A Complete Characterization of Statistical Query Learning with Applications to Evolvability*

Vitaly Feldman

IBM Almaden Research Center

`vitaly@post.harvard.edu`

Abstract

Statistical query (SQ) learning model of Kearns is a natural restriction of the PAC learning model in which a learning algorithm is allowed to obtain estimates of statistical properties of the examples but cannot see the examples themselves [23]. We describe a new and simple characterization of the query complexity of learning in the SQ learning model. Unlike the previously known bounds on SQ learning [8, 10, 36, 3, 32] our characterization preserves the accuracy and the efficiency of learning. The preservation of accuracy implies that that our characterization gives the first characterization of SQ learning in the agnostic learning framework of Haussler and Kearns, Schapire and Sellie [19, 25]. The preservation of efficiency is achieved using a new boosting technique and allows us to derive a new approach to the design of evolutionary algorithms in Valiant's model of evolvability [35]. We use this approach to demonstrate the existence of a large class of monotone evolutionary learning algorithms based on square loss fitness estimation. These results differ significantly from the few known evolutionary algorithms and give evidence that evolvability in Valiant's model is a more versatile phenomenon than there had been previous reason to suspect.

1 Introduction

We study the complexity of learning in Kearns' well-known *statistical query* (SQ) learning model [23]. Statistical query learning is a natural restriction of the PAC learning model in which a learning algorithm is allowed to obtain estimates of statistical properties of the examples but cannot see the examples themselves. Formally, the learning algorithm is given access to $\text{STAT}(f, D)$ – a *statistical query oracle* for the unknown target function f and distribution D over some domain X. A query to this oracle is a function of an example $\phi : X \times \{-1, 1\} \to \{-1, 1\}$. The oracle may respond to the query with any value v satisfying $|\mathbf{E}_{x \sim D}[\phi(x, f(x))] - v| \leq \tau$ where $\tau \in [0, 1]$ is the *tolerance* of the query.

Kearns demonstrated that any learning algorithm that is based on statistical queries can be automatically converted to a learning algorithm robust to random classification noise of arbitrary rate smaller than the information-theoretic barrier of 1/2. Most known learning algorithms can be converted to statistical query algorithms and hence the SQ model proved to be a powerful technique for the design of noise-tolerant learning algorithms (e.g. [23, 12, 7, 13]). In fact,

---

*Earlier version appeared in the proccedings of FOCS 2009.

since the introduction of the model virtually all[1] known noise-tolerant learning algorithms were obtained from SQ algorithms. The basic approach was also extended to deal with noise in numerous other learning scenarios and has also found applications in other areas [30, 6, 22]. This makes the study of the complexity of SQ learning crucial for the understanding of noise-tolerant learning and PAC learning in general.

Kearns has also demonstrated that there are information-theoretic impediments unique to SQ learning: parity functions require an exponential number of SQs to be learned [23]. Further, Blum *et al.* proved that the number of SQs required for weak learning (that is, one that gives a non-negligible advantage over the random guessing) of a concept class C is characterized by a relatively simple combinatorial parameter of C called the *statistical query dimension* SQ-DIM(C, D) [8]. SQ-DIM(C, D) measures the maximum number of "nearly uncorrelated" (relative to distribution D) functions in C. Bshouty and Feldman gave an alternative way to characterize weak learning by statistical query algorithms that is based on the number of functions required to weakly approximate each function in C [10]. These bounds for weak learning were strengthened and extended to other variants of statistical queries in several works [9, 36, 14]. Notable applications of these bounds are lower bounds on SQ-DIM of several concept classes by Klivans and Sherstov [28] and an upper-bound on the SQ dimension of halfspaces by Sherstov [31].

While the query complexity of weak SQ learning is fairly well-studied, few works have addressed the query complexity of strong SQ learning. It is easy to see that there exist classes of functions for which strong SQ complexity is exponentially higher than the weak SQ complexity. One such example is learning of monotone functions with respect to the uniform distribution. The complexity of weak SQ learning and hence the statistical query dimension are polynomial [24, 11]. However, strong PAC learning of monotone functions with respect to the uniform distribution requires an exponential number of examples and hence an exponential number of statistical queries [24, 5]. In addition, it is important to note that the statistical query dimension and other known notions of statistical query complexity are distribution-specific and therefore one cannot directly invoke the equivalence of weak and strong SQ learning in the distribution-independent setting [1]. The first explicit[2] characterization of strong SQ learning with respect to a fixed distribution D was only recently derived by Simon [32].

1.1 Our Results

Our main result is a complete characterization of the query complexity of SQ learning in both PAC and agnostic models. Informally, our characterization states that a concept class C is SQ learnable over a distribution D if and only if for every real-valued function ψ, there exists a small (i.e. polynomial-size) set of functions G_ψ such that for every $f \in C$, if $\mathtt{sign}(\psi)$ is not "close" to f then one of the functions in G_ψ is "noticeably" correlated with $f - \psi$. More formally, for a distribution D over X, we define the (semi-)inner product over the space of real-valued functions on X as $\langle \phi, \psi \rangle_D = \mathbf{E}_{x \sim D}[\phi(x) \cdot \psi(x)]$. Then C is SQ learnable to accuracy ϵ if and only if for every $\psi : X \to [-1, 1]$, there exists a set of functions G_ψ such that (1) for every $f \in C$, if $\mathbf{Pr}_D[\mathtt{sign}(\psi) \neq f] \geq \epsilon$ then $|\langle g, f - \psi \rangle_D| \geq \gamma$ for some $g \in G_\psi$; (2) $|G_\psi|$ is polynomial and $\gamma > 0$ is inverse-polynomial in $1/\epsilon$ and n (the size of the learning problem). It is known [10] that the number of functions required to weakly approximate every

---

[1]A notable exception is the algorithm for learning parities of Blum *et al.* [9] which is tolerant to random noise, albeit not in the same strong sense as the algorithms derived from SQs.

[2]An earlier work has also considered this question but the characterization that was obtained is in terms of query-answering protocols that are essentially specifications of non-adaptive algorithms [3].

function in a set of functions F is precisely the (weak) statistical query dimension of F (after the appropriate generalization of the notion to any set of real-valued functions). Therefore, our approximation-based characterization leads to a characterization based on the following, orthogonality-based dimension: $\text{SQ-SDIM}(C, D, \epsilon) = \sup_\psi \{\text{SQ-DIM}((C \setminus B^D(\texttt{sign}(\psi), \epsilon)) - \psi, D)\}$, where $B^D(\texttt{sign}(\psi), \epsilon)$ is the set of functions that differ from $\texttt{sign}(\psi)$ on at most ϵ fraction of X and $F - \psi = \{f - \psi \mid f \in F\}$.

An important property of both of these characterizations is that the accuracy parameter in the dimension corresponds to the accuracy parameter ϵ of learning (up to the tolerance of the SQ learning algorithm). The advantage of the approximation-based characterization is that it preserves computational efficiency of learning. Namely, the set of approximating functions for ϵ-accurate learning can be computed efficiently if and only if there exists an efficient SQ learning algorithm achieving error of at most ϵ. The orthogonality-based characterization does not preserve efficiency but is more easy to analyze when proving lower bounds. Neither of these properties are possessed by the previous characterizations of strong SQ learning [3, 32, 33].

The preservation of accuracy implies that both of our characterizations can be naturally extended to agnostic learning by replacing the concept class C with the set of all functions that are Δ-close to at least one concept in C (see Th. 4.1). Learning in this model is notoriously hard and this is readily confirmed by the SQ dimension we introduce. For example, in Theorem 4.6 we prove that the SQ dimension of agnostic learning of monotone disjunctions with respect to the uniform distribution is super polynomial. This provides new evidence that agnostic learning of conjunctions is a hard problem even when restricted to the monotone case over the uniform distribution. The preservation of accuracy is critical for the generalization to agnostic learning since, unlike in the PAC model, achieving, for example, twice the error (i.e. $2 \cdot \Delta$) might be a substantially easier task than learning to accuracy $\Delta + \epsilon$.

We note that the characterization of (strong) SQ learning by Simon [32] has some similarity to ours. It also examines weak statistical query dimension of $F - \psi$ for $F \subseteq C$ and some function ψ. However, the maximization is over all sets of functions F satisfying several properties and ϕ is fixed to be the average of functions in F. Simon's SQ dimension and the characterization were substantially simplified in a very recent and independent work of Szörényi [33]. As it was shown by Szörényi, his dimension can be easily related to the dimension we use in our second characterization (Th. 3.11). Szörényi's result is based on a very different technique and does not have preserve efficiency and accuracy.

1.2 Overview of the Proof

To prove the first direction of our characterization we simulate the SQ learning algorithm for C while replying to its statistical queries using ψ in place of the unknown target function f. If ψ is not close to f then one of the queries in this execution has to distinguish between f and ψ, giving a function that weakly approximates $f - \psi$. Hence the polynomial number of queries in this execution implies the existence of the set G_ψ with the desired property.

For the second direction we use the fact that $\langle g, f - \psi \rangle_D \geq \gamma$ means that g "points" in the direction of f from ψ, that is, $\psi + \gamma \cdot g$ is closer to f than ψ by at least γ^2 in the norm corresponding to our inner product. Therefore one can "learn" the target function f by taking steps in the direction of f until the hypothesis converges to f. This argument requires the hypothesis at each step to have range in $[-1, 1]$ and therefore we apply a projection step after each update. This process is closely related to projected gradient descent – a well-known technique in a number of areas. The closest analogues of this technique in learning are some boosting algorithms (e.g. [4]). In particular, our algorithm is closely related to the hard-core set construction of Impagliazzo

3

[20] adapted to boosting by Klivans and Servedio [27]. The proof of our result can also be seen as a new type of boosting algorithm that instead of using a weak learning algorithm on different distributions uses a weak learning algorithm on different target functions (namely $f - \psi$). This connection is explored in [17].

1.3   Applications to Evolvability

The characterization and its efficiency-preserving proofs imply that if C is SQ learnable then for every hypothesis function ψ, there exists a small and *efficiently computable* set of functions $N(\psi)$ such that if ψ is not "close" to $f \in C$ then one of the functions in $N(\psi)$ is "closer" to f than ψ (Th. 5.4). This property implies that every SQ learnable C is learnable by a canonical learning algorithm which learns C via a sequential process in which at every step the best hypothesis is chosen from a small and fixed pool of hypotheses "adjacent" to the current hypothesis.

This type of learning has been recently proposed by Valiant as one that can explain the acquisition of complex functionality by living organisms through the process of evolution guided by natural selection [35]. One particular important issue addressed by the model is the ability of an evolutionary algorithm to adjust to a change of the target function without sacrificing the fitness of the current hypothesis (beyond the decrease caused by the change itself). Existence of algorithms that are robust to such changes (we refer to them as *monotone*) could explain the ability of some organisms to adapt to changes in environmental conditions without the need for a "restart". While the power of non-monotone evolvability was resolved in our recent work [14, 16], very few examples of monotone evolutionary algorithms are known. Michael's algorithm for evolving decision lists with respect to the uniform distribution [29] and the distribution-independent algorithm for evolving singletons (functions that are positive on a single point) in [16] are the only examples we are aware of. Our canonical learning algorithms can be fairly easily translated into evolutionary algorithms demonstrating that every concept class C SQ learnable with respect to a distribution D, is evolvable monotonically over D (Th. 5.5).

While we do not know how to extend this general method to the more robust distribution-independent evolvability, we show that the underlying ideas can be useful for this purpose as well. Namely, we prove distribution-independent and monotone evolvability of Boolean disjunctions (or conjunctions) using a simple and natural mutation algorithm (Th. 5.7). The mutation algorithm is based on slight adjustments of the contribution of each of the Boolean variables while bounding the total value of contributions (which corresponds to the projection step). Both of these results are based on measuring fitness of a hypothesis using the quadratic loss function. Formal definitions of the model and the results are given in Section 5.

1.4   Relation to the Earlier Version

Since the appearance of the earlier version of this work [15] we have found ways to strengthen some of the parameters of the characterizations. As a result the dimensions used here differ from the ones introduced in [15]. Also, unlike the dimension we use here, the SQD_ϵ dimension in [15] preserves the output hypothesis space and hence is suitable for characterizing *proper* learning. To emphasize the difference we use different notation for the dimensions defined in the two versions of the work. In addition, the characterization of learning in the agnostic model is now simplified using recent distribution-specific agnostic boosting algorithms [17, 21].

2 Preliminaries

For a positive integer ℓ, let $[\ell]$ denote the set $\{1, 2, \ldots, \ell\}$. We denote the domain of our learning problems by X and let $\mathcal{F}_1^\infty$ denote the set of all functions from X to $[-1, 1]$ (that is all the functions with L_∞ norm bounded by 1). It will be convenient to view a distribution D over X as defining the product $\langle \phi, \psi \rangle_D = \mathbf{E}_{x \sim D}[\phi(x) \cdot \psi(x)]$ over the space of real-valued functions on X. It is easy to see that this is simply a non-negatively weighted version of the standard dot product over $\mathbb{R}^X$ and hence is a positive semi-inner product over $\mathbb{R}^X$. The corresponding norm is defined as $\|\phi\|_D = \sqrt{\mathbf{E}_D[\phi^2(x)]} = \sqrt{\langle \phi, \phi \rangle_D}$. We define an ϵ-ball around a Boolean function h as $B^D(h, \epsilon) = \{g : X \to \{-1, 1\} \mid \mathbf{Pr}_D[f \neq g] \leq \epsilon\}$. For two real-valued functions ϕ and ψ we let $L_1^D(\phi, \psi) = \mathbf{E}_D[|\phi(x) - \psi(x)|]$. For a set of real-valued functions F and a real-valued function ψ we denote by $F - \psi = \{f - \psi \mid f \in F\}$.

2.1 PAC Learning

For a *domain* X, a *concept class* over X is a set of $\{-1, 1\}$-valued functions over X referred to as *concepts*. A concept class together with a specific way to represent all the functions in the concept class is referred to as a *representation class*. For brevity, we often refer to a representation class as just a concept class with some implicit representation scheme.

There is often a complexity parameter n associated with the domain X and the concept class C such as the number of Boolean variables describing an element in X or the number of real dimensions. In such a case it is understood that $X = \bigcup_{n \geq 1} X_n$ and $C = \bigcup_{n \geq 1} C_n$. We drop the subscript n when it is clear from the context. In some cases it useful to consider another complexity parameter associated with C: the minimum description length of f under the representation scheme of C. Here, for brevity, we assume that n (or a fixed polynomial in n) bounds the description length of all functions in C_n.

The models we consider are based on the well-known PAC learning model introduced by Valiant [34]. Let C be a representation class over X. In the basic PAC model a learning algorithm is given examples of an unknown function f from C on points randomly chosen from some unknown distribution D over X and should produce a hypothesis h that approximates f. Formally, an *example oracle* $\mathrm{EX}(f, D)$ is an oracle that upon being invoked returns an example $\langle x, f(x) \rangle$, where x is chosen randomly with respect to D, independently of any previous examples.

An algorithm is said to PAC learn C in time t if for every $\epsilon > 0$, $\delta > 0$, $f \in C$, and distribution D over X, the algorithm given ϵ and access to $\mathrm{EX}(f, D)$ outputs, in time t and with probability at least $2/3$, a hypothesis h that is evaluatable in time t and satisfies $\mathbf{Pr}_D[f(x) \neq h(x)] \leq \epsilon$. For convenience we also allow real-valued hypotheses in $\mathcal{F}_1^\infty$. Such a hypothesis needs to satisfy $\langle f(x), h(x) \rangle_D \geq 1 - 2\epsilon$. A real-valued hypothesis $\phi(x)$ can be also thought of as a randomized Boolean hypothesis $\Phi(x)$, such that $\phi(x)$ equals the expected value of $\Phi(x)$. Hence $\langle f(x), \phi(x) \rangle_D \geq 1 - 2\epsilon$ is equivalent to saying that the expected error of $\Phi(x)$ is at most ϵ. We say that an algorithm *efficiently* learns C when t is upper bounded by a polynomial in n, $1/\epsilon$.

The basic PAC model is also referred to as *distribution-independent* learning to distinguish it from *distribution-specific* PAC learning in which the learning algorithm is required to learn only with respect to a single distribution D known in advance.

A *weak* learning algorithm [26] is a learning algorithm that produces a hypothesis whose disagreement with the target concept is noticeably less than $1/2$ (and not necessarily any $\epsilon > 0$). More precisely, a weak learning algorithm produces a hypothesis $h \in \mathcal{F}_1^\infty$ such that $\langle f(x), h(x) \rangle_D \geq 1/p(n)$ for some fixed polynomial p.

2.2   Agnostic Learning

The *agnostic* learning model was introduced by Haussler [19] and Kearns *et al.* [25] in order to model situations in which the assumption that examples are labeled by some $f \in C$ does not hold. In the most general version of the model the examples are generated from some unknown distribution A over $X \times \{-1, 1\}$. The goal of an agnostic learning algorithm for a concept class C is to produce a hypothesis whose error on examples generated from A is close to the best possible by a concept from C. Any distribution A over $X \times \{-1, 1\}$ can be described uniquely by its marginal distribution D over X and the expectation of the label b given x. That is, we refer to a distribution A over $X \times \{-1, 1\}$ by a pair (D_A, ϕ_A) where $D_A(z) = \mathbf{Pr}_{\langle x,b \rangle \sim A}[x = z]$ and

$$\phi_A(z) = \mathbf{E}_{\langle x,b \rangle \sim A}[b \mid z = x].$$

Formally, for a function $h \in \mathcal{F}_1^\infty$ and a distribution $A = (D, \phi)$ over $X \times \{-1, 1\}$, we define

$$\Delta(A, h) = L_1^D(\phi, h)/2 \ .$$

Note that for a Boolean function h, $\Delta(A, h)$ is exactly the error of h in predicting an example drawn randomly from A or $\mathbf{Pr}_{\langle x,b \rangle \sim A}[h(x) \neq b]$. For a concept class C, let $\Delta(A, C) = \inf_{h \in C}\{\Delta(A, h)\}$.

Kearns *et al.* [25] define agnostic learning as follows.

Definition 2.1 *An algorithm $\mathcal{A}$ agnostically learns a representation class C if for every $\epsilon > 0, \delta > 0$, distribution A over $X \times \{-1, 1\}$, $\mathcal{A}$ given access to examples drawn randomly from A, outputs, with probability at least $2/3$, a hypothesis $h \in \mathcal{F}_1^\infty$ such that $\Delta(A, h) \leq \Delta(A, C) + \epsilon$.*

As in the PAC learning, the learning algorithm is *efficient* if it runs in time polynomial $1/\epsilon$ and n.

More generally, for $0 < \alpha \leq \beta \leq 1/2$ an (α, β)-agnostic learning algorithm is the algorithm that produces a hypothesis h such that $\Delta(A, h) \leq \beta$ whenever $\Delta(A, C) \leq \alpha$. In the distribution-specific version of this model, learning is only required for every $A = (D, \phi)$, where D equals to some fixed distribution known in advance.

2.3   The Statistical Query Learning Model

In the *statistical query model* of Kearns [23] the learning algorithm is given access to $\mathrm{STAT}(f, D)$ – a *statistical query oracle* for target concept f with respect to distribution D instead of $\mathrm{EX}(f, D)$. A query to this oracle is a function $\psi : X \times \{-1, 1\} \rightarrow \{-1, 1\}$. The oracle may respond to the query with any value v satisfying $|\mathbf{E}_D[\psi(x, f(x))] - v| \leq \tau$ where $\tau \in [0, 1]$ is a real number called the *tolerance* of the query. For convenience, we allow the query functions to be real-valued in the range $[-1, 1]$. As it has been observed by Aslam and Decatur [2], this extension is equivalent to the original SQ model.

An algorithm $\mathcal{A}$ is said to learn C in time t from statistical queries of tolerance τ if $\mathcal{A}$ PAC learns C using $\mathrm{STAT}(f, D)$ in place of the example oracle. In addition, each query ψ made by $\mathcal{A}$ has tolerance τ and can be evaluated in time t. The *statistical query learning complexity* of C over D is the minimum number of queries of tolerance τ required to learn C over D to accuracy ϵ and is denoted by $\mathrm{SLC}(C, D, \epsilon, \tau)$.

The algorithm is said to (efficiently) SQ learn C if t is polynomial in n and $1/\epsilon$, and τ is lower-bounded by the inverse of a polynomial in n and $1/\epsilon$.

The SQ learning model extends to the agnostic setting analogously. That is, random examples from A are replaced by queries to the SQ oracle $\mathrm{STAT}(A)$. For a query ψ as above, $\mathrm{STAT}(A)$

returns a value v satisfying $|\mathbf{E}_{\langle x,b \rangle \sim A}[\psi(x,b)] - v| \leq \tau$. We denote the *agnostic statistical query learning complexity* of C over D by $\text{ASLC}(C, D, \epsilon, \tau)$.

A *correlational* statistical query is a statistical query for a correlation of a function over X with the target [10]. Namely the query function $\psi(x, \ell) \equiv \phi(x) \cdot \ell$ for a function $\phi \in \mathcal{F}_1^\infty$. We say that a query is *target-independent* if $\psi(x, \ell) \equiv \phi(x)$ for a function $\phi \in \mathcal{F}_1^\infty$, that is, if ψ is a function of the point x alone. We will need the following simple fact by Bshouty and Feldman [10] to relate learning by statistical queries to learning by CSQs.

Lemma 2.2 ([10]) *For any function $\psi : X \times \{-1, 1\} \to [-1, 1]$, $\psi(x, \ell) \equiv \phi_1(x) \cdot \ell + \phi_2(x)$, for some $\phi_1, \phi_2 \in \mathcal{F}_1^\infty$. In particular a statistical query (ψ, τ) with respect to any distribution D can be answered using a statistical query that is target-independent and a correlational statistical query, each of tolerance $\tau/2$.*

2.4   (Weak) SQ Dimension

Blum *et al.* showed that concept classes weakly SQ learnable using only a polynomial number of statistical queries of inverse polynomial tolerance are exactly the concept classes that have polynomial *statistical query dimension* or SQ-DIM [8]. The dimension is based on the largest number of almost orthogonal (using the $\langle \cdot, \cdot \rangle_D$ inner product) functions in the set.

Definition 2.3 ([8, 36]) *For a concept class C we say that $SQ\text{-}DIM(C, D) = d$ if d is the largest value for which there exist d functions $f_1, f_2, \ldots, f_d \in C$ such that for every $i \neq j$, $|\langle f_i, f_j \rangle_D| \leq 1/d$.*

Bshouty and Feldman gave an alternative way to characterize weak learning by statistical query algorithms that is based on the number of functions required to weakly approximate each function in the set [10].

Definition 2.4 *For a concept class C and $\gamma > 0$ we say that $SQD(C, D, \gamma) = d$ if there exists a set of d functions $G \subset \mathcal{F}_1^\infty$ such that for every $f \in C$, $|\langle f, g \rangle_D| \geq \gamma$ for some $g \in G$. In addition, no value smaller than d has this property.*

Bshouty and Feldman show that a concept class C is weakly SQ learnable over D using a polynomial number of queries if and only if $\text{SQD}(C, D, 1/t(n)) = d(n)$ for some polynomials $d(\cdot)$ and $t(\cdot)$ [10]. It is also possible to relate SQD and SQ-DIM more directly. It is well-known that the maximal set of almost orthogonal functions in C is also the approximating set for C. In other words, $\text{SQD}(C, D, 1/d) \leq d$, where $d = \text{SQ-DIM}(C, D)$. The connection in the other direction is implicit in the work of Blum *et al.* [8]. Here we will use a stronger version given by Yang [36].

Lemma 2.5 ([36]) *Let C be a concept class and D be a distribution over X. Then $SQD(C, D, d^{-1/3}) \geq d^{1/3}/2$, where $d = SQ\text{-}DIM(C, D)$.*

3   Strong SQ Dimension

In this section we give a generalization of the weak statistical query dimension to strong learning. We first extend the approximation-based characterization of Bshouty and Feldman [10] and then obtain an orthogonality-based characterization from it.

3.1 Approximation-Based Characterization

In order to define our strong statistical query dimension we first need to generalize the approximation-based characterization of Bshouty and Feldman [10] to sets of real-valued functions rather than just concept classes. To achieve this we simply note that the definition of $SQD(C, D, \gamma)$ does not use the fact that functions in C are Boolean and hence we can define $SQD(F, D, \gamma)$ for any set of real-valued functions F in exactly the same way. We now define the strong statistical query dimension of a class of functions C.

Definition 3.1 *For a concept class C, distribution D and $\epsilon, \gamma > 0$ we define*

$$SQSD(C, D, \epsilon, \gamma) = \sup_{\psi \in \mathcal{F}_1^\infty} \left\{ SQD(C \setminus B^D(\mathtt{sign}(\psi), \epsilon) - \psi, D, \gamma) \right\},$$

In other words, we say that $SQSD(C, D, \epsilon, \gamma) = d$ if for every $\psi \in \mathcal{F}_1^\infty$, there exists a set of d functions $G_\psi \subset \mathcal{F}_1^\infty$ such that for every $f \in C$, either

1. *$\mathbf{Pr}_D[f(x) \neq \mathtt{sign}(\psi(x))] \leq \epsilon$ or*

2. *there exists $g \in G_\psi$ such that $|\langle f - \psi, g \rangle_D| \geq \gamma$.*

In addition, no value smaller than d has this property.

We now give a simple proof that $SQSD(C, D, \epsilon, \gamma)$ characterizes (within a polynomial) the number of statistical queries required to learn C over D with accuracy ϵ and query tolerance γ.

Theorem 3.2 *For every concept class C, distribution D over X and $\epsilon, \tau > 0$, $SLC(C, D, \epsilon, \tau) \geq SQSD(C, D, \epsilon + \tau, \tau) - 2$.*

Proof: Let $\mathcal{A}$ be a SQ algorithm that learns C over D using $q = SLC(C, D, \epsilon, \tau)$ queries of tolerance τ. According to Lemma 2.2 we can assume that $\mathcal{A}$ makes only correlational SQs of tolerance τ since $\mathcal{A}$ can compute the values of the target-independent SQs exactly.

Now let $\psi \in \mathcal{F}_1^\infty$ be any function. The set G_ψ is constructed as follows. Simulate algorithm $\mathcal{A}$ and for every correlational query $(\phi_i \cdot \ell, \tau)$ add ϕ_i to G_ψ and respond to the query with the value $\langle \psi, \phi_i \rangle_D = \mathbf{E}_D[\phi_i(x) \cdot \psi(x)]$. Continue the simulation until $\mathcal{A}$ outputs a hypothesis h_ψ. Add $\mathtt{sign}(\psi)$ and h_ψ to G_ψ.

First, by the definition of G_ψ, $q \geq |G_\psi| - 2$. Now, let f be any function in C. If there does not exist $g \in G_\psi$ such that $|\langle f - \psi, g \rangle_D| \geq \tau$ then for every correlational query function $\phi_i \in G_\psi$, $|\langle \psi, \phi_i \rangle_D - \langle f, \phi_i \rangle_D| < \tau$. This means that in our simulation, $\langle \psi, \phi_i \rangle_D$ is within τ of $\langle f, \phi_i \rangle_D$. Therefore the answers provided by our simulator are valid for the execution of $\mathcal{A}$ when the target function is f. That is they could have been returned by $\mathrm{STAT}(f, D)$ with tolerance τ. Therefore, by the definition of $\mathcal{A}$, the hypothesis h_ψ satisfies $\langle f, h_\psi \rangle_D \geq 1 - 2\epsilon$. Both $\mathtt{sign}(\psi)$ and h_ψ are in G_ψ and therefore we also know that $|\langle f - \psi, \mathtt{sign}(\psi) \rangle_D| \leq \tau$ and $|\langle f - \psi, h_\psi \rangle_D| \leq \tau$. These conditions imply that $\langle f, \mathtt{sign}(\psi) \rangle_D \geq \langle \psi, \mathtt{sign}(\psi) \rangle_D - \tau$ and $\langle \psi, h_\psi \rangle_D \geq \langle f, h_\psi \rangle_D - \tau$. In addition for all $\psi, h_\psi \in \mathcal{F}_1^\infty$, $\langle \psi, \mathtt{sign}(\psi) \rangle_D \geq \langle \psi, h_\psi \rangle_D$. By combining these inequalities we conclude that

$$\langle f, \mathtt{sign}(\psi) \rangle_D \geq \langle \psi, \mathtt{sign}(\psi) \rangle_D - \tau \geq \langle \psi, h_\psi \rangle_D - \tau \geq \langle f, h_\psi \rangle_D - 2\tau \geq 1 - 2\epsilon - 2\tau ,$$

which is equivalent to $\mathbf{Pr}_D[f(x) \neq \mathtt{sign}(\psi(x))] \leq \epsilon + \tau$. In other words, if there does not exist $g \in G_\psi$ such that $|\langle f - \psi, g \rangle_D| \geq \tau$ then $f \in B^D(\mathtt{sign}(\psi), \epsilon + \tau)$, giving us the claimed inequality. $\qquad \square$

Remark 3.3 *If $\mathcal{A}$ is randomized then it can be converted to a non-uniform deterministic algorithm via a standard transformation (e.g. [10]). Therefore, Theorem 3.2 also applies to SQ learning by randomized algorithms.*

We now establish the other direction of our characterization.

Theorem 3.4 *For every concept class C, distribution D over X and $\epsilon, \tau > 0$, $SLC(C, D, \epsilon, \tau) \leq SQSD(C, D, \epsilon, 4 \cdot \tau)/(3\tau^2)$.*

Proof: Let $d = \text{SQSD}(C, D, \epsilon, 4 \cdot \tau)$. Our learning algorithm for C builds an approximation to the target function f in steps. In each step we have a current hypothesis $\psi_i \in \mathcal{F}_1^\infty$. If $\text{sign}(\psi_i)$ is not ϵ-close to f then we find a function $g \in G_{\psi_i}$ such that $|\langle f - \psi_i, g \rangle_D| \geq \gamma$. Such g can be viewed as a vector "pointing" in the direction of f from ψ_i. We therefore set $\psi'_{i+1} = \psi_i + \langle f - \psi_i, g \rangle_D \cdot g$. As we will show ψ'_{i+1} is closer (in distance measured by $\| \cdot \|_D$) to f than ψ_i. However ψ'_{i+1} is not necessarily in $\mathcal{F}_1^\infty$. We define ψ_{i+1} to be the projection of ψ'_{i+1} onto $\mathcal{F}_1^\infty$. As we will show this projection step only decreases the distance to the target function. We will now provide the details of the proof.

Let $\psi_0 \equiv 0$. Given ψ_i we define ψ_{i+1} as follows. Let G_{ψ_i} be the set of size at most d that correlates with every function in $C \setminus B^D(\text{sign}(\psi_i), \epsilon) - \psi_i$ (as given by Definition 3.1). For every $g \in G_{\psi_i}$ we make a query for $\langle f, g \rangle_D$ to $\text{STAT}(f, D)$ with tolerance τ and denote the answer by $v(g)$. If there exists $g \in G_{\psi_i}$ such that $|v(g) - \langle \psi_i, g \rangle_D| \geq 3\tau$ then we set $g_i = g$, $\gamma_i = v(g_i) - \langle \psi_i, g_i \rangle_D$, and $\psi'_{i+1} = \psi_i + \gamma_i \cdot g_i$. Otherwise the algorithm outputs $\text{sign}(\psi_i)$. Note that if $\text{sign}(\psi_i)$ is not ϵ-close to f then there exists $g \in G_{\psi_i}$ such that $|\langle f - \psi_i, g \rangle_D| \geq 4\tau$ and, in particular, $|v(g) - \langle \psi_i, g \rangle_D| \geq 3\tau$.

We then set ψ_{i+1} to be the projection of ψ'_{i+1} onto $\mathcal{F}_1^\infty$:

$$\psi_{i+1}(x) = P_1(\psi'_{i+1}) \triangleq \begin{cases} \psi'_{i+1}(x) & |\psi'_{i+1}(x)| \leq 1 \\ \text{sign}(\psi'_{i+1}(x)) & \text{otherwise.} \end{cases}$$

We then continue to the next iteration using ψ_{i+1}.

As we can see $\text{sign}(\psi_i)$ is only output when $\text{sign}(\psi_i)$ is ϵ-close to f. Therefore in order to prove the desired bound on the number of queries it is sufficient to show that the algorithm will output $\text{sign}(\psi_i)$ after an appropriate number of iterations. This is established via the following claim.

Claim 3.5 *For every i, $\|f - \psi_i\|_D^2 \leq 1 - 3 \cdot i \cdot \tau^2$.*

Proof: First, $\|f - \psi_0\|_D^2 = \|f\|_D^2 = 1$. Next,

$$\|f - \psi'_{i+1}\|_D^2 = \|(f - \psi_i) - \gamma_i \cdot g_i\|_D^2 = \|f - \psi_i\|_D^2 + \|\gamma_i \cdot g_i\|_D^2 - 2\langle f - \psi_i, \gamma_i \cdot g_i \rangle_D.$$

Therefore,

$$\|f - \psi_i\|_D^2 - \|f - \psi'_{i+1}\|_D^2 = 2\gamma_i \langle f - \psi_i, g_i \rangle_D - \gamma_i^2 \|g_i\|_D^2 \geq 2 \cdot \gamma_i \cdot \langle f - \psi_i, g_i \rangle_D - \gamma_i^2$$
$$=^{(*)} 2 \cdot |\gamma_i| \cdot |\langle f - \psi_i, g_i \rangle_D| - \gamma_i^2 \geq 2 \cdot |\gamma_i|(|\gamma_i| - \tau) - \gamma_i^2 \geq \gamma_i^2/3 \geq 3 \cdot \tau^2.$$

To obtain $(*)$ we note that $|\gamma_i| \geq 3\tau$ and $|\langle f - \psi_i, g_i \rangle_D - \gamma_i| = |\langle f, g_i \rangle_D - v(g_i)| \leq \tau$. Therefore the sign of γ_i is the same as the sign of $\langle f - \psi_i, g_i \rangle_D$ and $|\langle f - \psi_i, g_i \rangle_D| \geq |\gamma_i| - \tau \geq 2\gamma_i/3$.

We now claim that $\|f - \psi'_{i+1}\|_D^2 \geq \|f - \psi_{i+1}\|_D^2$. This follows easily from the definition of ψ_{i+1}. If for a point x, $\psi_{i+1}(x) = \psi'_{i+1}(x)$ then clearly $f(x) - \psi'_{i+1}(x) = f(x) - \psi_{i+1}(x)$.

Otherwise, if $|\psi'_{i+1}(x)| > 1$ then $\psi_{i+1}(x) = \mathtt{sign}(\psi'_{i+1}(x))$ and for any value $f(x) \in \{-1, 1\}$, $|f(x) - \psi'_{i+1}(x)| \geq |f(x) - \psi_{i+1}(x)|$. This implies that $\mathbf{E}_D[(f - \psi'_{i+1})^2] \geq \mathbf{E}_D[(f - \psi_{i+1})^2]$.

We therefore obtain that for every i, $\|f - \psi_i\|_D^2 - \|f - \psi_{i+1}\|_D^2 \geq 3\tau^2$ giving us the claim. $\square$ (Cl. 3.5)

Claim 3.5 implies that the algorithm makes at most $1/(3\tau^2)$ iterations. In each iteration at most d queries are made and therefore the algorithm uses at most $d/(3\tau^2)$ queries of tolerance τ. $\square$ (Th. 3.4)

An important property of the proofs of Theorems 3.2 and 3.4 that they give a simple and *efficient* way to convert a learning algorithm for C into an algorithm that given access to target-independent statistical queries with respect to D builds an approximating set G_ψ for every ψ and vice versa. The access to target-independent statistical queries with respect to D can be replaced by a circuit that provides random samples from D if D is efficiently samplable or a fixed polynomial-size random (unlabeled) sample from D (in this case the resulting algorithm is non-uniform). For convenience we refer to either of these options as access to D.

Theorem 3.6 *Let C be a concept class and D be a distribution over X. C is efficiently SQ learnable if and only if there exists an algorithm $\mathcal{B}$ that for every $\epsilon > 0$, given ϵ, access to D and a circuit for $\psi \in \mathcal{F}_1^\infty$ can produce a set of functions G_ψ such that*

1. *G_ψ satisfies the conditions of Definition 3.1 for some polynomial d and inverse-polynomial γ (in n, $1/\epsilon$);*

2. *circuit size of every function in G_ψ is polynomial in n and $1/\epsilon$;*

3. *the running time of $\mathcal{B}$ is polynomial in n, $1/\epsilon$ and the circuit size of ψ.*

Proof: The proof of Theorem 3.2 gives a way to construct the set G_ψ by simulating $\mathcal{A}$ while using ψ in place of the target function f. This construction of G_ψ would be efficient provided the exact values of $\mathbf{E}_D[\phi_i(x) \cdot \psi(x)]$ and the exact values of target-independent SQs in the simulation of algorithm $\mathcal{A}$ were available. However it is easy to see that the exact values are not necessary and can be replaced by estimates within $\tau/2$. Such estimates can be easily obtained given access to D.

Similarly, in the proof of Theorem 3.4 the iterative procedure would yield an efficient SQ learning algorithm for C provided the exact values of $\langle \psi_i, g \rangle_D$ were available. It is again easy to see that in place of exact values estimates within $\tau/2$ can be used if the accuracy of statistical queries is also increased to $\tau/2$. This implies that if there exists an efficient algorithm that given a polynomial size circuit for $\psi \in \mathcal{F}_1^\infty$ and access to D generates G_ψ then C is efficiently SQ learnable over D. $\square$

3.2 Orthogonality-Based Characterization

In order to simplify the application of our characterization we show that, with only a polynomial loss in the bounds one can obtain an orthogonality-based version of SQSD. Specifically, we convert the bound on the number of functions required to weakly approximate every function in some set of functions F to a bound on the maximum number of almost uncorrelated functions in F.

In Lemma 3.8 we generalize Yang's conversion (Lemma 2.5) to sets of arbitrary real-valued functions. But first we need to appropriately extend the definition of SQ-DIM to sets of arbitrary real-valued functions. For this purpose we simply use Definition 2.3 applied to sets of real-valued functions.

10

Definition 3.7 *For a set of real-valued functions F we say that $SQ\text{-}DIM(F, D) = d$ if d is the largest value for which there exist d functions $f_1, f_2, \ldots, f_d \in F$ such that for every $i \neq j$, $|\langle f_i, f_j \rangle_D| \leq 1/d$.*

Lemma 3.8 *Let D be a distribution and F be set of functions such that every $\phi \in F$, $m \leq \|\phi\|_D \leq M$ for some $M \geq 1 \geq m$. Then $SQD(F, D, M(dm^2)^{-1/3}) \geq (dm^2)^{1/3}/2$, where $d = SQ\text{-}DIM(F, D)$.*

Proof: Yang shows that our claim is correct if for every $\phi \in F$, $\|\phi\|_D = 1$ [36, Cor. 1]. While his claim (Lemma 2.5) is only for Boolean functions the only property of Boolean functions used in his proof is their $\|\cdot\|_D$-norm being equal to 1. We reduce our general case to this special case by defining $F' = \{f/\|f\|_D \mid f \in F\}$. We claim that $SQ\text{-}DIM(F', D) \geq d \cdot m^2$. It is easy to see this since if for $f_1, f_2 \in F$, $\langle f_1, f_2 \rangle_D \leq 1/d$ then

$$\left\langle \frac{f_1}{\|f_1\|_D}, \frac{f_2}{\|f_2\|_D} \right\rangle_D \leq \frac{1}{dm^2} \; .$$

This means that the existence of a set of d functions in F with correlations of at most $1/d$ would imply the existence of $d \geq d \cdot m^2$ functions in F' with mutual correlations of at most $1/(dm^2)$.

We apply Yang's result to F' and obtain $SQD(F', D, (dm^2)^{-1/3}) \geq (dm^2)^{1/3}/2$. This implies that $SQD(F, D, M(dm^2)^{-1/3}) \geq (dm^2)^{1/3}/2$. To see this assume for the sake of contradiction that there exists a set G of size less than $(dm^2)^{1/3}/2$ such that for every $f \in F$, $|\langle f, g \rangle_D| \geq M(dm^2)^{-1/3}$ for some $g \in G$. Then for every $f' = f/\|f\|_D \in F'$, $|\langle f', g \rangle_D| = |\langle f, g \rangle_D|/\|f\|_D \geq M(dm^2)^{-1/3}/\|f\|_D \geq (dm^2)^{-1/3}$. This would violate the bound on $SQD(F', D, (dm^2)^{-1/3})$ that we have obtained. $\qquad\square$

We define $SQ\text{-}SDIM(C, D, \epsilon)$ to be the generalization of $SQ\text{-}DIM$ to ϵ-accurate learning as follows.

Definition 3.9 $SQ\text{-}SDIM(C, D, \epsilon) = \sup_{\psi \in \mathcal{F}_1^\infty} SQ\text{-}DIM(C \setminus B^D(\mathtt{sign}(\psi), \epsilon) - \psi, D).$

We now ready to relate SQSD and SQ-SDIM.

Theorem 3.10 *Let C be a concept class D be a distribution over X, $\epsilon > 0$ and $d = SQ\text{-}SDIM(C, D, \epsilon)$. Then $SQSD(C, D, \epsilon, 1/(2d)) \leq d$ and $SQSD(C, D, \epsilon, 2(\epsilon d)^{-1/3}) \geq (\epsilon d)^{1/3}/2$.*

Proof: Let $\psi \in \mathcal{F}_1^\infty$ be any function, let $F_\psi = C \setminus B^D(\mathtt{sign}(\psi), \epsilon) - \psi$ and let $d' = SQ\text{-}DIM(F_\psi, D) \leq SQ\text{-}SDIM(C, D, \epsilon) = d$.

For the first part of the claim we use a minor modification of the standard relation between SQD and SQ-SDIM (see Section 2.4). Let $F_1 = \{f_1, f_2, \ldots, f_{d'}\} \subseteq F_\psi$ be the largest set of functions such that for every $i \neq j$, $|\langle f_i, f_j \rangle_D| \leq 1/d'$. The maximality of d' implies that for every $f \in F_\psi$, there exists $f_i \in F_1$ such that $|\langle f_i, f \rangle_D| > 1/d'$. Thus F_1 is an approximating set for F_ψ. The only minor problem is that we need an approximating set of functions in $\mathcal{F}_1^\infty$. The domain of each function in F_ψ is $[-2, 2]$ and therefore to obtain an approximating set in $\mathcal{F}_1^\infty$ we simply scale F_1 by $1/2$. By taking $G_\psi = \{f/2 \mid f \in F_1\}$ we obtain that $SQD(F_\psi, D, 1/(2d')) \leq d'$. This holds for every $\psi \in \mathcal{F}_1^\infty$ and therefore $SQSD(C, D, \epsilon, 1/(2d)) \leq d$.

For the second part of the claim we first observe that that for every $f \in F_\psi$, $f = c - \psi$ for $c \in C$ and $\mathbf{Pr}_D[c \neq \mathtt{sign}(\psi)] > \epsilon$. This implies that $\mathbf{Pr}_D[|f| \geq 1] \geq \epsilon$ and hence $2 \geq \|f\|_D \geq \sqrt{\epsilon}$. We now use Lemma 3.8 to obtain $SQD(F_\psi, D, 2(\epsilon d')^{-1/3}) \geq (\epsilon d')^{1/3}/2$. This implies that $SQSD(C, D, \epsilon, 2(\epsilon d)^{-1/3}) \geq (\epsilon d)^{1/3}/2$. $\qquad\square$

We can combine Theorem 3.10 with the approximation-based characterization (Th. 3.2 and 3.4) to obtain a characterization of strong SQ learnability based on SQ-SDIM.

Theorem 3.11 *Let C be a concept class, D be a distribution over X and $\epsilon > 0$. If there exists a polynomial $p(\cdot, \cdot)$ such that C is SQ learnable to accuracy ϵ from $p(n, 1/\epsilon)$ queries of tolerance $1/p(n, 1/\epsilon)$ then $SQ\text{-}SDIM(C, D, \epsilon + 1/p(n, 1/\epsilon)) \leq p'(n, 1/\epsilon)$ for some polynomial $p'(\cdot, \cdot)$. Further, if $SQ\text{-}SDIM(C, D, \epsilon) \leq p(n, 1/\epsilon)$ then C is SQ learnable to accuracy ϵ from $p'(n, 1/\epsilon)$ queries of tolerance $1/p'(n, 1/\epsilon)$ for some polynomial $p'(\cdot, \cdot)$.*

4 SQ Dimension for Agnostic Learning

In this section we extend the statistical query dimension characterization to agnostic learning. Our characterization is based on the well-known observation that agnostic learning of a concept class C requires (a weak form of) learning of the set of functions F in which every function is weakly approximated by some function in C [25]. For example agnostic learning of Boolean conjunctions implies weak learning of DNF expressions. We formalize this by defining an L_1^D ϵ-ball around a real-valued function ϕ over X as $B_1^D(\phi, \epsilon) = \{\psi \in \mathcal{F}_1^\infty \mid L_1^D(\psi, \phi) \leq \epsilon\}$ and around a set of functions C as $L_1^D(C, \epsilon) = \cup_{f \in C} B_1^D(\phi, \epsilon)$. In (α, β)-agnostic learning of a function class C over the marginal distribution D, the learning algorithm only needs to learn when the distribution over examples $A = (D, \phi)$ satisfies $\Delta(A, C) \leq \alpha$. In other words, for any $A = (D, \phi)$ such that there exists $c \in C$, for which $\Delta(A, c) = L_1^D(\phi, c)/2 \leq \alpha$. Therefore (α, β)-agnostic learning with respect to distribution D can be seen as learning of the set of distributions $\mathcal{D} = \{(D, \phi) \mid \phi \in B_1^D(C, 2\alpha)\}$ with error of at most β. This observation allows us to apply the characterizations from Section 3 after the straightforward generalization of SQSD and SQ-SDIM to general sets of real-valued functions. Namely, for a set of real-valued functions F, we define

$$\text{SQSD}(F, D, \epsilon, \gamma) = \sup_{\psi \in \mathcal{F}_1^\infty} \left\{ \text{SQD}(F \setminus B_1^D(\texttt{sign}(\psi), 2\epsilon) - \psi, D, \gamma) \right\} \ .$$

The SQ-SDIM(F, D, ϵ) is defined analogously. It is easy to see that when F contains only $\{-1, 1\}$ functions these generalized definitions are identical to Definitions 3.1 and 3.9.

We can now characterize the query complexity of (α, β)-agnostic SQ learning using $\text{SQSD}(B_1^D(C, 2 \cdot \alpha), D, \beta, \gamma)$ in exactly the same way as SLC is characterized using $\text{SQSD}(C, D, \epsilon, \gamma)$. Formally, we obtain the following theorem.

Theorem 4.1 *Let C be a concept class, D be a distribution D over X and $0 < \alpha \leq \beta \leq 1/2$. Let d be the smallest number of SQs of tolerance τ sufficient to (α, β)-agnostically learn C. Then*

1. *$d \geq SQSD(B_1^D(C, 2 \cdot \alpha), D, \beta + \tau, \tau) - 2$,*
2. *$d \leq SQSD(B_1^D(C, 2 \cdot \alpha), D, \beta, 4 \cdot \tau)/(3\tau^2)$.*

To prove Theorem 4.1 we only need to observe that the proofs of Theorems 3.2 and 3.4 do not assume that the concept class C contains only Boolean functions and hold for any class of functions contained in $\mathcal{F}_1^\infty$. To obtain a characterization of (α, β)-agnostic SQ learning using SQ-SDIM we would also need to extend Theorem 3.10 to general sets of functions in $\mathcal{F}_1^\infty$.

Theorem 4.2 *Let $F \subseteq \mathcal{F}_1^\infty$ be a set of functions, D be a distribution over X, $\epsilon > 0$ and $d = SQ\text{-}SDIM(F, D, \epsilon)$. Then $SQSD(F, D, \epsilon, 1/(2d)) \leq d$ and $SQSD(F, D, \epsilon, 2d^{-1/5}) \geq (\epsilon d)^{1/3}/2$.*

Proof: We first observe that the first part of the proof of Theorem 3.10 can be used verbatim for more general sets of functions. However the proof of the second part relies on a lower bound of $\sqrt{\epsilon}$ on the $\|\cdot\|_D$-norm of every function in $F_\psi = F \setminus B^D(\texttt{sign}(\psi), \epsilon) - \psi$. In place of this lower bound we observe that if there exists a function $f \in F_\psi$ such that $\|f\|_D \leq 2d^{-1/5}$ then there does

12

not exist $g \in \mathcal{F}_1^\infty$ such that $\langle f, g \rangle_D \geq 2d^{-1/5}$) and, in particular, $\text{SQD}(F_\psi, D, \epsilon, 2d^{-1/5}) = \infty$. This would imply the claim. Otherwise (when $\|f\|_D \geq 2d^{-1/5}$ for all $f \in F_\psi$), we can apply Lemma 3.8 (with $m = 2d^{-1/5}$) to obtain $\text{SQD}(F_\psi, D, 2^{1/3}d^{-1/5}) \geq (\epsilon d)^{1/3}/2$. In either case we obtain that $\text{SQSD}(F, D, \epsilon, 2d^{-1/5}) \geq (\epsilon d)^{1/3}/2$ $\square$

While we can now use SQSD or SQ-SDIM to characterize SQ learnability in the basic agnostic model a simpler approach to characterization is suggested by recent distribution-specific agnostic boosting algorithms [17, 21]. Formally, a *weak agnostic* learning algorithm is an algorithm that can recover at least a polynomial fraction of the advantage over the random guessing of the best approximating function in C. Specifically, on a distribution $A = (D, \phi)$ it produces a hypothesis h such that $\langle h, \phi \rangle_D \geq p(1/n, 1 - 2\Delta(A, C))$ for some polynomial $p(\cdot, \cdot)$. Distribution-specific agnostic boosting algorithms of Kalai and Kanade [21] and Feldman [17] imply the equivalence of weak and strong distribution-specific agnostic learning.

Theorem 4.3 ([17, 21]) *Let C be a concept class and D be a distribution over X. If C is efficiently weakly agnostically learnable over D then C is agnostically learnable over D.*

This result is proved only for the example-based agnostic learning but, as with other boosting algorithms, it can be easily translated to the SQ model (*cf.* [1]). Given Theorem 4.3, we can use the known characterizations of weak learning together with our simple observation to characterize the (strong) agnostic SQ learning using either SQD or SQ-DIM.

Theorem 4.4 *Let C be a concept class and D be a distribution over X. There exists a polynomial $p(\cdot, \cdot)$ such that $ASLC(C, D, \epsilon, 1/p(n, 1/\epsilon)) \leq p(n, 1/\epsilon)$ if and only if there exists a polynomial polynomial $p'(\cdot, \cdot)$ such that for every $1 > \Gamma > 0$, $SQD(B_1^D(C, 1 - \Gamma), D, 1/p'(n, 1/\Gamma)) \leq p'(n, 1/\Gamma)$.*

Proof: The proof is essentially the same as the characterization of weak learning by Bshouty and Feldman [10]. We review it briefly for completeness. Given $\Gamma > 0$ and an agnostic learning algorithm $\mathcal{A}$ for C, we simulate $\mathcal{A}$ with $\epsilon = \Gamma/4$ as in the proof of Theorem 3.2 for $\psi \equiv 0$. Let G be the set containing the correlational queries obtained from $\mathcal{A}$ and the final hypothesis. By the same analysis as in the proof of Theorem 3.2, the size of G is upper-bounded by a polynomial in n and $1/\epsilon = 4/\Gamma$. Further, for every $\phi \in B_1^D(C, 1 - \Gamma)$, there exists $g \in G$ such that $|\langle g, \phi \rangle_D| \geq \min\{\tau, \Gamma - 2\epsilon\} = \min\{\tau, \Gamma/2\}$. The tolerance of the learning algorithm is lower bounded by the inverse of a polynomial (in n and $1/\Gamma$) and therefore we obtain the first direction of the claim.

If for every $\Gamma > 0$, $\text{SQD}(B_1^D(C, 1 - \Gamma), D, 1/p'(n, 1/\Gamma)) \leq p'(n, 1/\Gamma)$ then C can be weakly agnostically SQ learned by the following algorithm. First, ask the query $g \cdot \ell$ with tolerance $1/(3p'(n, 1/\Gamma))$ for each function g in the approximating set G. Let $v(g)$ denote the answer to the query for g. For a distribution $A = (D, \phi)$, $\mathbf{E}_A[g(x) \cdot b] = \langle g, \phi \rangle_D$ and therefore $|v(g) - \langle g, \phi \rangle_D| \leq 1/(3p'(n, 1/\Gamma))$. By choosing $g' = \text{argmax}_{g \in G}\{|v(g)|\}$ we are guaranteed that $|\langle g', \phi \rangle_D| \geq 1/(3p'(n, 1/\Gamma))$. Therefore $\text{sign}(v(g')) \cdot g'$ is a weak hypothesis for f. Finally, we can appeal to Theorem 4.3 to convert this weak agnostic learning algorithm to a strong agnostic learning algorithm for C over D. $\square$

As before, we can now obtain an SQ-DIM–based characterization from the SQD–based one.

Theorem 4.5 *Let C be a concept class and D be a distribution over X. There exists a polynomial $p(\cdot, \cdot)$ such that $ASLC(C, D, \epsilon, 1/p(n, 1/\epsilon)) \leq p(n, 1/\epsilon)$ if and only if there exists a polynomial polynomial $p'(\cdot, \cdot)$ such that for every $1 > \Gamma > 0$, $SQ\text{-}DIM(B_1^D(C, 1 - \Gamma), D) \leq p'(n, 1/\Gamma)$.*

Proof: Let $d = \text{SQ-DIM}(B_1^D(C, 1 - \Gamma), D)$. Every function $f \in B_1^D(C, 1 - \Gamma)$ satisfies $\|f\|_D \geq \mathbf{E}_D[|f|] \geq \Gamma$. Therefore, Lemma 3.8 implies that $\text{SQD}(B_1^D(C, 1 - \Gamma), D, (\Gamma^2 d)^{-1/3}) \geq (\Gamma^2 d)^{1/3}/2$.

This implies that $d \leq p_1(\mathrm{SQD}(B_1^P(C, 1-\Gamma), D, 1/p_2(n, 1/\Gamma)), 1/\Gamma)$ for some polynomials $p_1(\cdot, \cdot)$ and $p_2(\cdot, \cdot)$. As in the case of concept classes, it follows immediately from the definition that $d \geq \mathrm{SQD}(B_1^P(C, 1-\Gamma), D, 1/d)$. These bounds together with Theorem 4.4 imply the claim. $\square$

We now give a simple example of the use of this characterization. For $X = \{0,1\}^n$, let U denote the uniform distribution over $\{0,1\}^n$ and let $C_{n,k}$ denote the concept class of all monotone conjunctions of at most k Boolean variables.

Theorem 4.6 *For every $k = \omega(1)$, the concept class $C_{n,k}$ is not efficiently agnostically SQ learnable over the uniform distribution U.*

Proof: Let χ_T denote the parity function of the variables with indices in $T \subseteq [n]$. Let c_T denote the monotone conjunction of the same set of variables. $\mathbf{Pr}_U[\chi_T(x) \neq c_T(x)] = 1/2 - 2^{-|T|}$ and therefore $L_1^U(\chi_T, c_T) = 1 - 2^{-|T|+1}$. In particular, for $P_{n,k} = \{\chi_T \mid |T| \leq k\}$, we get $P_{n,k} \subseteq B_1^U(C_{n,k}, 1 - 2^{-k+1}))$. For any two distinct parity functions χ_S and χ_T, $\langle \chi_S, \chi_T \rangle_U = 0$ and therefore $\mathrm{SQ\text{-}DIM}(B_1^U(C_{n,k}, 1 - 2^{-k+1}), U) \geq |P_k| = \sum_{i \leq k} \binom{n}{i}$. By choosing $\Gamma = 1/n$ we obtain that $\mathrm{SQ\text{-}DIM}(B_1^U(C_{n,k}, 1 - \Gamma), U) = n^{\omega(1)}$. Theorem 4.5 now implies the claim. $\square$

5 Applications to Evolvability

In this section we use the characterization of SQ learnability and the analysis in the proof of Theorem 3.4 to derive a new type of evolutionary algorithms in Valiant's framework of evolvability [35].

5.1 Overview of the Model

We start by presenting a brief overview of the model. For detailed description and intuition behind the various choices made in model the reader is referred to [35, 16]. The goal of the model is to specify how organisms can acquire complex mechanisms via a resource-efficient process based on random mutations and guided by fitness-based selection. The mechanisms are described in terms of the multi argument functions they implement. The fitness of such a mechanism is measured by evaluating the correlation of the mechanism with some "ideal" behavior function. The value of the "ideal" function on some input describes the most beneficial behavior for the condition represented by the input. The evaluation of the correlation with the "ideal" function is derived by evaluating the function on a moderate number of inputs drawn from a probability distribution over the conditions that arise. These evaluations correspond to the experiences of one or more organisms that embody the mechanism. A specific "ideal" function and a distribution over the domain of inputs effectively define a fitness landscape over all functions.

Random variation is modeled by the existence of an explicit algorithm that acts on some fixed representation of mechanisms and for each representation of a mechanism produces representations of mutated versions of the mechanism. The model essentially does not place any restrictions on the mutation algorithm other than it being efficiently implementable. Selection is modeled by an explicit rule that determines the probabilities with which each of the mutations of a mechanism will be chosen to "survive" based on the fitness of all the mutations of the mechanism and the probabilities with which each of the mutations is produced by the mutation algorithm.

As can be seen from the above description, a fitness landscape (given by a specific "ideal" function and a distribution over the domain), a mutation algorithm, and a selection rule jointly

14

determine how each step of an evolutionary process is performed. A class of functions C is considered evolvable in selection rule `Sel` with respect to a distribution D over the domain if there exist a representation of mechanisms R and a mutation algorithm M such that for every "ideal" function $f \in C$, a sequence of evolutionary steps starting from any representation in R and performed according to f, D, M and selection rule `Sel` converges in a polynomial number of steps to f. The convergence is defined as achieving fitness (which is the correlation with f over D) of at least $1 - \epsilon$ for some $\epsilon > 0$ referred to as the *accuracy* parameter. This process is essentially PAC learning of C over distribution D with the selection rule (rather than explicit examples) providing the only target-specific feedback. An evolvable class of functions C represents the complexity of structures that can evolve in a single phase of evolution driven by a single "ideal" function. We now define the model formally using the notation from [16].

5.2 Definition of Evolvability

The description of a mutation algorithm $\mathcal{A}$ consists of the definition of the representation class R of possibly randomized hypotheses in $\mathcal{F}_1^\infty$ and the description of polynomial time algorithm that for every $r \in R$ and $\epsilon > 0$ outputs a random mutation of r

Definition 5.1 *A mutation algorithm $\mathcal{A}$ is defined by a pair (R, M) where*

- *R is a representation class of functions over X with range in $[-1, 1]$.*

- *M is a randomized polynomial time Turing machine that, given $r \in R$ and $1/\epsilon$ as input, outputs a representation $r_1 \in R$ with probability $\mathbf{Pr}_{\mathcal{A}}(r, r_1)$. The set of representations that can be output by $M(r, \epsilon)$ is referred to as the* neighborhood *of r for ϵ and denoted by* `Neigh`$_{\mathcal{A}}(r, \epsilon)$.

A *loss function L* is a non-negative mapping $L : Y \times Y \to \mathbb{R}^+$. $L(y, y')$ measures the "distance" between the desired value y and the predicted value y'. We will discuss *linear loss $L_1(y, y') = |y' - y|$* and the *quadratic loss $L_Q(y, y') = (y' - y)^2$* functions. For a function $\phi \in \mathcal{F}_1^\infty$ its fitness (also referred to as *performance* in earlier work) relative to loss function L, distribution D over the domain and target function f is defined as

$$L\mathrm{Perf}_f(\phi, D) = 1 - 2 \cdot \mathbf{E}_D[L(f(x), \phi(x))]/L(-1, 1) \ .$$

For an integer s, functions $\phi, f \in \mathcal{F}_1^\infty$ over X, distribution D over X and loss function L, the *empirical fitness $L\mathrm{Perf}_f(\phi, D, s)$* of ϕ is a random variable that equals $1 - \frac{1}{s}\frac{2}{L(-1,1)}\sum_{i \in [s]} L(f(z_i), \phi(z_i))$ for $z_1, z_2, \ldots, z_s \in X$ chosen randomly and independently according to D.

A number of natural ways of modeling selection were discussed in prior work [35, 16]. For concreteness here we use the selection rule used in Valiant's main definition in a slightly generalized version from [16]. In selection rule `SelNB`$[L, t, p, s]$ p candidate mutations are sampled using the mutation algorithm. Then beneficial and neutral mutations are defined on the basis of their empirical fitness $L\mathrm{Perf}$ in s experiments (or examples) using tolerance t. If some beneficial mutations are available one is chosen randomly according to their relative frequencies in the candidate pool. If none is available then one of the neutral mutations is output randomly according to their relative frequencies. If neither neutral or beneficial mutations are available, $\bot$ is output to mean that no mutation "survived".

Definition 5.2 *For a loss function L, tolerance t, candidate pool size p, sample size s, selection rule `SelNB`$[L, t, p, s]$ is an algorithm that for any function f, distribution D, mutation algorithm $\mathcal{A} = (R, M)$, a representation $r \in R$, accuracy ϵ, `SelNB`$[L, t, p, s](f, D, \mathcal{A}, r)$ outputs a random*

variable that takes a value r_1 determined as follows. First run $M(r, \epsilon)$ p times and let Z be the set of representations obtained. For $r' \in Z$, let $\mathbf{Pr}_Z(r')$ be the relative frequency with which r' was generated among the p observed representations. For each $r' \in Z \cup \{r\}$, compute an empirical value of fitness $v(r') = L\mathtt{Perf}_f(r', D, s)$. Let $\mathtt{Bene}(Z) = \{r' \mid v(r') \geq v(r) + t\}$ and $\mathtt{Neut}(Z) = \{r' \mid |v(r') - v(r)| < t\}$. Then

(i) if $\mathtt{Bene}(Z) \neq \emptyset$ then output $r_1 \in \mathtt{Bene}$ with probability $\mathbf{Pr}_Z(r_1)/\sum_{r' \in \mathtt{Bene}(Z)} \mathbf{Pr}_Z(r')$;

(ii) if $\mathtt{Bene}(Z) = \emptyset$ and $\mathtt{Neut}(Z) \neq \emptyset$ then output $r_1 \in \mathtt{Neut}(Z)$ with probability $\mathbf{Pr}_Z(r_1)/\sum_{r' \in \mathtt{Neut}(Z)} \mathbf{Pr}_Z(r')$.

(iii) If $\mathtt{Neut}(Z) \cup \mathtt{Bene}(Z) = \emptyset$ then output $\bot$.

A concept class C is said to be evolvable by a mutation algorithm A guided by a selection rule $\mathtt{Sel}$ over distribution D if for every target concept $f \in C$, mutation steps as defined by A and guided by $\mathtt{Sel}$ will converge to f.

Definition 5.3 *For concept class C over X, distribution D, mutation algorithm $\mathcal{A}$, loss function L and a selection rule $\mathtt{Sel}$ based on $L\mathtt{Perf}$ we say that the class C is evolvable over D by $\mathcal{A}$ in $\mathtt{Sel}$ if there exists a polynomial $g(n, 1/\epsilon)$ such that for every n, $f \in C$, $\epsilon > 0$, and every $r_0 \in R$, with probability at least $1 - \epsilon$, a sequence $r_0, r_1, r_2, \ldots$, where $r_i \leftarrow \mathtt{Sel}(f, D, \mathcal{A}, r_{i-1})$ will have $L\mathtt{Perf}_f(r_{g(n,1/\epsilon)}, D) > 1 - \epsilon$. We refer to the algorithm obtained as evolutionary algorithm $(\mathcal{A}, \mathtt{Sel})$.*

We say that an evolutionary algorithm $(\mathcal{A}, \mathtt{Sel})$ evolves C over D *monotonically* if with probability at least $1 - \epsilon$, for every $i \leq g(n, 1/\epsilon)$, $L\mathtt{Perf}_f(r_i, D) \geq L\mathtt{Perf}_f(r_0, D)$, where $g(n, 1/\epsilon)$ and $r_0, r_1, r_2, \ldots$ are defined as above.

As in PAC learning, we say that a concept class C is evolvable in $\mathtt{Sel}$ if it is evolvable over all distributions by a single evolutionary algorithm (we emphasize this by saying *distribution-independently* evolvable). A more relaxed notion of evolvability requires convergence only when the evolution starts from a single fixed representation r_0. Such evolvability is referred to as evolvability *with initialization*.

5.3 Monotone Distribution-Specific Evolvability from SQ Learning Algorithms

In our earlier work [16] it was shown that every SQ learnable concept class C is evolvable in $\mathtt{SelNB}[L_Q, t, p, s]$ (that is the basic selection rule with quadratic loss) for some polynomials $p(n, 1/\epsilon)$ and $s(n, 1/\epsilon)$ and an inverse polynomial $t(n, 1/\epsilon)$. The mutation algorithms obtained in this result do not require initialization but instead are based on a form of implicit initialization that involves gradual reduction of fitness to 0 if the process of evolution is not started in some fixed r_0. Such "deliberate" gradual reduction in fitness appears to be somewhat unnatural. Hence we consider the question of whether it is possible to evolve from any starting representation without the need for such implicit initialization and fitness decreases in general in other words, which concept classes are evolvable monotonically. In this section we show that monotone evolutionary algorithms exist for every SQ learnable concept class when evolving with respect to a fixed distribution D and using the quadratic loss function.

The key element of the proof of this result is essentially an observation that the SQ algorithm that we designed in the proof Theorem 3.4 can be seen as repeatedly testing a small set of candidate hypotheses, and choosing one that reduces the $\| \cdot \|_D^2$ distance to the target function. Converting such an algorithm to an evolutionary algorithm is a rather straightforward process. First we show that Theorem 3.2 gives a way to compute a neighborhood of every function ψ

that always contains a function with fitness higher than ψ (unless the fitness of ψ is close to the optimum).

Theorem 5.4 *Let C be a concept class over X and D be a distribution. If C is efficiently SQ learnable over D then there exists an algorithm $\mathcal{N}$ that for every $\epsilon > 0$, given ϵ, access to D and a circuit for $\psi \in \mathcal{F}_1^\infty$ can produce a set of functions $N(\psi, \epsilon)$ such that*

1. *For every $f \in C$, there exists $\phi \in N(\psi, \epsilon)$ such that*

$$\|f - \phi\|_D^2 \le \min\{\epsilon, \|f - \psi\|_D^2 - \theta(n, 1/\epsilon)\},$$

for an inverse-polynomial $\theta(\cdot, \cdot)$;

2. *the size of $N(\psi, \epsilon)$ is polynomial in n and $1/\epsilon$;*

3. *the circuit size of every function in $N(\psi, \epsilon)$ is (additively) larger than the circuit size of ψ by at most a polynomial in n and $1/\epsilon$;*

4. *the running time of $\mathcal{N}$ is polynomial in n, $1/\epsilon$ and the circuit size of ψ.*

Proof: We use Theorem 3.6 to obtain an algorithm $\mathcal{B}$ that given a circuit for ψ and access to D, efficiently constructs set G_ψ of polynomial size for some inverse polynomial $\gamma(n, 1/\epsilon)$. Let $G_\psi(\epsilon/4)$ be the output of $\mathcal{B}$ with its accuracy parameter set to $\epsilon/4$. Now let

$$N(\psi, \epsilon) = \{P_1(\psi + \gamma \cdot g) \mid g \in G_\psi(\epsilon/4)\} \bigcup \{P_1(\psi - \gamma \cdot g) \mid g \in G_\psi(\epsilon/4)\} \cup \{\texttt{sign}(\psi)\} .$$

By the properties of $G_\psi(\epsilon/4)$, for every $f \in C$, either there exists a function $g \in G_\psi(\epsilon/4)$ such that $|\langle f - \psi, g \rangle_D| \ge \gamma(n, 4/\epsilon)$, or $\mathbf{Pr}_D[f \ne \texttt{sign}(\psi)] \le \epsilon/4$. In the first case, by the analysis in the proof of Theorem 3.4, $\psi_g = P_1(\psi + b \cdot \gamma(n, 4/\epsilon) \cdot g)$ satisfies $\|f - \psi_g\|_D^2 \le \|f - \psi\|_D^2 - \gamma(n, 4/\epsilon)^2$ for $b = \texttt{sign}(\langle f - \psi, g \rangle_D)$. In the second case, $\|\texttt{sign}(\psi) - f\|_D^2 \le 4 \cdot \epsilon/4 = \epsilon$. Theorem 3.6 also implies that the algorithm that we have defined satisfies the bounds in conditions (2)-(4). $\square$

An immediate corollary of this result is monotone evolvability of every SQ-learnable concept class in $\texttt{SelNB}[L_Q, t, p, s]$ over any fixed distribution D.

Theorem 5.5 *Let D be a distribution and C be a concept class efficiently SQ learnable over D. There exist polynomials $p(n, 1/\epsilon)$ and $s(n, 1/\epsilon)$, an inverse polynomial $t(n, 1/\epsilon)$ and a mutation algorithm $\mathcal{A} = (R, M)$ such that C is evolvable monotonically by $\mathcal{A}$ over D in $\texttt{SelNB}[L_Q, t(n, 1/\epsilon), p(n, 1/\epsilon), s(n, 1/\epsilon)]$. Here if D is not efficiently samplable then $\mathcal{A}$ is a non-uniform algorithm.*

Proof: Let R be the representation class containing all circuits over X and let r be any representation in R. Given r and $1/\epsilon$ the algorithm M uses the algorithm $\mathcal{N}$ from Theorem 5.4 with parameters r and ϵ to obtain $N(r, \epsilon)$. The algorithm $\mathcal{N}$ requires access to distribution D and can be simulated efficiently if D is efficiently samplable or simulated using a fixed random sample of points from D otherwise (as it is done for example in [14]). The algorithm M outputs a randomly and uniformly chosen representation in $N(r, \epsilon)$.

In order for this evolutionary algorithm to work we need to make sure that a representation with the highest fitness in $N(r, \epsilon)$ is present in the candidate pool and that the fitness of each candidate mutation is estimated sufficiently accurately. We denote a representation with the highest fitness by r^*. The bound on the number of generations that we are going to prove is $g(n, 1/\epsilon) = 8/\theta$. To ensure that r^* is with probability at least $1 - \epsilon/4$ in the candidate pool in every generation we set $p(n, 1/\epsilon) = |N(r, \epsilon)| \cdot \ln \frac{4 \cdot g(n, 1/\epsilon)}{\epsilon}$. To ensure that with probability at least $1 - \epsilon/4$ in every generation the fitness of each mutation is estimated within $\theta(n, 1/\epsilon)/8$ we set $s(n, 1/\epsilon) = c \cdot \theta(n, 1/\epsilon)^{-2} \cdot \log \frac{8 \cdot p(n, 1/\epsilon) \cdot g(n, 1/\epsilon)}{\epsilon}$ for a constant c (obtained via the Hoeffding's bound). We set the tolerance of the selection rule to $t(n, 1/\epsilon) = 3 \cdot \theta(n, 1/\epsilon)/8$.

By the properties of $\mathcal{N}$, $L_Q\text{Perf}_f(r^*, D) \geq \min\{L_Q\text{Perf}_f(r, D) + \theta(n, 1/\epsilon)/2, 1 - \epsilon/2\}$. If $L_Q\text{Perf}_f(r, D) \leq 1 - \epsilon$ then $L_Q\text{Perf}_f(r^*, D) \geq L_Q\text{Perf}_f(r, D) + \theta(n, 1/\epsilon)/2$ (without loss of generality $\theta(n, 1/\epsilon) \leq \epsilon$). In this case if r^* is in the pool of candidates Z and the empirical fitness of every mutation in Z is within $\theta(n, 1/\epsilon)/8$ of the true fitness then $\text{Bene}_Z(r)$ is non-empty and for every $r' \in \text{Bene}_Z(r)$, $L_Q\text{Perf}_f(r', D) \geq L_Q\text{Perf}_f(r, D) + \theta(n, 1/\epsilon)/4$. In particular, the output of $\text{SelNB}[L_Q, t(n, 1/\epsilon), p(n, 1/\epsilon), s(n, 1/\epsilon)]$ will have fitness at least $L_Q\text{Perf}_f(r, D) + \theta(n, 1/\epsilon)/4$. The lowest initial fitness is -1 and therefore, with probability at least $1 - \epsilon/2$, after at most $g(n, 1/\epsilon) = 8/\theta(n, 1/\epsilon)$ steps a representation with fitness at least $1 - \epsilon$ will be reached.

We also need to establish that once the fitness of at least $1 - \epsilon$ is reached it does not decrease within $g(n, 1/\epsilon)$ steps and also prove that the evolution algorithm is monotone. To ensure this we modify slightly the mutation algorithm M. The algorithm M' outputs a randomly and uniformly chosen representation in $N(r, \epsilon)$ with probability $\Delta = \epsilon/(2 \cdot g(n, 1/\epsilon))$ and outputs r with probability $1 - \Delta$. We also increase $p(n, 1/\epsilon)$ accordingly to ensure that r^* is still in the pool of candidates with sufficiently high probability. This change does not influence the analysis when $\text{Bene}_Z(r)$ is non-empty. If $\text{Bene}_Z(r)$ is empty then, by the definition of M', $\text{SelNB}[L_Q, t(n, 1/\epsilon), p(n, 1/\epsilon), s(n, 1/\epsilon)]$ will output r with probability at least $1 - \Delta$. That is in every step, either the fitness improves or it does not change with probability at least $1 - \Delta$. In particular, with probability at least $1 - \epsilon/2$ the fitness will not decrease during any of the first $g(n, 1/\epsilon)$ generations. $\qquad\square$

5.4 Distribution-Independent Evolvability of Disjunctions

A substantial limitation of the general transformation given in the previous section is that the mutation algorithm given there requires access to D and hence only implies evolvability for a fixed distribution. In this section we show that for the concept class of disjunctions (and conjunctions) the ideas of the transformation in Section 5.3 can be used to derive a simple algorithm for distribution-independent monotone evolvability of disjunctions.

As usual in distribution-independent learning, we can assume that the disjunction is monotone. We represent a monotone disjunction by a subset $T \subset [n]$ containing the indices of the variables in the disjunction and refer to it as t_T. In addition, we define $\theta_T = (t_T + 1)/2$ and for every $i \in [n]$, let $\theta_i(x) = (1 + x_i)/2$ (θ_T and θ_i are simply the $\{0, 1\}$ versions of t_T and x_i). Given a current hypothesis computing function $\phi \in \mathcal{F}_1^\infty$ we try to modify it in two ways. The first one is to add $\gamma \cdot \theta_i(x)$ and project using P_1 for some $i \in [n]$ and $\gamma > 0$. The other one is to subtract γ and project using P_1. The purpose of the first type of modification is to increase fitness on points where the target disjunction equals to 1. It is easy to see that such steps can make the fitness on such points as close to 1 as desired. The problem with such steps is that they might also add $\gamma \cdot \theta_i$ such that x_i is not in the target disjunction and thereby decrease the fitness on points where the target equals -1. We fix this by using the second type of modification. This modification increases the fitness on points where the target equals -1 but may decrease the fitness on points where the target equals 1. The reason why this combination of modifications will converge to a good hypothesis is that for the quadratic loss function the change in loss due to an update is larger on points where the loss is larger. Namely, $L_Q(y, y' + \Delta) = L_Q(y, y') + 2 \cdot \Delta \cdot (y - y') + \Delta^2$. This means that if the first type of modification can no longer improve fitness then the second type will. We formalize this argument in the lemma below.

Lemma 5.6 For $\phi(x) \in \mathcal{F}_1^\infty$, let $N_\gamma(\phi) = \{P_1(\phi + \gamma \cdot \theta_i) \mid i \in [n]\} \cup \{\phi, P_1(\phi - \gamma)\}$. There exist inverse polynomial $\tau(\cdot, \cdot)$ and $\gamma(\cdot, \cdot)$ such that for every distribution D over $\{0, 1\}^n$, every target monotone disjunction f, every $\epsilon > 0$ and every $\phi(x) \in \mathcal{F}_1^\infty$ there exists $\phi' \in N_{\gamma(n, 1/\epsilon)}(\phi)$

for which
$$L_Q \text{Perf}_f(\phi', D) \geq \min\{L_Q \text{Perf}_f(\phi, D) + \tau(n, 1/\epsilon), 1 - \epsilon\} \ .$$

Proof: Let $f = t_T$ denote the target monotone disjunction. By the definition $\|f - \phi\|_D^2 = 2(1 - L_Q \text{Perf}_f(\phi, D))$. We denote the loss of ϕ when f restricted to 1 and -1 by $\Delta_1 = \mathbf{E}_D[(f - \phi)^2 \cdot (f + 1)/2]$ and $\Delta_{-1} = \mathbf{E}_D[(f - \phi)^2 \cdot (f - 1)/2]$ respectively. Let $\gamma = \epsilon^{3/2}/21$ and $\tau = \gamma^4/(8n)$. We split the analysis into several cases.

1. If $L_Q \text{Perf}_f(\phi, D) \geq 1 - \epsilon$ then $\phi' = \phi$ satisfies the condition.

2. $L_Q \text{Perf}_f(\phi, D) \leq 1 - \epsilon$ and $\Delta_1 \geq 2\gamma^2$. In this case,
$$\Delta_1 \leq \mathbf{Pr}_D[f(x) = 1, \ \phi(x) \geq 1 - \gamma] \cdot \gamma^2 + \mathbf{Pr}_D[f(x) = 1, \ \phi(x) < 1 - \gamma] \cdot 4 \ .$$

Therefore
$$\mathbf{Pr}_D[f(x) = 1, \ \phi(x) < 1 - \gamma] \geq (\Delta_1 - \gamma^2)/4 \geq \gamma^2/4 \ .$$

The target function is a disjunction of at most n variables therefore there exists $i \in T$ such that $\mathbf{Pr}_D[x_i = 1, \ \phi(x) < 1 - \gamma] \geq \gamma^2/(4n)$. For such i, let $\phi' = P_1(\phi + \gamma \cdot \theta_i)$. Note that for every point x, the loss of $\phi'(x)$ is at most the loss of $\phi(x)$ while for every point where $x_i = 1$ and $\phi(x) < 1 - \gamma$ the loss of $\phi'(x)$ is smaller than the loss of $\phi(x)$ by at least γ^2. Therefore,
$$\|f - \phi'\|_D^2 \leq \|f - \phi\|_D^2 - \gamma^2 \cdot \mathbf{Pr}_D[x_i = 1, \ \phi(x) < 1 - \gamma] \leq \|f - \phi\|_D^2 - \frac{\gamma^4}{(4n)} \ .$$

This implies that
$$L_Q \text{Perf}_f(\phi', D) \geq L_Q \text{Perf}_f(\phi, D) + \tau(n, 1/\epsilon)$$

for τ defined as above.

3. $L_Q \text{Perf}_f(\phi, D) \leq 1 - \epsilon$ and $\Delta_1 < 2\gamma^2$. In this case $\Delta_{-1} \geq 2\epsilon - \Delta_1 > 3 \cdot \epsilon/2$. Let $\phi' = P_1(\phi - \gamma)$. We now upper bound the increase in error on points where $f = 1$ and lower bound the decrease in error on points where $f = -1$. For the upper bound we have
$$\mathbf{E}_D[(f - \phi - \gamma)^2] \leq 2 \cdot \mathbf{E}_D[(f - \phi)^2] + 2 \cdot \gamma^2,$$

and therefore the increase in error when $f = 1$ is at most $\Delta_1 + 2 \cdot \gamma^2 \leq 4 \cdot \gamma^2$. For the lower bound similarly to the previous case we get the inequality
$$\Delta_{-1} \leq \mathbf{Pr}_D[f(x) = -1, \ \phi(x) \leq -1 + \sqrt{\epsilon}/2] \cdot \epsilon/2 + \mathbf{Pr}_D[f(x) = -1, \ \phi(x) > -1 + \sqrt{\epsilon}/2] \cdot 4 \ .$$

Therefore
$$\mathbf{Pr}_D[f(x) = -1, \ \phi(x) > -1 + \sqrt{\epsilon}/2] \geq (\Delta_{-1} - \epsilon/2)/4 \geq \epsilon/4 \ . \tag{1}$$

On every point x where $f(x) = -1$ and $\phi(x) > -1 + \sqrt{\epsilon}/2$,
$$|f(x) - \phi'(x)|^2 \leq |f(x) - \phi(x)|^2 - (2\gamma(\phi(x) - f(x)) - \gamma^2) \leq |f(x) - \phi(x)|^2 - 2\gamma\sqrt{\epsilon}/2 + \gamma^2 \ .$$

By combining this with equation (1) and our choice of $\gamma = \epsilon^{3/2}/21$ we get
$$\|f - \phi'\|_D^2 \leq \|f - \phi\|_D^2 - \frac{\epsilon}{4} \cdot (\gamma\sqrt{\epsilon} - \gamma^2) \leq \|f - \phi\|_D^2 - 5 \cdot \gamma^2 \ .$$

Therefore in this case
$$L_Q \text{Perf}_f(\phi', D) \geq L_Q \text{Perf}_f(\phi, D) + (5 \cdot \gamma^2 - 4 \cdot \gamma^2)/2 \geq L_Q \text{Perf}_f(\phi, D) + \tau(n, 1/\epsilon) \ .$$

19

□

The neighborhood $N_\gamma(\phi)$ can be computed efficiently and therefore Lemma 5.6 can be converted to an evolutionary algorithm in exactly the same way as it was done in Theorem 5.5. This implies monotone and distribution-independent evolvability of disjunctions in $\texttt{SelNB}[L_Q, t, p, s]$.

Theorem 5.7 *There exist polynomials $p(n, 1/\epsilon)$ and $s(n, 1/\epsilon)$, an inverse polynomial $t(n, 1/\epsilon)$ and a mutation algorithm $\mathcal{A} = (R, M)$ such that for every distribution D disjunctions are evolvable monotonically by $\mathcal{A}$ over D in $\texttt{SelNB}[L_Q, t(n, 1/\epsilon), p(n, 1/\epsilon), s(n, 1/\epsilon)]$.*

6 Discussion and Further Work

One natural question not covered in this work is whether and how our characterization can be applied to understanding of the SQ complexity of learning specific concept classes for which the previously known characterizations are not sufficient. As we explained in the introduction, one such example is learning of monotone functions. This question is addressed in a recent work [18], where the first lower bounds for SQ learning of depth-3 monotone formulas over the uniform distribution are derived using SQ-SDIM. The main open problem in this direction is evaluating the SQ-SDIM of monotone DNF over the uniform distribution.

As we have mentioned, another way to see our proof of Theorem 3.4 is as a boosting algorithm that instead of using a weak learning algorithm on different distributions uses a weak learning algorithm on different target functions (specifically on $f - \psi_i$ at iteration i). This perspective turned out to be useful for understanding of boosting in the agnostic learning framework. In particular, it has lead to the distribution-specific boosting algorithm given in Theorem 4.3 and to a new connection between agnostic and PAC learning.

We also believe that the insights into the structure of SQ learning given in this work will be useful in further exploration of Valiant's model of evolvability. For example, Theorem 5.4 can also be used to obtain distribution-specific evolvability of every SQ-learnable concept class with only very weak assumptions on the selection rule (such as (t, γ)-distinguishing defined in [16]). In addition, we believe that our results are not restricted to the quadratic loss function and can applied to evolvability with other loss functions. Perhaps, the most interesting question in this direction is whether results analogous to Theorem 5.5 can also be obtained for distribution-independent evolvablity. The distribution-indepedent evolvability of disjunctions given in Theorem 5.7 suggests that the answer might be "yes" for many other interesting concept classes. We hope that these questions will be investigated in subsequent work.

Acknowledgements

References

[1] J. Aslam and S. Decatur. General bounds on statistical query learning and pac learning with noise via hypothesis boosting. *Information and Computation*, 141(2):85–118, 1998.

[2] J. Aslam and S. Decatur. Specification and simulation of statistical query algorithms for efficiency and noise tolerance. *Journal of Computer and System Sciences*, 56:191–208, 1998.

[3] J. Balcázar, J. Castro, D. Guijarro, J. Köbler, and W. Lindner. A general dimension for query learning. *Journal of Computer and System Sciences*, 73(6):924–940, 2007.

[4] B. Barak, M. Hardt, and S. Kale. The uniform hardcore lemma via approximate bregman projections. In *Proceedings of SODA*, pages 1193–1200, 2009.

[5] A. Blum, C. Burch, and J. Langford. On learning monotone boolean functions. In *Proceedings of FOCS*, pages 408–415, 1998.

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

[7] 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.

[8] 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 *Proceedings of STOC*, pages 253–262, 1994.

[9] A. Blum, A. Kalai, and H. Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of the ACM*, 50(4):506–519, 2003.

[10] N. Bshouty and V. Feldman. On using extended statistical queries to avoid membership queries. *Journal of Machine Learning Research*, 2:359–395, 2002.

[11] N. Bshouty and C. Tamon. On the Fourier spectrum of monotone functions. *Journal of the ACM*, 43(4):747–770, 1996.

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

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

[14] V. Feldman. Evolvability from learning algorithms. In *Proceedings of STOC*, pages 619–628, 2008.

[15] V. Feldman. A complete characterization of statistical query learning with applications to evolvability. In *Proceedings of FOCS*, pages 375–384, 2009.

[16] V. Feldman. Robustness of evolvability. In *Proceedings of COLT*, pages 277–292, 2009.

[17] V. Feldman. Distribution-specific agnostic boosting. In *Proceedings of Innovations in Computer Science*, pages 375–384, 2010.

[18] V. Feldman, H. Lee, and R. Servedio. Lower bounds and hardness amplification for learning shallow monotone formulas. Manuscript, 2010.

[19] D. Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation*, 100(1):78–150, 1992.

[20] R. Impagliazzo. Hard-core distributions for somewhat hard problems. In *Proceedings of FOCS*, pages 538–545, 1995.

[21] A. Kalai and V. Kanade. Potential-based agnostic boosting. In *Proceeding of NIPS*, 2009.

[22] S. Kasiviswanathan, H. Lee, K. Nissim, S. Raskhodnikova, and A. Smith. What can we learn privately? In *Proceedings of FOCS*, pages 531–540, 2008.

[23] M. Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM*, 45(6):983–1006, 1998.

[24] M. Kearns, M. Li, and L. Valiant. Learning boolean formulas. *Journal of the ACM*, 41(6):1298–1328, 1994.

[25] M. Kearns, R. Schapire, and L. Sellie. Toward efficient agnostic learning. *Machine Learning*, 17(2-3):115–141, 1994.

[26] M. Kearns and L. Valiant. Cryptographic limitations on learning boolean formulae and finite automata. *Journal of the ACM*, 41(1):67–95, 1994.

[27] A. Klivans and R. Servedio. Boosting and hard-core set construction. *Machine Learning*, 51(3):217–238, 2003.

[28] A. Klivans and A. Sherstov. Improved lower bounds for learning intersections of halfspaces. In *Proceedings of COLT*, 2006.

[29] L. Michael. Evolving decision lists. Manuscript, 2007.

[30] N.Bansal, A. Blum, and S. Chawla. Correlation clustering. In *Proceedings of FOCS*, pages 238–247, 2002.

[31] A. A. Sherstov. Halfspace matrices. In *Proceedings of Conference on Computational Complexity*, pages 83–95, 2007.

[32] H. Simon. A characterization of strong learnability in the statistical query model. In *Proceedings of Symposium on Theoretical Aspects of Computer Science*, pages 393–404, 2007.

[33] B. Szörényi. Characterizing statistical query learning:simplified notions and proofs. In *ALT*, pages 186–200, 2009.

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

[35] L. G. Valiant. Evolvability. *Journal of the ACM*, 56(1):3.1–3.21, 2009.

[36] Ke Yang. New lower bounds for statistical query learning. *Journal of Computer and System Sciences*, 70(4):485–509, 2005.