasureDistance($u^t, \frac{1}{8 \cdot 2^t \sqrt{d}}, 2^{-t}$)
    for $i = 1$ to $d$ do
        Set $v^i = (v + 2^{-t} x^i)/\|v + 2^{-t} x^i\|$
        Set $\alpha_i =$MeasureDistance($v^i, \frac{1}{24 \cdot 2^t \sqrt{d}}, 2^{-t+1}$)
        Set $\gamma_i = 2^{t-1}(\|v + 2^{-t} x^i\|(2 - \alpha_i^2) - 2 + \alpha^2)$
    end for
    Set $v' = \sum_{i \in [d]} \gamma_i x^i$
    Set $u^{t+1} = v'/\|v'\|$
end for
return $u^{\lceil \log(1/\epsilon) \rceil - 1}$.
```

bound Δ' and accuracy parameter $\rho = \Delta'/(8\sqrt{d})$ to obtain α such that $|\|v - w\| - \alpha| \leq \Delta'/(8\sqrt{d})$. Similarly, for each $i \in [d]$ we use `MeasureDistance` for v^i, distance bound $2\Delta'$ and parameter $\rho = \Delta'/(24\sqrt{d})$ to obtain α_i such that $|\|v^i - w\| - \alpha_i| \leq \Delta'/(24\sqrt{d})$.

For $i \in [d]$, we define

$$\gamma_i = \frac{\|v + \beta x^i\|(2 - \alpha_i^2) - 2 + \alpha^2}{2\beta} \ .$$

We view γ_i as a function of α and α_i and observe that for $\alpha \in [0, \Delta']$,

$$|\partial_\alpha \gamma_i| = \left| \frac{\alpha}{\beta} \right| \leq 1$$

and for $\alpha_i \in [0, 2\Delta']$,

$$|\partial_{\alpha_i} \gamma_i| = \left| \frac{-\|v + \beta x^i\| \alpha_i}{\beta} \right| \leq 2\|v + \beta x^i\| \leq 3 \ .$$

Therefore, by Lemma 4.3 and Lemma 4.1,

$$\left| \gamma_i - \langle x^i, w \rangle \right| \leq 3 \cdot \Delta'/(24\sqrt{d}) + \Delta'/(8\sqrt{d}) = \Delta'/(4\sqrt{d}) \ .$$

Now let

$$v' = \sum_{i \in [d]} \gamma_i x^i \ .$$

Parseval's identity implies that

$$\|w - v'\|^2 = \sum_{i \in [d]} |\gamma_i - \langle x^i, w \rangle|^2 \leq d \cdot \Delta'^2/(16d) = \Delta'^2/16 \ .$$

Let $v^* = v'/\|v'\|$. Clearly, $\|v^* - v'\| \leq \|w - v'\| \leq \Delta'/4$ and therefore, by triangle inequality, $\|v^* - w\| \leq \Delta'/2 = 2^{-t-1}$.

All that is left to prove are the claimed bounds on active SQs used in this algorithm and its running time. First note that each step uses $d + 1$ active SQs and there are at most $\log(1/\epsilon)$

steps. By Lemma 4.8 the tolerance of each query at step t is $\Omega((\Delta'/\sqrt{d})/\Delta') = \Omega(1/\sqrt{d})$ and filter tolerance is $\Omega(2^{-t}) = \Omega(\epsilon)$. Lemma 4.8 together with the bound on the number of stages also imply the claimed running time bound. $\square$

An immediate corollary of Theorems 4.9 and 2.2 is an active learning algorithm for $\mathcal{H}_d$ that works in the presence of random classification noise.

Corollary 4.10. *There exists a polynomial-time active learning algorithm that given any $\eta \in [0, 1/2)$, learns $\mathcal{H}_d$ over U_d with random classification noise of rate η to error ϵ using $O((1-2\eta)^{-2} \cdot d^2 \log(1/\epsilon) \log(d \log(1/\epsilon)))$ labeled examples and $O((1-2\eta)^{-2} \cdot d^2 \cdot \epsilon^{-1} \log(1/\epsilon) \log(d \log(1/\epsilon)))$ unlabeled samples.*

4.3 Learning with unknown noise rate

One limitation of Corr. 4.10 is that the simulation requires knowing the noise rate η. We show that this limitation can be overcome by giving a procedure that approximately finds the noise rate which can then be used in the simulation of `ActiveLearnHS-U`. The idea for estimating the noise rate is to measure the agreement rate of random halfspaces with the target halfspace. The agreement rate of a fixed halfspace is a linear function of the noise rate and therefore by comparing the distribution of agreement rates in the presence of noise to the distribution of agreement rates in the noiseless case we can factor out the noise rate.

Lemma 4.11. *There is an algorithm $\mathcal{B}$ that for every unit vector w and values $\eta \in [0, 1/2)$, $\tau, \delta \in (0, 1)$, given τ, δ and access to random examples from distribution $P^\eta = (U_d, (1-2\eta)h_w)$ will, with probability at least $1 - \delta$, output a value η' such that $\frac{1-2\eta}{1-2\eta'} \in [1 - \tau, 1 + \tau]$. Further, $\mathcal{B}$ runs in time polynomial in $d, 1/\tau, 1/(1 - 2\eta)$ and $\log(1/\delta)$ and uses $O(d\tau^{-2}(1 - 2\eta)^{-2} \cdot \log(d/((1 - 2\eta)\tau\delta)))$ random examples.*

Proof. We consider the expected correlation of a randomly chosen halfspace with some fixed unknown halfspace h_u. Namely let

$$\nu = \mathop{\mathbf{E}}_{v \sim U_d} [| \mathop{\mathbf{E}}_{x \sim U_d} [h_u(x) \cdot h_v(x)]|].$$

Spherical symmetry implies that ν does not depend on u. First note that

$$\mathop{\mathbf{E}}_{x \sim U_d} [h_u(x) \cdot h_v(x)] = 2 \arcsin(\langle u, v \rangle)/\pi \geq 2\langle u, v \rangle/\pi \ .$$

A well-known fact is that for a randomly and uniformly chosen unit vector v, with probability at least 1/8, $\langle u, v \rangle \geq 1/\sqrt{d}$. This implies that with probability at least 1/8, $\mathbf{E}_{x \sim U_d}[h_u(x) \cdot h_v(x)] \geq 2/(\pi\sqrt{d})$ and hence $\nu \geq c/\sqrt{d}$ for some fixed constant c. Henceforth we can assume that ν is known exactly (as it is easy to estimate the necessary integral with the accuracy sufficient for our use).

At the same time we know that

$$\mathop{\mathbf{E}}_{P^\eta} [\ell \cdot h_v(x)] = (1 - 2\eta) \mathop{\mathbf{E}}_{U_d} [h_w(x) \cdot h_v(x)] \ .$$

This means that

$$\nu_\eta = \mathop{\mathbf{E}}_{v \sim U_d} [| \mathop{\mathbf{E}}_{(x,\ell) \sim P^\eta} [h_w(x) \cdot h_v(x)]|] = (1 - 2\eta)\nu \ .$$

Therefore in order to estimate $1-2\eta$ we estimate ν_η given samples from P^η. To estimate ν_η we draw a set of random unit vectors V and a set S of random examples from P. For each $v \in V$ we estimate $|\mathbf{E}_{(x,\ell)\sim P^\eta}[h_w(x)\cdot h_v(x)]|$ using the random examples in S and let α_v denote the corresponding estimate. The average of α_v's is an estimate of ν_η. Chernoff-Hoeffding bounds imply that for $t_1(\theta,\delta') = O(\theta^{-2}\log(1/\delta'))$ and $t_2(\theta,\delta') = O(\theta^{-2}\log(t_1(\theta,\delta')/\delta'))$, the estimation procedure above with $|V| = t_1(\theta,\delta')$ and $S = t_2(\theta,\delta')$, will with probability $1-\delta'$ return an estimate of ν_η within θ.

We first find a good lower bound on $1-2\eta$ via a simple guess, estimate and double process. For $i = 1,2,3,\ldots$ we estimate ν_η with tolerance $\nu\cdot 2^{-i}$ and confidence $\delta/2^{-i-1}$ until we get an estimate that equals at least $2\cdot\nu\cdot 2^{-i}$. Let i_η denote the first step at which this condition was satisfied. We claim that with probability at least $1-\delta/2$,

$$2^{-i_\eta} \le (1-2\eta) \le 3\cdot 2^{-i_\eta+1}.$$

First we know that the estimates are successful for every i with probability at least $1-\delta/2$. The stopping condition implies that $\nu_\eta \ge \nu\cdot 2^{-i_\eta}$ and in particular, $(1-2\eta)\ge 2^{-i_\eta}$. Now to prove that $(1-2\eta) \le 3\cdot 2^{-i_\eta+1}$ we show that $i_\eta \le \lceil \log(3/(1-2\eta))\rceil$. This is true since for $k = \lceil\log(3/(1-2\eta))\rceil$ we get that $(1-2\eta)\ge 3\cdot 2^{-k}$ and hence $\nu_\eta = (1-2\eta)\nu \ge 3\nu 2^{-k}$. Therefore an estimate of ν_η with tolerance $\nu\cdot 2^{-k}$ must be at least $2\nu\cdot 2^{-k}$. This means that $i_\eta \le k$.

Given a lower bound of 2^{-i_η}, we estimate ν_η to accuracy $\nu\tau 2^{-i_\eta}/2 \le (1-2\eta)\nu\tau/2$ with confidence $1-\delta/2$ and let ν'_η denote the estimate. We set $1-2\eta' = \nu'/\nu$. We first note that $|(1-2\eta')-(1-2\eta)| \le 2^{-i_\eta}\tau/2 \le (1-2\eta)\tau/2$ and therefore

$$\frac{1-2\eta}{1-2\eta'} \in \left[\frac{1}{1+\frac{\tau}{2}}, \frac{1}{1-\frac{\tau}{2}}\right] \subseteq [1-\tau, 1+\tau]\ .$$

Using the fact that $\nu \ge c/\sqrt{d}$ and $i_\eta \le \lceil\log(3/(1-2\eta))\rceil$ we can conclude that the first step of the estimation procedure requires $O(d(1-2\eta)^{-2}\cdot\log\frac{d}{(1-2\eta)\delta})$ examples and the second step requires $O(d\tau^{-2}(1-2\eta)^{-2}\cdot\log\frac{d}{(1-2\eta)\delta\tau})$ examples. The straightforward implementation has running time of $O(d^3\tau^{-4}(1-2\eta)^{-4}\cdot\log\frac{d}{(1-2\eta)\delta\tau})$. □

Note that by Theorem 4.9 our algorithm for learning halfspaces uses $\tau = \Omega(1/\sqrt{d})$. We can apply Lemma 4.11 together with Remark 2.3 and Corollary 4.10 to obtain a version of the algorithm that does not require the knowledge of η.

Corollary 4.12. *There exists a polynomial-time active learning algorithm that for any $\eta \in [0,1/2)$, learns $\mathcal{H}_d$ over U_d with random classification noise of rate η to error ϵ using*

$$O\left((1-2\eta)^{-2}\cdot d^2\left(\log\frac{d}{(1-2\eta)\delta\tau} + \log(1/\epsilon)\log(d\log(1/\epsilon))\right)\right)$$

labeled examples and

$$O((1-2\eta)^{-2}\cdot d^2\cdot\epsilon^{-1}\log(1/\epsilon)\log(d\log(1/\epsilon)))$$

unlabeled samples.

5 Differentially-private active learning

In this section we show that active SQ learning algorithms can also be used to obtain differentially private active learning algorithms. We assume that a learner has full access to unlabeled portion of some database of n examples $S \subseteq X \times Y$ which correspond to records of individual participants in the database. In addition, for every element of the database S the learner can request the label of that element. As usual, the goal is to minimize the number of label requests. In addition, we would like to preserve the *differential privacy* of the participants in the database, a now-standard notion of privacy introduced in [DMNS06]. A simple scenario in which active differentially-private learning could be valuable is medical research in which the goal is to create an automatic predictor of whether a person has certain medical condition. It is often the case that while many unlabeled patient records are available, discovering the label requires work by a medical expert or an expensive test (or both). In such a scenario an active learning algorithm could significantly reduce costs of producing a good predictor of the condition while differential privacy ensures that the predictor does not reveal any information about the patients whose data was used for the algorithm.

Formally, for some domain $X \times Y$, we will call $S \subseteq X \times Y$ a *database*. Databases $S, S' \subset X \times Y$ are *adjacent* if one can be obtained from the other by modifying a single element. Here we will always have $Y = \{-1, 1\}$. In the following, A is an algorithm that takes as input a database D and outputs an element of some finite set R.

Definition 5.1 (Differential privacy [DMNS06])**.** *A (randomized) algorithm $A : 2^{X\times Y} \rightarrow R$ is α-*differentially-private *if for all $r \in R$ and every pair of adjacent databases S, S', we have* $\Pr[A(S) = r] \leq e^{\epsilon} \Pr[A(S') = r]$.

Here we consider algorithms that operate on S in an active way. That is the learning algorithm receives the unlabeled part of each point in S as an input and can only obtain the label of a point upon request. The total number of requests is the label complexity of the algorithm. We note the definition of differential privacy we use does not make any distinction between the entries of the database for which labels were requested and the other ones. In particular, the privacy of all entries is preserved. Further, in our setting the indices of entries for which the labels are requested are not a part of the output of the algorithm and hence do not need to be differentially private.

As first shown by Blum *et al.* [BDMN05], SQ algorithms can be automatically translated into differentially-private[4] algorithms. We now show that, analogously, active SQ learning algorithms can be automatically transformed into differentially-private active learning algorithms.

Theorem 5.2. *Let A be an algorithm that learns a class of functions H to accuracy $1-\epsilon$ over distribution D using M_1 active SQs of tolerance τ and filter tolerance τ_0 and M_2 target-independent queries of tolerance τ_u. There exists a learning algorithm A' that given $\alpha > 0, \delta > 0$ and active access to database $S \subseteq X \times \{-1, 1\}$ is α-differentially-private and uses at most $O([\frac{M_1}{\alpha\tau} + \frac{M_1}{\tau^2}]\log(M_1/\delta))$ labels. Further, for some $n = O([\frac{M_1}{\alpha\tau_0\tau} + \frac{M_1}{\tau_0\tau^2} + \frac{M_2}{\alpha\tau_u} + \frac{M_2}{\tau_u^2}]\log((M_1+M_2)/\delta))$, if S consists of at least n examples drawn randomly from D then with, probability at least $1-\delta$, A' outputs a hypothesis with accuracy $\geq 1-\epsilon$ (relative to distribution D). The running time of A' is the same as the running time of A plus $O(n)$.*

[4] In [BDMN05] a related but different definition of privacy was used. However, as pointed out in [KLN+11] the same translation can be used to achieve differential privacy.

Proof. We first consider the active SQs of A. We will answer each such query using a disjoint set of $O([\frac{1}{\alpha\tau_0\tau} + \frac{1}{\tau_0\tau^2}]\log(M_1/\delta))$ unlabeled examples. The subset T of these examples that satisfy the filter will be queried for their labels and used to compute an answer to the statistical query (by taking the empirical average in the usual way). Additional noise drawn from a Laplace distribution will then be added to the answer in order to preserve privacy.

We begin by analyzing the amount of noise needed to achieve the desired privacy guarantee. First, since each query is being answered using a disjoint set of examples, changing any given example can affect the answer to at most one query; so, it suffices to answer each query with α-differential privacy. Second, modifying any given example can change the empirical answer to a query by at most $1/|T|$ (there are three cases: the example was already in T and remains in T after modification, the example was in T and is removed from T due to the modification, or the example was not in T and is added to T due to the modification; each changes the empirical answer by at most $1/|T|$). Therefore, α-privacy can be achieved by adding a quantity ξ selected from a Laplace distribution of width $O(\frac{1}{\alpha|T|})$ to the empirical answer over the labeled sample. Finally, we solve for the size of T needed to ensure that with sufficiently high probability $|\xi| \leq \tau/2$ so that the effect on the active SQ after correction for noise is at most $\tau/2$. Specifically, the Laplace distribution has the property that with probability at least $1 - \delta'$, the magnitude of ξ is at most $O(\frac{1}{\alpha|T|}\log(1/\delta'))$. Setting this to $\tau/2$ and using $\delta' = \delta/(6M_1)$ we have that privacy α can be guaranteed with perturbation at most $\tau/2$ as long as we have $|T| \geq c(\frac{1}{\alpha\tau}\log(M_1/\delta))$ for sufficiently large constant c.

Next, we also need to ensure that T is large enough so that with probability at least $1 - \delta'$, even without the added Laplace noise, the empirical average of the query function over T is within $\tau/2$ of the true value. By Hoeffding bounds, this is ensured if $|T| \geq c(\frac{1}{\tau^2}\log(M_1/\delta))$ for sufficiently large constant c.

Finally, we need the unlabeled sample to be large enough so that with probability at least $1-\delta'$, the labeled sample T will satisfy both the above conditions. By Hoeffding bounds, this is ensured by an unlabeled sample of size $O(\frac{1}{\tau_0}[\frac{1}{\alpha\tau} + \frac{1}{\tau^2}]\log(M_1/\delta))$.

The above analysis was for each active SQ. There are M_1 active SQs in total so the total sample size is a factor M_1 larger, and by a union bound over all M_1 queries we have that with probability at least $1 - \delta/2$, all are answered within their desired tolerance levels.

Now, we analyze the M_2 target-independent queries. Here, by standard analysis (which is also a special case of the analysis above), we get that it is sufficient to use $O([\frac{1}{\alpha\tau_u} + \frac{1}{\tau_u^2}]\log(M_2/\delta))$ unlabeled samples to answer all the queries with probability at least $1-\delta/2$. Finally, summing up the sample sizes and applying a union bound over the failure probabilities we get the claimed bounds on the sample complexity and running time. We remark that the algorithm is α-differentially-private even when the samples are not drawn from distribution D. □

The simulation above can be easily made tolerant to random classification (or uncorrelated) noise in exactly the same way as in Theorem 2.2.

In our setting it is also natural to treat the privacy of labeled and unlabeled parts differently. For much the same reason that unlabeled data is often much more plentiful than labeled data, in many cases the label information is much more sensitive, in a privacy sense, than the unlabeled feature vector. For example the unlabeled data may be fully public (obtained by crawling the web or from a public address-book) and the labels obtained from a questionnaire. To reflect this one can define two privacy parameters α_ℓ and α with α_ℓ denoting the (high) sensitivity of the labeled information and α denoting the (lower) sensitivity of the feature vector alone. More formally, in addition to requiring α-differential privacy we can require α_ℓ-differential privacy on databases which differ only in a single label (for $\alpha_\ell < \alpha$). A special case of this model where only label privacy matters was studied in [CH11] (a model with a related but weaker requirement in which labeled points are private and unlabeled are not was recently considered in [JMP13]). It is not hard to see that with this definition our analysis will give an algorithm that uses $O([\frac{M_1}{\alpha_\ell \tau} + \frac{M_1}{\tau^2}]\log(M_1/\delta))$ labels and requires a database of size n for some $n = \Omega([\frac{M_1}{\alpha_\ell \tau_0 \tau} + \frac{M_1}{\tau_0 \tau^2} + \frac{M_2}{\alpha \tau_u} + \frac{M_2}{\tau_u^2}]\log((M_1+M_2)/\delta))$. Note that in this result the privacy constraint on labels does not affect the number of samples required to simulate target-independent queries.

Improvement over passive differentially-private learning An immediate consequence of Theorem 5.2 is that for learning of homogeneous halfspaces over uniform or log-concave distributions we can obtain differential privacy while essentially preserving the label complexity. For example, by combining Theorems 5.2 and 4.9, we can efficiently and differentially-privately learn homogeneous halfspaces under the uniform distribution with privacy parameter α and error parameter ϵ by using only $\tilde{O}(d\sqrt{d}\log(1/\epsilon))/\alpha + d^2\log(1/\epsilon))$ labels. However, it is known that any passive learning algorithm, even ignoring privacy considerations and noise requires $\Omega\left(d/\epsilon\right)$ labeled examples [Lon95]. So for $\alpha \geq 1/\sqrt{d}$ and small enough ϵ we get better label complexity.

6 Discussion

We described a framework for designing efficient active learning algorithms that are tolerant to random classification noise. We used our framework to obtain the first computationally-efficient algorithm for actively learning homogeneous linear separators over log-concave distributions with exponential improvement in the dependence on the error ϵ over its passive counterpart. In addition, we showed that our algorithms can be automatically converted to efficient active differentially-private algorithms.

Our work suggests that, as in passive learning, active statistical algorithms might be essentially as powerful as example-based efficient active learning algorithms. It would be interesting to find more general evidence supporting this claim or, alternatively, a counterexample. An important aspect of (passive) statistical learning algorithms is that it is possible to prove unconditional lower bounds on such algorithms using SQ dimension [BFJ$^+$94] and its extensions. It would be interesting to develop an active analogue of these techniques and give meaningful lower bounds based on them. This could provide a useful tool for understanding the sample complexity of differentially private active learning algorithms.

Acknowledgments

We thank Avrim Blum and Santosh Vempala for useful discussions. This work was supported in part by NSF grants CCF-0953192, CCF-110128, and CCF 1422910, AFOSR grant FA9550-09-1-0538, ONR grant N00014-09-1-0751, and a Microsoft Research Faculty Fellowship.

References

[ABL14] P. Awasthi, M.-F. Balcan, and P. Long. The power of localization for efficiently learning linear separators with noise. In *Proc.46th ACM Symposium on Theory of Computing*, 2014.

[AD98] J. Aslam and S. Decatur. Specification and simulation of statistical query algorithms for efficiency and noise tolerance. *JCSS*, 56:191–208, 1998.

[AL88] D. Angluin and P. Laird. Learning from noisy examples. *Machine Learning*, 2:343–370, 1988.

[BBL05] O. Bousquet, S. Boucheron, and G. Lugosi. Theory of Classification: A Survey of Recent Advances. *ESAIM: Probability and Statistics*, 9:323–375, 2005.

[BBL06] M.-F. Balcan, A. Beygelzimer, and J. Langford. Agnostic active learning. In *ICML*, 2006.

[BBZ07] M.-F. Balcan, A. Broder, and T. Zhang. Margin based active learning. In *COLT*, pages 35–50, 2007.

[BDL09] A. Beygelzimer, S. Dasgupta, and J. Langford. Importance weighted active learning. In *ICML*, pages 49–56, 2009.

[BDMN05] A. Blum, C. Dwork, F. McSherry, and K. Nissim. Practical privacy: the SuLQ framework. In *Proceedings of PODS*, pages 128–138, 2005.

[BEHW89] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, 36(4):929–965, 1989.

[BF02] N. Bshouty and V. Feldman. On using extended statistical queries to avoid membership queries. *JMLR*, 2:359–395, 2002.

[BFJ$^+$94] A. Blum, M. Furst, J. Jackson, M. Kearns, Y. Mansour, and S. Rudich. Weakly learning DNF and characterizing statistical query learning using Fourier analysis. In *STOC*, pages 253–262, 1994.

[BFKV97] A. Blum, A. Frieze, R. Kannan, and S. Vempala. A polynomial time algorithm for learning noisy linear threshold functions. *Algorithmica*, 22(1/2):35–52, 1997.

[BH12] M.-F. Balcan and S. Hanneke. Robust interactive learning. In *COLT*, 2012.

[BHLZ10] A. Beygelzimer, D. Hsu, J. Langford, and T. Zhang. Agnostic active learning without constraints. In *NIPS*, 2010.

[BHW08] M.-F. Balcan, S. Hanneke, and J. Wortman. The true sample complexity of active learning. In *COLT*, 2008.

[BL13] M.-F. Balcan and P. M. Long. Active and passive learning of linear separators under log-concave distributions. *COLT*, 2013.

[Byl94] T. Bylander. Learning linear threshold functions in the presence of classification noise. In *Proceedings of COLT*, pages 340–347, 1994.

[CGZ10] N. Cesa-Bianchi, C. Gentile, and L. Zaniboni. Learning noisy linear classifiers via adaptive and selective sampling. *Machine Learning*, 2010.

[CH11] K. Chaudhuri and D. Hsu. Sample complexity bounds for differentially private learning. *JMLR - COLT Proceedings*, 19:155–186, 2011.

[CKL$^+$06] C. Chu, S. Kim, Y. Lin, Y. Yu, G. Bradski, A. Ng, and K. Olukotun. Map-reduce for machine learning on multicore. In *Proceedings of NIPS*, pages 281–288, 2006.

[CN07] R. Castro and R. Nowak. Minimax bounds for active learning. In *COLT*, 2007.

[Das05] S. Dasgupta. Coarse sample complexity bounds for active learning. In *NIPS*, volume 18, 2005.

[Das11] S. Dasgupta. Active learning. *Encyclopedia of Machine Learning*, 2011.

[DGS12] O. Dekel, C. Gentile, and K. Sridharan. Selective sampling and active learning from single and multiple teachers. *JMLR*, 2012.

[DH08] S. Dasgupta and D. Hsu. Hierarchical sampling for active learning. In *ICML*, pages 208–215, 2008.

[DHM07] S. Dasgupta, D.J. Hsu, and C. Monteleoni. A general agnostic active learning algorithm. *NIPS*, 20, 2007.

[DKM09] S. Dasgupta, A. Tauman Kalai, and C. Monteleoni. Analysis of perceptron-based active learning. *Journal of Machine Learning Research*, 10:281–299, 2009.

[DMNS06] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, pages 265–284, 2006.

[DV04] J. Dunagan and S. Vempala. A simple polynomial-time rescaling algorithm for solving linear programs. In *STOC*, pages 315–320, 2004.

[Fel12] V. Feldman. A complete characterization of statistical query learning with applications to evolvability. *Journal of Computer System Sciences*, 78(5):1444–1459, 2012.

[FGR$^+$13] V. Feldman, E. Grigorescu, L. Reyzin, S. Vempala, and Y. Xiao. Statistical algorithms and a lower bound for detecting planted cliques. In *ACM STOC*, 2013.

[FSST97] 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.

[GSSS13] A. Gonen, S. Sabato, and S. Shalev-Shwartz. Efficient pool-based active learning of halfspaces. In *ICML*, 2013.

[Han] S. Hanneke. Theory of active learning.

[Han07] S. Hanneke. A bound on the label complexity of agnostic active learning. In *ICML*, 2007.

[JMP13] G. Jagannathan, C. Monteleoni, and K. Pillaipakkamnatt. A semi-supervised learning approach to differential privacy. In *Proceedings of the 2013 IEEE International Conference on Data Mining Workshops (ICDMW), IEEE Workshop on Privacy Aspects of Data Mining (PADM)*, 2013.

[Kea98] M. Kearns. Efficient noise-tolerant learning from statistical queries. *JACM*, 45(6):983–1006, 1998.

[KLN$^+$11] Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM J. Comput.*, 40(3):793–826, June 2011.

[Kol10] V. Koltchinskii. Rademacher complexities and bounding the excess risk in active learning. *JMLR*, 11:2457–2485, 2010.

[KV94] M. Kearns and U. Vazirani. *An introduction to computational learning theory.* MIT Press, Cambridge, MA, 1994.

[KVV10] V. Kanade, L. G. Valiant, and J. Wortman Vaughan. Evolution with drifting targets. In *Proceedings of COLT*, pages 155–167, 2010.

[Lon95] P. M. Long. On the sample complexity of PAC learning halfspaces against the uniform distribution. *IEEE Transactions on Neural Networks*, 6(6):1556–1559, 1995.

[LV07] L. Lovász and S. Vempala. The geometry of logconcave functions and sampling algorithms. *Random Struct. Algorithms*, 30(3):307–358, 2007.

[MN98] A. McCallum and K. Nigam. Employing EM in pool-based active learning for text classification. In *ICML*, pages 350–358, 1998.

[Ros58] F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–407, 1958.

[RR11] M. Raginsky and A. Rakhlin. Lower bounds for passive and active learning. In *NIPS*, pages 1026–1034, 2011.

[Val84] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[Vap98] V. Vapnik. *Statistical Learning Theory.* Wiley-Interscience, 1998.

[Vem13] S. Vempala. Personal communication, 2013.

[Wan11] L. Wang. Smoothness, Disagreement Coefficient, and the Label Complexity of Agnostic Active Learning. *JMLR*, 2011.

A Passive SQ learning of halfspaces

The first SQ algorithm for learning general halfspaces was given by Blum *et al.* [BFKV97]. This algorithm requires access to unlabeled samples from the unknown distribution and therefore is only label-statistical. This algorithm can be used as a basis for our active SQ algorithm but the resulting active algorithm will also be only label-statistical. As we have noted in Section 2, this is sufficient to obtain our RCN tolerant active learning algorithm given in Cor. 3.7. However our differentially-private simulation needs the algorithm to be (fully) statistical. Therefore we base our algorithm on the algorithm of Dunagan and Vempala for learning halfspaces [DV04]. While [DV04] does not contain an explicit statement of the SQ version of the algorithm it is known and easy to verify that the algorithm has a SQ version [Vem13]. This follows from the fact that the algorithm in [DV04] relies on a combination on the Perceptron [Ros58] and the modified Perceptron algorithms [BFKV97] both of which have SQ versions [Byl94, BFKV97]. Another small issue that we need to take care of to apply the algorithm is that the running time and tolerance of the algorithm depend polynomially (in fact, linearly) on $\log(1/\rho_0)$, where ρ_0 is the *margin* of the points given to the algorithm. Namely, $\rho_0 = \min_{x\in S} \frac{|w\cdot x|}{\|x\|}$, where h_w is the target homogeneous halfspace and S is the set of points given to the algorithm. We are dealing with continuous distributions for which the margin is 0 and therefore we make the following observation. In place of ρ_0 we can use any margin ρ_1 such that the probability of being within margin $\leq \rho_1$ around the target hyperplane is small enough that it can be absorbed into the tolerance of the statistical queries of the Dunagan-Vempala algorithm for margin ρ_1. Formally,

Definition A.1. *For positive $\delta < 1$ and distribution D, we denote*

$$\gamma(D,\delta) = \inf_{\|w\|=1} \sup_{\gamma>0} \left\{ \gamma \;\middle|\; \Pr_D\left[\frac{|w\cdot x|}{\|x\|} \leq \gamma\right] \leq \delta \right\},$$

namely the smallest value of γ such that for every halfspace h_w, γ is the largest such that the probability of being within margin γ of h_w under D is at most δ. Let $\tau_{\mathtt{DV}}(\rho,\epsilon)$ be the tolerance of the SQ version of the Dunagan-Vempala algorithm when the initial margin is equal to ρ and error is set to ϵ. Let

$$\rho_1(D,\epsilon) = \frac{1}{2}\sup_{\rho\geq 0}\{\rho \mid \gamma(D, \tau_{\mathtt{DV}}(\rho,\epsilon/2)/3) \geq \rho\}.$$

Now, for $\rho_1 = \rho_1(D,\epsilon)$ we know that $\gamma(D, \tau_{\mathtt{DV}}(\rho,\epsilon/2)/3) \geq \rho$. Let D' be defined as distribution D conditioned on having margin at least ρ around the target hyperplane h_w. By the definition of the function γ, the probability of being within margin $\leq \rho$ is at most $\tau_{\mathtt{DV}}(\rho,\epsilon/2)/3$. Therefore for any query function $g : X \times \{-1,1\} \to [-1,1]$,

$$\left| \mathop{\mathbf{E}}_{D}[g(x,h_w(x))] - (1-\tau_{\mathtt{DV}}(\rho,\epsilon/2)/3) \mathop{\mathbf{E}}_{D'}[g(x,h_w(x))] \right| \leq \tau_{\mathtt{DV}}(\rho,\epsilon/2)/3$$

and hence $|\mathbf{E}_D[g(x,h_w(x))] - \mathbf{E}_{D'}[h_w(x,f(x))]| \leq 2\tau_{\mathtt{DV}}(\rho,\epsilon/2)/3$. This implies that we can obtain an answer to any SQ relative to D' with tolerance $\tau_{\mathtt{DV}}(\rho,\epsilon/2)$ by using the same SQ relative to D with tolerance $\tau_{\mathtt{DV}}(\rho,\epsilon/2)/3$. This means that by running the Dunagan-Vempala algorithm in this way we will obtain a hypothesis with error at most $\epsilon/2$ relative to D'. This hypothesis has error at most $\epsilon/2 + 2\tau_{\mathtt{DV}}(\rho,\epsilon/2)/3$ which, without loss of generality, is at most ϵ. Combining these observations about the Dunagan-Vempala algorithm, we obtain the following statement.

Theorem A.2 ([DV04])**.** *There exists a SQ algorithm* `LearnHS-DV` *that learns $\mathcal{H}_d$ to accuracy $1-\epsilon$ over any distribution D. Further* `LearnHS` *outputs a homogeneous halfspace, runs in time polynomial in d,$1/\epsilon$ and $\log(1/\rho_1)$ and uses SQs of tolerance $\geq 1/poly(d, 1/\epsilon, \log(1/\rho_1))$, where $\rho_1 = \rho_1(D, \epsilon)$.*

To apply Theorem A.2 we need to obtain bounds on $\rho_1(D, \epsilon)$ for any distribution D on which we might run the Dunagan-Vempala algorithm.

Lemma A.3. *Let D be an isotropic log-concave distribution. Then for any $\delta \in (0, 1/20)$, $\gamma(D, \delta) \geq \delta/(6\ln(1/\delta))$.*

Proof. Let $\gamma \in (0, 1/16)$ and w be any unit vector. We first upper-bound $\Pr_D\left[\frac{|w\cdot x|}{\|x\|} \leq \gamma\right]$.

$$\begin{aligned}\Pr_D\left[\frac{|w\cdot x|}{\|x\|} \leq \gamma\right] &\leq \Pr_D\left[\|x\| \leq \ln(1/\gamma) \text{ and } \frac{|w\cdot x|}{\|x\|} \leq \gamma\right] + \Pr_D\left[\|x\| > \ln(1/\gamma)\right] \\ &\leq \Pr_D\left[|w\cdot x| \leq \gamma \cdot \ln(1/\gamma)\right] + \Pr_D\left[\|x\| > \ln(1/\gamma)\right]. \qquad (8)\end{aligned}$$

By Lemma 5.7 in [LV07], for an isotropic log-concave D and any $R > 1$, $\Pr_D[\|x\| > R] < e^{-R+1}$. Therefore

$$\Pr_D\left[\|x\| > \ln(1/\gamma)\right] \leq e \cdot \gamma.$$

Further, by Lemma 3.2,

$$\Pr_D\left[|w\cdot x| \leq \gamma \cdot \ln(1/\gamma)\right] \leq 2\gamma \cdot \ln(1/\gamma).$$

Substituting, these inequalities into eq. (8) we obtain that for $\gamma \in (0, 1/16)$,

$$\Pr_D\left[\frac{|w\cdot x|}{\|x\|} \leq \gamma\right] \leq 2\gamma \cdot \ln(1/\gamma) + e\cdot\gamma \leq 3\gamma \cdot \ln(1/\gamma).$$

This implies that for $\gamma = \delta/(6\ln(1/\delta))$ and any unit vector w,

$$\Pr_D\left[\frac{|w\cdot x|}{\|x\|} \leq \gamma\right] \leq 3\delta/(6\ln(1/\delta)) \cdot (\ln(1/\delta) + \ln(6\ln(1/\delta)) \leq \delta,$$

where we used that for $\delta < 1/20$, $6\ln(1/\delta) \leq 1/\delta$. By definition of $\gamma(D, \delta)$, this implies that $\gamma(D, \delta) \geq \delta/(6\ln(1/\delta))$. □

We are now ready to prove Theorem A. There exists a SQ algorithm `LearnHS` that learns $\mathcal{H}_d$ to accuracy $1-\epsilon$ over any distribution $D_{|\chi}$, where D is an isotropic log-concave distribution and $\chi : \mathbb{R}^d \to [0, 1]$ is a filter function. Further `LearnHS` outputs a homogeneous halfspace, runs in time polynomial in d,$1/\epsilon$ and $\log(1/\lambda)$ and uses SQs of tolerance $\geq 1/\mathrm{poly}(d, 1/\epsilon, \log(1/\lambda))$, where $\lambda = \mathbf{E}_D[\chi(x)]$.

Proof of Thm. A. To prove the theorem we bound $\rho_1 = \rho_1(D_{|\chi}, \epsilon)$ and then apply Theorem A.2. We first observe that for any event Λ,

$$\mathop{\mathbf{Pr}}_{D_{|\chi}}[\Lambda] \leq \mathop{\mathbf{Pr}}_{D}[\Lambda] / \mathop{\mathbf{E}}_{D}[\chi].$$

Applying this to the event $\frac{|w \cdot x|}{\|x\|} \leq \gamma$ in Definition A.1 we obtain that $\gamma(D_{|\chi}, \delta) \geq \gamma(D, \delta \cdot \mathbf{E}_D[\chi])$. By Lemma A.3, we get that $\gamma(D_{|\chi}, \delta) = \Omega(\lambda\delta/\log(1/(\lambda\delta)))$.

In addition, by Theorem A.2, $\tau_{\mathtt{DV}}(\rho, \epsilon) \geq 1/p(d, 1/\epsilon, \log(1/\rho))$ for some polynomial p. This implies that

$$\gamma(D_{|\chi}, \tau_{\mathtt{DV}}(\rho, \epsilon/2)/3) \leq \gamma\left(D_{|\chi}, \Omega\left(\frac{1}{p(d, 1/\epsilon, \log(1/\rho))}\right)\right) = \tilde{\Omega}\left(\frac{\lambda}{p(d, 1/\epsilon, \log(1/\rho))}\right).$$

Therefore, we will obtain that,

$$\rho_1(D_{|\chi}, \epsilon) = \tilde{\Omega}\left(\frac{\lambda}{p(d, 1/\epsilon, 1)}\right).$$

□

By plugging this bound into Theorem A.2 we obtain the claim.

B Proofs from Section 4

We now prove Lemmas 4.6 and 4.7 which we restate for convenience.

Lemma B.1 (Lem. 4.6 restated). *For any $v, w \in S_{d-1}$ such that $\|v - w\| = \Delta \leq \sqrt{2}$ and $\gamma > 0$,*

$$\mathbf{Pr}[h_v(x) \neq h_w(x) \mid |\langle v, x\rangle| \leq \gamma] = \frac{A_{d-3}\int_0^\gamma (1-r^2)^{(d-3)/2} \int_{\frac{r\cdot\sqrt{2-\Delta^2}}{\Delta\cdot\sqrt{1-r^2}}}^1 (1-s^2)^{(d-4)/2} ds \cdot dr}{A_{d-2}\int_0^\gamma (1-r^2)^{(d-3)/2} dr}.$$

We denote the probability by $\mathsf{cp}_d(\gamma, \Delta)$.

Proof. By using spherical symmetry, we can assume without loss of generality that $v = (1, 0, 0, \ldots, 0)$ and $w = (\sqrt{1 - \Delta^2/2}, \Delta/\sqrt{2}, 0, 0, \ldots, 0)$. We now examine the surface area of the points that satisfy $h_w(x) = -1$ and $0 \leq \langle v, x\rangle \leq \gamma$ (which is a half of the error region at distance at most γ from v). To compute it we consider the points on S_{d-1} that satisfy $\langle v, x\rangle = r$. These points form a hypersphere σ of dimension $d - 2$ and radius $\sqrt{1 - r^2}$. In this hypersphere points that satisfy $h_w(x) = -1$ are points $(r, s, x_3, .., x_d) \in S_d$ for which $r\sqrt{1 - \Delta^2/2} + s\Delta/\sqrt{2} \leq 0$. In other words, $s \geq r\sqrt{2 - \Delta^2}/\Delta$ or points of σ which are at least $r\sqrt{2 - \Delta^2}/\Delta$ away from hyperplane $(0, 1, 0, 0, \ldots, 0)$ passing through the origin of σ (also referred to as a hyperspherical cap). As in the equation (6), we obtain that its $d - 2$-dimensional surface area is:

$$(1-r^2)^{(d-2)/2} \int_{\frac{r\cdot\sqrt{2-\Delta^2}}{\Delta\cdot\sqrt{1-r^2}}}^1 A_{d-3}(1-s^2)^{(d-4)/2} ds$$

Integrating over all r from 0 to γ gives the surface area of the region $h_w(x) = -1$ and $0 \leq \langle v, x\rangle \leq \gamma$:

$$\int_0^\gamma (1-r^2)^{(d-3)/2} \int_{\frac{r\cdot\sqrt{2-\Delta^2}}{\Delta\cdot\sqrt{1-r^2}}}^1 A_{d-3}(1-s^2)^{(d-4)/2} ds \cdot dr.$$

Hence the conditional probability is as claimed. □

Lemma B.2 (Lem. 4.7 restated)**.** *For $\Delta \leq \sqrt{2}$, any $d \geq 4$, and $\gamma \geq \Delta/(2\sqrt{d})$, $\partial_\Delta \mathsf{cp}_d(\gamma, \Delta) \geq 1/(56\gamma \cdot \sqrt{d})$.*

Proof. First note that

$$\tau(\gamma) = \frac{A_{d-3}}{A_{d-2}\int_0^\gamma (1-r^2)^{(d-3)/2} dr}$$

is independent of Δ and therefore it is sufficient to differentiate

$$\theta(\gamma, \Delta) = \int_0^\gamma (1-r^2)^{(d-3)/2} \int_{\frac{r\cdot\sqrt{2-\Delta^2}}{\Delta\cdot\sqrt{1-r^2}}}^1 (1-s^2)^{(d-4)/2} ds \cdot dr.$$

Let $\gamma' = \Delta/(2\sqrt{d})$ (note that by our assumption $\gamma' \leq \gamma$). By the Leibnitz integral rule,

$$\begin{aligned}
\partial_\Delta \theta(\gamma, \Delta) &= \int_0^\gamma (1-r^2)^{(d-3)/2} \partial_\Delta \left(\int_{\frac{r\cdot\sqrt{2-\Delta^2}}{\Delta\cdot\sqrt{1-r^2}}}^1 (1-s^2)^{(d-4)/2} ds \right) \cdot dr \\
&= \int_0^\gamma (1-r^2)^{(d-3)/2} \left(1 - \frac{r^2(2-\Delta^2)}{\Delta^2(1-r^2)}\right)^{\frac{d-4}{2}} \cdot \frac{2r}{\Delta^2\sqrt{1-r^2}\sqrt{2-\Delta^2}} \cdot dr \\
&\geq \int_0^\gamma (1-r^2)^{(d-4)/2} \left(1 - \frac{2r^2}{\Delta^2(1-r^2)}\right)^{\frac{d-4}{2}} \cdot \frac{2r}{\sqrt{2}\Delta^2} \cdot dr \\
&\geq \int_0^{\gamma'} (1-r^2)^{(d-4)/2} \left(1 - \frac{2r^2}{\Delta^2(1-r^2)}\right)^{\frac{d-4}{2}} \cdot \frac{\sqrt{2}\cdot r}{\Delta^2} \cdot dr.
\end{aligned}$$

Now using the conditions $\Delta \leq \sqrt{2}$, $d \geq 4$, we obtain that $\gamma' \leq 1/(2\sqrt{2})$ and hence for all $r \in [0, \gamma']$, $1 - r^2 \geq 7/8$ and $r^2/\Delta^2 \leq \gamma'^2/\Delta^2 = 1/(4d)$. This implies that for all $r \in [0, \gamma']$,

$$1 - \frac{2r^2}{\Delta^2(1-r^2)} \geq 1 - \frac{2}{\frac{7}{8}4d} = 1 - \frac{4}{7d} \ .$$

Now,

$$\left(1 - \frac{4}{7d}\right)^{(d-4)/2} \geq 1 - \frac{4(d-4)}{14d} \geq \frac{5}{7}.$$

Substituting this into our expression for $\partial_\Delta\theta(\gamma, \Delta)$ we get

$$\begin{aligned}
\partial_\Delta\theta(\gamma, \Delta) &\geq \int_0^{\gamma'} (1-r^2)^{(d-4)/2} \frac{\sqrt{2}\cdot 5r}{7\Delta^2} \cdot dr \geq \frac{1}{\Delta^2}\int_0^{\gamma'} (1-r^2)^{(d-4)/2} \cdot r \cdot dr \\
&= \frac{1}{\Delta^2(d-2)}\left(1 - (1-\gamma'^2)^{(d-2)/2}\right) \geq \frac{1}{\Delta^2(d-2)}\left(1 - e^{-\gamma'^2(d-2)/2}\right) \\
&\geq^{(*)} \frac{1}{\Delta^2(d-2)}\left(1 - (1 - \frac{(d-2)\gamma'^2}{4})\right) = \frac{\gamma'^2}{4\Delta^2} = \frac{1}{16d},
\end{aligned}$$

where to derive $(*)$ we use the fact that $e^{-\gamma'^2(d-2)/2} \leq 1 - \gamma'^2(d-2)/4$ since $e^{-x} \leq 1 - x/2$ for every $x \in [0, 1]$ and $\gamma'^2(d-2)/2 \leq \frac{\Delta^2(d-2)}{8d} < 1$. At the same time, $\int_0^\gamma (1-r^2)^{(d-3)/2} dr \leq \gamma$ and therefore,

$$\partial_\Delta \mathsf{cp}_d(\gamma, \Delta) = \tau(\gamma) \cdot \partial_\Delta\theta(\gamma, \Delta) \geq \frac{A_{d-3}}{16d\gamma A_{d-2}} \geq \frac{1}{32\sqrt{3}\gamma\sqrt{d}} > \frac{1}{56\gamma\sqrt{d}},$$

where we used that $\frac{A_{d-3}}{A_{d-2}} \geq \sqrt{d}/(2\sqrt{3})$ (e.g. [DKM09]). □