Evolution with Recombination

Varun Kanade
SEAS
Harvard University
Cambridge, MA, USA
vkanade@fas.harvard.edu

***Abstract*—Valiant (2007) introduced a computational model of evolution and suggested that Darwinian evolution be studied in the framework of computational learning theory. Valiant describes evolution as a restricted form of learning where exploration is limited to a set of possible mutations and feedback is received through the survival of the *fittest* mutation. In subsequent work Feldman (2008) showed that evolvability in Valiant's model is equivalent to learning in the *correlational statistical query* (CSQ) model. We extend Valiant's model to include genetic recombination and show that in certain cases, recombination can significantly speed-up the process of evolution in terms of the number of generations, though at the expense of population size. This follows via a reduction from *parallel*-CSQ algorithms to evolution with recombination. This gives an exponential speed-up (in terms of the number of generations) over previous known results for evolving conjunctions and halfspaces with respect to restricted distributions.**

***Keywords*-evolvability; computational learning theory;**

I. Introduction

Darwinian evolution is one of the most important scientific theories and suggests that complex life-forms emerged from simpler ones. Yet, the nature of the complexity that can evolve in organisms and the processes therein are not well understood. The two central aspects of Darwin's theory are 1) creation of variation due to *mutations*, and 2) *natural selection* among the variants, a.k.a. survival of the fittest. It is now more or less understood that the underlying DNA sequence or genome of an organism contains code for proteins and also encodes rules governing their regulation. Thus, with some exceptions, the genome almost entirely controls the functions of an individual organism.

One example of a function encoded in the *genome* of an organism could be a circuit that decides the level of enzyme activity based on the environmental conditions (e.g. temperature, presence of oxygen etc.). A natural question that arises is, what is the complexity of functions that can be encoded in a genome, given that it must have occurred through Darwinian evolution? In a seminal paper, Valiant [1] proposed a computational model for evolution that captures the central ideas of mutation and natural selection. Valiant's work seeks to study evolution in the framework of computational learning theory, and understand the evolutionary process as a restricted form of learning. The goal of computational learning theory is to separate concept classes that can be efficiently learned from those that cannot. Examples of concept classes of interest include polynomial size decision trees, polynomial size DNFs, etc. Valiant [1] posed the same question in the context of evolution: what is the complexity of functions that can be evolved, or alternatively what concept classes are *evolvable*?

In subsequent work Feldman [2], [3] showed that evolvability in Valiant's model is equivalent to correlational statistical query (CSQ) learning and also that the evolution model is robust in the sense that changing the model slightly does not change what is *evolvable*. Kanade, Valiant and Vaughan [4] showed that Valiant's model of evolution is also robust with respect to *drift* in the target function. Thus, Valiant's model seems well-suited to study computational questions arising in evolution.

One of the most important aspects of the biological world not modeled explicitly by Valiant is the existence of two sexes and the process of recombination. Sexual reproduction is nearly universal in higher organisms and thus is thought to be an important factor in evolution. There are several proposed explanations for the role of sex and recombination. We discuss some of these in the related work section, but the most relevant argument for our work is the one that sexual reproduction can accelerate evolution through *parallelism*. Fisher [5] first proposed that sexual reproduction can speed up evolution (see also [6], [7]):

> A consequence of sexual reproduction which seems to be of fundamental importance to evolutionary theory is that advantageous changes in different structural elements of the germ plasm can be taken advantage of independently; whereas with asexual organisms either the genetic uniformity of the whole group must be such that evolutionary progress is greatly retarded, or if there is considerable genetic diversity, many beneficial changes will be lost through occurring in individuals destined to leave no ultimate descendants in the species.
> (from The Genetical Theory of Natural Selection - R. A. Fisher 1930)

A simple explanation for this is the following: Suppose that there are two allelic mutations $a \to A$ and $b \to B$ that

are both favorable and also additive in their effect, then an individual having both A and B alleles will be the fittest. However, beneficial mutations are extremely rare in nature. Under asexual reproduction an individual possessing both alleles A and B would appear, only if a mutation, say $b \rightarrow B$, occurs in an individual already possessing allele A. For this to occur with high probability, the A allele must have already spread in the population. For a large fraction of the population to acquire both A and B, several generations may be required. Under sexual reproduction, even if the two mutations $a \rightarrow A$ and $b \rightarrow B$ occur in different members of the population, via recombination there will be a member with both mutations in much fewer generations. Roughly speaking, if there are n loci on which selection acts additively, asexual reproduction may require $O(n)$ generations to produce the *fittest* variant, while sexual reproduction is able to achieve the same in only $O(\log(n))$. Our main result shows that recombination allows for *parallelism*,

Theorem 1. *If C is parallel* CSQ *learnable in T query steps, then C is evolvable under recombination in $O(T\ \log^2(n/\epsilon))$ generations.*

Using our result we can immediately show the following:

1) Under any fixed distribution, conjunctions can be evolved with recombination in $O(\log^2(n/\epsilon))$ generations. The best known result in Valiant's model requires $O^*(n)$ generations.
2) Under radially symmetric distributions and product Gaussian distributions, halfspaces passing through the origin can be evolved with recombination in $O(\log^2(n/\epsilon))$ generations. The best known result in Valiant's model requires $O^*(n)$ generations.

We briefly discuss some related work before describing our evolution model in detail and proving the main results.

Related Work: The extent of concept classes that can be evolved in Valiant's model is quite limited. Valiant [1] in his original paper proved that the class of monotone conjunctions is evolvable with respect to the uniform distribution. Feldman [2] proved that evolvability is equivalent to a restricted form of statistical query learning [8], called *correlational statistical query learning* [9]. He also showed that the class of singletons is evolvable distribution independently, which is the only known non-trivial result for distribution-independent evolution. Kanade, Valiant and Vaughan [4] showed that the class of homogeneous halfspaces is evolvable under radially symmetric distributions and also under product Gaussian distributions. However, Feldman [10] showed that the class of conjunctions is not evolvable in a distribution independent sense, thus showing that distribution-independent evolvability in Valiant's model is severely limited.

In an alternative line of research Michael [11] showed that under squared-loss feedback (Valiant's model uses *zero-one* loss), decision lists can be evolved under the uniform distribution using Fourier techniques. Feldman [3], [12] showed that under a large class of (nonlinear) loss functions evolvability is equivalent to statistical query learning. Feldman [10] also showed that linear threshold functions with a margin are evolvable distribution-independently under a large class of (nonlinear) loss functions. P. Valiant [13] has shown that for evolvability of real-valued functions, the class of linear and polynomial functions is evolvable under all convex loss functions (including linear loss) and all distributions.

Regarding the factors responsible for maintaining sex and recombination there seems to be no single answer. We have already discussed the argument that sexual reproduction can speed-up evolution. A different advantage of recombination was proposed by Muller [6] - recombination is useful as a means to weed out deleterious mutations. In a finite population mildly deleterious mutations are likely to be fixed in the population one at a time merely by chance and eventually a large number of these will accumulate. In the absence of recombination, a deleterious mutation cannot be removed except by a back-mutation which is extremely rare. This effect is known as Muller's ratchet. Epistasis refers to the non-independence between loci with respect to their effect on fitness. Hitchhiking is the phenomenon where certain (possibly deleterious) alleles are maintained because they are coupled to other beneficial mutations. Epistasis, hitchhiking and other factors are thought to play a role in the maintenance of recombination. Livnat et al. [14], [15] have suggested that sexual reproduction gives rise to modularity and mixability. Maynard Smith [16], [17] and Barton and Charlesworth [18] survey several proposed explanations for the maintenance of sex and recombination.

Our Contributions: In this work, we are interested in understanding how evolution in Valiant's model can be sped up in the presence of recombination. Thus, Fisher's ideas are most relevant to our work, and indeed our model is inspired by his work. In order to model recombination we need to consider a finite population with variation, whereas in Valiant's model all members of the population were considered identical. Our model is inspired by models from population genetics, particularly the Wright-Fisher model [5], [19]. As in the Wright-Fisher model we have discrete generations and a fixed-size population in each generation. The members of each generation choose their parents randomly from the previous generation[1]. However, unlike the models in population genetics, the individuals in our population are representations of boolean functions. These functions may be modified through mutation and recombination. The goal, as in Valiant's model, is to evolve a representation that predicts an *unknown* ideal function (almost) accurately using

[1]However, a generation in our model does not necessarily refer to one biological generation. See Section VI

feedback obtained through (natural) selection. We show that recombination does indeed speed-up evolution and in fact an exponential speed-up can be achieved for evolving certain concept classes, in terms of the number of generations. We show this via a reduction from parallel learning algorithms using techniques introduced by Feldman [2]. The reductions in the paper require *initialization*, i.e. we are allowed to decide the starting population.

Section II describes our extension to Valiant's evolution model to explicitly include recombination. Section III describes certain models of statistical query learning and Section IV contains the reduction from parallel CSQ learning to evolvability. In section V we give parallel algorithms for certain learning problems, thus showing their faster evolvability with recombination. Section VI concludes with some discussion and directions for future work.

II. Evolution Model

We extend Valiant's model of evolution [1] to model explicitly the process of recombination. The environment is thought of as an instance space X and let D be a distribution over X. A concept class C over X is a set of boolean functions, $f : X \to \{-1, 1\}$. We assume that the *representation size* of the instance space X is captured by a parameter n. It is common in computational learning theory to define instance space as $\cup_{n\geq 1} X_n$, however, to simplify presentation we only use X and keep the size parameter n implicit. The notion of efficient computation requires resources to be bounded by a polynomial in n, as well as parameters ϵ^{-1} (accuracy) and δ^{-1} (confidence). Suppose that $f \in C$ is the ideal or target function. In the context of evolution, we can think of an ideal function as one that indicates whether or not a given protein should be synthesized in a given environment. The environment may be defined by several input parameters such as temperature, humidity, presence of oxygen etc.

A *representation class* R is a set of representations, such that every $r \in R$ is a (possibly randomized) boolean function, $r : X \to \{-1, +1\}$. In this paper, we use *zero-one* loss as in Valiant's original model, however our reductions could easily generalize to different loss functions (cf. [2], [3]). Throughout this paper, we abuse notation in the case when r is a randomized boolean function and use $r(x)$ to also denote $\mathbb{E}[r(x)]$, where the expectation is over the internal randomization of r. Thus, r can be viewed as a real-valued function with range $[-1, 1]$. The performance of a representation r with respect to target function f and distribution D is defined as,

$$\mathrm{Perf}_{f,D}(r) = \mathbb{E}_{x\sim D}[f(x) \cdot r(x)] \in [-1, 1].$$

Our model has two main components: 1) A *recombinator* operator that takes as input two representations and outputs a list of possible descendants. 2) *Selection* that takes place based on empirical performance of variants created by the recombinator.

A. Recombinator

Recombination is a process that takes two genotypes and produces new ones combining the two. The exact process of recombination in biology is not completely understood, and modeling such a process seems inherently problematic. Following Valiant's idea of defining mutations as the output of a randomized polytime Turing machine, we similarly define a recombinator as a randomized polytime Turing machine that takes as input two representations $r_1, r_2 \in R$, the error parameter ϵ and outputs a list of possible descendant genotypes $\mathrm{Desc}(r_1, r_2, \epsilon)$ of r_1 and r_2. Mutations are the sole way to add *new* variation in the process of evolution, recombination may only explore existing variation in different ways. In the case of sexual reproduction and recombination, only mutations that occur on the germ line, i.e. those that would be passed down to descendants matter. In our model the recombinator operator is allowed the power of an efficient Turing machine, and is assumed to model both mutation and recombination, as it is impossible to separate the two from the point of view of selection in a sexually reproducing population.

Definition 1 (Recombinator)**.** *For polynomial $p(\cdot, \cdot)$, a p-bounded* recombinator *is a randomized Turing machine that takes as input two representations $r_1, r_2 \in R$ and ϵ and outputs a multi-set of representations $\mathrm{Desc}(r_1, r_2, \epsilon) \subseteq R$. The running time of the Turing machine is bounded by $p(n, \epsilon^{-1})$ and $|\mathrm{Desc}(r_1, r_2, \epsilon)| \leq p(n, \epsilon^{-1})$. $\mathrm{Desc}(r_1, r_2, \epsilon)$ is allowed to be empty which is interpreted as r_1 and r_2 being unable to mate.*

The function $\mathrm{Desc}(r_1, r_2, \epsilon)$ is similar to the neighborhood function in Valiant's model. There are two main differences: First, we define $\mathrm{Desc}(r_1, r_2, \epsilon)$ as a multi-set; in Valiant's model along with the neighborhood, a probability measure is defined to specify the probability of generation of each mutation. The set $\mathrm{Desc}(r_1, r_2, \epsilon)$ is allowed to be empty; this reflects the fact that certain pairs of genomes do not allow for recombination.

B. Selection

In order to define selection we need to define variants that are feasible (viable) for survival. We assume that we have oracle access to a noisy performance function, $\tau\text{-}\widehat{\mathrm{Perf}}_{f,D}(\cdot)$, that takes as input a representation r, and outputs $\tau\text{-}\widehat{\mathrm{Perf}}_{f,D}(r)$ that satisfies $|\tau\text{-}\widehat{\mathrm{Perf}}_{f,D}(r) - \mathrm{Perf}_{f,D}(r)| \leq \tau$. This is similar to the statistical query oracle of Kearns [8]. We require that τ^{-1} is bounded by a polynomial in n, ϵ^{-1}. Such a noisy $\tau\text{-}\widehat{\mathrm{Perf}}_{f,D}(\cdot)$ oracle can easily be simulated by drawing a (reasonably large) sample and returning the empirical estimate.

In order to define feasible variants (from the descendants of r_1 and r_2), we need to define a tolerance function $t(r_1, r_2, \epsilon)$.

Definition 2. *(Tolerance Function) A (valid) tolerance function t is a function that takes as input two representations $r_1, r_2 \in R$, and accuracy parameter ϵ and outputs $t(r_1, r_2, \epsilon) \in [0, 1]$ that is bounded above and below by two polynomially-related polynomials, i.e. there exist polynomials $tl(\cdot,\cdot), tu(\cdot,\cdot)$ such that for every $r_1, r_2 \in R$ and ϵ, $1/tu(n, \epsilon^{-1}) \leq t(r_1, r_2, \epsilon) \leq 1/tl(n, \epsilon^{-1})$ and that there exists a constant a such that $tu(n, \epsilon^{-1}) \leq (tl(n, \epsilon^{-1}))^a$. Furthermore, t can be computed in polynomial time.*

The output of a recombinator is a multi-set of representations that are variants of the starting representations. We define the multi-set of *feasible variants* denoted by Feas, using the noisy τ-$\widehat{\mathrm{Perf}}_{f,D}(\cdot)$ performance function and a tolerance function $t(r_1, r_2, \epsilon)$. Feasibility is defined as having performance at most $t(r_1, r_2, \epsilon)$ lower than one of the starting representations. Thus, for two representations r_1, r_2, define $\mathrm{Feas}_{t,\tau}(r_1, r_2, \epsilon)$ as a multi-set of feasible representations that is a sub-multi-set of $\mathrm{Desc}(r_1, r_2, \epsilon)$.

$$\mathrm{Feas}_{t,\tau}(r_1, r_2, \epsilon) = \{r \in \mathrm{Desc}(r_1, r_2, \epsilon) | \tau\text{-}\widehat{\mathrm{Perf}}_{f,D}(r) \geq v^* - t(r_1, r_2, \epsilon)\}$$
$$\text{where } v^* = \min(\tau\text{-}\widehat{\mathrm{Perf}}_{f,D}(r_1), \tau\text{-}\widehat{\mathrm{Perf}}_{f,D}(r_2)).$$

The feasible variants are those whose performance is not much worse than that of the representations that produced them. We believe this is a reasonable definition of feasibility since if the original representations were able to survive in the existing environment, variants whose performance (fitness) is not noticeably lower should be able to survive as well. It is possible for different members of the population to have different fitness levels at any given generation.

Valiant's original model did not require a *diverse* population. Indeed in an evolutionary algorithm in the sense of Valiant [1], at each generation only a single genotype was preserved and changes across generations were caused solely due to mutation. For recombination to influence evolution in any way, variation in population at each generation is a must. In this paper we assume the existence of a finite population and if its size is bounded by a polynomial it is considered *reasonable*.

Define a *population* to be a subset of R, the set of representations. An evolutionary step takes a population P_0 to population P_1 at the next generation. This transition involves action of the recombinator on existing representations of P_0, and then selection of (random) feasible variants. The population P_1 is produced as follows. Each member of P_1 is picked by picking parents r_1, r_2 randomly from P_0 (with replacement). We consider the feasible variants $\mathrm{Feas}_{t,\tau}(r_1, r_2, \epsilon)$ of the multi-set $\mathrm{Desc}(r_1, r_2, \epsilon)$ and choose r randomly from $\mathrm{Feas}_{t,\tau}(r_1, r_2, \epsilon)$ if it is non-empty. This process is repeated until $|P_1| = |P_0|$. Formally,

Definition 3. *(Evolutionary Step) An evolutionary step takes as input a starting population $P_0 \subseteq R$, and using a recombinator that defines $\mathrm{Desc}(r_1, r_2, \epsilon)$ for every $r_1, r_2 \in R$, a tolerance function t and a performance oracle τ-$\widehat{\mathrm{Perf}}_{f,D}$ outputs a population P_1 as follows:*

Let $P_1 = \emptyset$.
Do while $|P_1| < |P_0|$
1. *Select randomly (with replacement) $r_1, r_2 \in P_0$.*
2. *Consider the descendants $\mathrm{Desc}(r_1, r_2, \epsilon)$.*
3. *Construct the feasible variants $\mathrm{Feas}_{t,\tau}(r_1, r_2, \epsilon)$ with respect to tolerance function t and performance oracle τ-$\widehat{\mathrm{Perf}}_{f,D}$.*
4. *If $\mathrm{Feas}_t(r_1, r_2, \epsilon) \neq \emptyset$, pick r randomly from $\mathrm{Feas}_t(r_1, r_2, \epsilon)$ and put r in P_1.*

Observe that as long as the population size $|P_0|$ is polynomially bounded, an evolutionary step can be simulated in polynomial time. An *evolutionary algorithm* consists of the following components: a representation class R, a recombinator operator Desc, a tolerance function t, a performance oracle τ-$\widehat{\mathrm{Perf}}$, and is defined as a sequence of evolutionary steps that starting with population P_0, successively produces the populations $P_0, P_1, P_2 \ldots$. Formally,

Definition 4. *(Evolutionary Algorithm) For a polynomial $p(\cdot,\cdot)$, a p-bounded evolutionary algorithm consists of a representation class R, a recombinator operator* Desc*, a tolerance function t and has access to a performance oracle τ-$\widehat{\mathrm{Perf}}_{f,D}$. An evolutionary algorithm starting with population P_0 is a sequence of evolutionary steps, that successively produce populations $P_0, P_1, P_2, \ldots$. It is required that the recombinator* Desc *is p-bounded, $|P_0| \leq p(n, \epsilon^{-1})$ and $\tau^{-1} \leq p(n, \epsilon^{-1})$.*

Finally, we define the notion of evolvability with recombination in g generations. We say that evolvability with recombination requires *initialization* if the evolutionary algorithm succeeds when starting from a particular population P_0.

Definition 5. *We say that a concept class C is evolvable with recombination with respect to distribution D over X in g generations, if for some polynomial $p(n, \epsilon^{-1})$ there exists a p-bounded evolutionary algorithm, that for every $\epsilon > 0$, from any starting population P_0 and for every target function $f \in C$, with probability at least $1 - \epsilon$ for some $t < g$ reaches a population P_t containing a member $r \in P_t$ such that $\mathrm{Perf}_{f,D}(r) \geq 1 - \epsilon$.*

Readers familiar with the Wright-Fisher model from population genetics will notice the similarity between their model and ours. However while population genetics is concerned mainly with allele frequency distributions, the focus of our work is on understanding the computational

limits of evolution. A natural question that arises is, does augmenting Valiant's model with recombination increase the power of evolvability? In the sense of polynomially bounded evolvability the answer is negative. This follows from Feldman's [2] result showing equivalence between evolvability and correlational statistical query learning, and this observation was also implicit in Valiant's paper (cf. Section 6 [1]). However, we show that evolvability with recombination may be exponentially faster for certain interesting concept classes under restricted distributions; this follows via a reduction from parallel learning algorithms to evolvability with recombination. Our reduction requires *initialization*. In the next section, we formally define a model for parallel correlational statistical query learning and then show in Section IV that parallel correlational statistical query algorithms can be simulated using evolutionary algorithms with recombination. The algorithms presented in this paper may appear somewhat unnatural, but the goal is not to model exact physical processes but to understand their computational limitations. Our model can be understood as specifying the outer limits of the evolutionary process, as we do not expect nature to perform computations not captured by efficient Turing machines.

III. Statistical Query Learning Models

The *statistical query* (SQ) model was introduced by Kearns [8] and has been an important tool in studying noise-tolerant learning algorithms. As in the PAC model, the goal of a learning algorithm is to output a hypothesis h that has low error with respect to the target function f under a fixed but unknown distribution D. The learning algorithm has access to a *statistical query oracle* but not random labeled examples. The learner is allowed to make queries of the form (ψ, τ) where $\psi : X \times \{-1, 1\} \to [-1, 1]$ is a polynomially evaluatable query function and τ is a tolerance parameter. The oracle may respond with any value v that satisfies $|\mathbb{E}_{x\sim D}[\psi(x, f(x))] - v| \leq \tau$. An SQ algorithm is efficient if it runs in polynomial (in n, ϵ^{-1}) time and furthermore τ^{-1} is bounded by a polynomial in n, ϵ^{-1} and outputs a hypothesis h such that $\mathrm{err}_D(h, f) = \Pr_{x\sim D}[h(x) \neq f(x)] \leq \epsilon$.

A *correlational statistical query* (CSQ) (cf. [2], [9]) is a query ϕ where $\phi : X \to [-1, 1]$ is polynomially evaluatable. A τ-CSQ oracle on receiving query ϕ returns a value v such that $|\mathbb{E}_{x\sim D}[\phi(x)f(x)] - v| \leq \tau$. Thus the τ-CSQ oracle returns the (noisy) correlation between a function ϕ and the target f, or alternatively the (noisy) performance of a function ϕ. We also require that τ^{-1} is polynomially bounded. Feldman [2] showed that Valiant's evolvability model is equivalent to CSQ learning. That an evolutionary algorithm in Valiant's model can be simulated using an algorithm with access to a CSQ oracle is immediate, and this is also the case when considering evolution with recombination. Thus Feldman's result shows that CSQ learning is equivalent to polynomially-bounded evolution (with or without recombination). However, recombination does in fact allow an exponential speed-up in terms of the number of generations required to evolve to a representation (hypothesis) close to the target. This follows using a general reduction from parallel CSQ learning to evolvability with recombination. Next, we define a model for parallel CSQ learning.

A. Parallel Correlational Statistical Query Learning

We define a model for parallel correlational statistical-query learning with a τ-CSQ oracle. A parallel CSQ learning algorithm uses p (polynomially bounded) processors and we assume that there is a common clock which defines parallel time steps. During each parallel time step a processor can make a CSQ query, perform polynomially-bounded computation and write a message that can be read by every other processor. We assume that communication happens at the end of each parallel time step and on the clock. The CSQ oracle answers all queries in parallel. We are not concerned with the exact mechanism for message passing, for example it could be implemented using shared memory.

Definition 6 (Parallel CSQ Learning)**.** *A concept class C over an instance space X is (τ, T)-parallel* CSQ *learnable using p processors under distribution D, if there exists a parallel* CSQ *algorithm that uses p processors and for every $\epsilon > 0$ and target function $f \in C$, after at most T parallel steps and with access to a τ-*CSQ *oracle, outputs a hypothesis h such that $\mathrm{Perf}_{f,D}(h) \geq 1 - \epsilon$. Each query ϕ must be polynomially (in n, ϵ^{-1}) evaluatable and τ^{-1} must be bounded by a polynomial in n, ϵ^{-1}. Each parallel step must be completed by each processor in polynomial time.*

To simplify the actual reduction, we define a slightly weaker model of correlational statistical query learning (ZCSQ). A τ-ZCSQ oracle (with τ^{-1} poly-bounded) on receiving query ϕ, where ϕ is a polynomially evaluatable function, returns 1 if $\mathrm{Perf}_{f,D}(\phi) \geq \tau$, returns 0 if $\mathrm{Perf}_{f,D}(\phi) \leq -\tau$ and otherwise returns either 0 or 1. We first show that access to a ZCSQ oracle is not restrictive in the sense that we can simulate a CSQ algorithm using access to ZCSQ oracle. The notion of a ZCSQ oracle is implicit in Feldman's work [3] (Section 6) and the proof of the following theorem follows from his work. A complete proof is provided in the full version of this paper.

Theorem 2. *For some polynomial $d(n, \epsilon^{-1})$, every $(8\tau, T)$-parallel* CSQ *learning algorithm using p (polynomial in n, ϵ^{-1}) processors, can be simulated by a (τ', T')-parallel* ZCSQ *algorithm with $p(d(n, \epsilon^{-1}))^4/\tau^2$ processors with $T' = T\cdot(\log(1/8\tau)+1)+2$ parallel steps, $(\tau')^{-1}$ is bounded by a polynomial in n, ϵ^{-1}.*

IV. Reduction to Evolution

In this section, we prove our main result that fast parallel CSQ learning implies fast evolution in the presence of

recombination. We use the main idea from Feldman's reduction [2], [3] that shows that correlational statistical queries can be simulated by evolution. The statement (restatement of Theorem 1) of our main theorem is:

Theorem 3. *Suppose concept class C is (τ, T) parallel-*ZCSQ *learnable using p processors, then C is evolvable with recombination starting with an initialized population P_0 within polynomially bounded resources in $O(T\log(\tau^{-1})(\log(p)+\log(n/\epsilon)))$ generations.*

The proof of Theorem 3 follows immediately from Theorem 2 and Proposition 1 stated below.

Proposition 1. *Suppose concept class C is (τ, T)-parallel* ZCSQ *learnable using p processors, then C is evolvable with recombination starting with an initialized population P_0 within polynomially bounded resources resources in at most $T(\log(p)+2)+1$ generations.*

We first describe a high-level outline of the proof strategy and then provide more details. Suppose $\mathcal{A}$ is a (τ, T)-parallel ZCSQ algorithm for learning C. We define a representation class R that contains representations encoding the state of a simulation of $\mathcal{A}$. We assume that at the end of each parallel step, all the processors have the same "state information" denoted by some string z. This is without loss of generality, since the processors are allowed any polynomially bounded communication. However, each processor has a unique identity i that differentiates its actions from those of other processors. We start with a population P_0 in which every member is identical with zero fitness. This is achieved by a representation r^0, which is a randomized representation where $\mathsf{r}^0(x) = 1$ or -1 with equal probability. We show that each parallel-step of the algorithm is simulated using $\log(p)+2$ evolutionary steps. We refer to the simulation of the k^{th} parallel step as phase k. The outline of the simulation is as follows:

1) At the start of phase k the population is identical, r^{k-1}. Each phase begins with a differentiation step, at the end of which each member of the population has an identity of the processor it will simulate, and there are roughly equal number of members simulating each processor. This is achieved by defining $\mathrm{Desc}(\mathsf{r}^{k-1}, \mathsf{r}^{k-1}, \epsilon) = \{\mathsf{r}^{i,k-1} | i = 1, \cdots, p\}$, where each $\mathsf{r}^{i,k-1}$ is functionally identical to r^{k-1} but encodes the identity of processor i that it will simulate.
2) After differentiation each individual simulates the processor it has chosen by using the recombinator (essentially just a mutation in this case) to simulate the ZCSQ query that the processor makes during that parallel step. This step is similar to Feldman's reduction that shows that ZCSQ queries can be effectively simulated by a single mutation step. Furthermore, the descendants that survive also encode the message that is to be passed to all other processors in the communication step.
3) The communication (message passing) in each parallel step is simulated using $\log(p)$ evolutionary steps. At the end of these the population once again becomes identical, r^k, and phase k is over.

At the end of phase T, each representation r^T encodes a state that represents a valid simulation of the parallel ZCSQ algorithm $\mathcal{A}$, and thus can produce a hypothesis h that has high performance. This can be achieved in one evolutionary step.

Now, we describe the main details of the construction of the representation class that we use and define Desc and the tolerance function. A more detailed proof is provided in the full version of this paper.

We specify exactly what r^{k-1} is later, but for now assume there is some representation r^{k-1} and we define $\mathrm{Desc}(\mathsf{r}^{k-1}, \mathsf{r}^{k-1}, \epsilon)$. Recall that $\mathsf{r}^{i,k-1}$ is a representation that is functionally identical to r^{k-1}, but encodes the additional information that it should take the role of processor i of the parallel ZCSQ algorithm $\mathcal{A}$. Define,

$$\mathrm{Desc}(\mathsf{r}^{k-1}, \mathsf{r}^{k-1}, \epsilon) = \{\mathsf{r}^{i,k-1} \mid 1 \le i \le p\}$$

Thus, at the end of the first evolutionary step of phase k, the population consists of members of the form $\mathsf{r}^{i,k-1}$ and there roughly equal number of members for each value of i. In the next step each of these representations simulate the ZCSQ query of the corresponding processor. Let $\phi_z^{i,k-1}$ be the query made by the processor i during parallel step k, where z captures the state of the simulation so far, and let $\sigma_b^{i,k}$ be the message it wishes to pass, which may depend on the query response b. Then define,

$$\mathsf{r}^{i,k-1}_{b,\sigma_b^{i,k-1}} = \mathsf{r}^{i,k-1} + \frac{1}{N}(-1)^{b+1}\phi_z^{i,k-1}$$

$\mathsf{r}^{i,k-1}_{b,\sigma_b^{i,k-1}}$ is defined for $b = 0, 1$, which are the possible query responses. N is some sufficiently large (but polynomially bounded) number that we define later, whose role is only to keep all representations bounded between $[-1, 1]$. In Claim 1, we show that $\mathsf{r}^{i,k-1}_{b,\sigma_b^{i,k-1}}$ is feasible (as a descendant of $\mathsf{r}^{i,k-1}$ and $\mathsf{r}^{i,k-1}$) if and only if b is a valid τ-ZCSQ oracle response to the query $\phi_z^{i,k-1}$. Also $\sigma_b^{i,k-1}$ is the message that processor i wishes to broadcast at that step. Claim 1 proves that after two evolutionary steps a population starting from identical members r^{k-1} is transformed into a population containing only members of the form $\mathsf{r}^{i,k-1}_{b,\sigma_b^{i,k-1}}$, each member representing a valid simulation of the τ-ZCSQ query made by the corresponding processor i. Furthermore, there are sufficiently many member of each type. Formally define,

$$\mathrm{Desc}(\mathsf{r}^{i,k-1}, \mathsf{r}^{i,k-1}, \epsilon) = \{\mathsf{r}^{i,k-1}_{b,\sigma_b^{i,k-1}} \mid b = 0, 1\}$$

$$\mathrm{Desc}(\mathsf{r}^{i,k-1}, \mathsf{r}^{i',k-1}, \epsilon) = \emptyset, \text{ for } i \neq i'$$

Finally, we also need to specify the tolerance function. Define $t(r_1, r_2, \epsilon) = \tau/2N$, whenever r_1 and r_2 are both one of

$\mathsf{r}^{k-1}, \mathsf{r}^{1,k-1}, \ldots, \mathsf{r}^{p,k-1}$. Since τ^{-1} and N are polynomially bounded, this is a valid tolerance function.

In the proofs in this section, we use the following version of the Chernoff-Hoeffding bound: If X_i are independent identically distributed with mean μ, such that $X_i \in [0,1]$, then

$$\Pr\left[\neg\left(\frac{\mu}{1+\delta} \leq \frac{1}{m}\sum_{i=1}^{m} X_i \leq \mu(1+\delta)\right)\right] \leq 2e^{-\delta^2\mu m/12}$$

Claim 1. *Suppose that we begin with a population P in which each member is identical, r^{k-1}. Suppose $\alpha = 1/p$, then for every $0 \leq \delta \leq 2^{1/4}-1$, after 2 evolutionary steps we get a population in which the following hold with probability at least $1 - 4pe^{-\delta^2\alpha|P|/24}$:*

1) *Each member of the population is of the form $r^{i,k-1}_{b,\sigma_b^{i,k-1}}$ where b is a valid query response to the query $\phi_z^{i,k-1}$ made by processor i during the k^{th} parallel step and $\sigma_b^{i,k-1}$ is the message the processor i wishes to broadcast.*
2) *If f_i is the fraction of the population of type $r^{i,k-1}_{b,\sigma_b^{i,k-1}}$, then*
$$\frac{\alpha}{(1+\delta)^5} \leq f_i \leq (1+\delta)^5\alpha.$$

Proof: Let P be the starting population; each member of P is r^{k-1} and $\mathrm{Desc}(\mathsf{r}^{k-1}, \mathsf{r}^{k-1}, \epsilon) = \{\mathsf{r}^{i,k-1} \mid 1 \leq i \leq p\}$. Then, imagine choosing the population for the next generation one member at a time (as in Def. 3); each time the probability of choosing $\mathsf{r}^{i,k-1}$ is exactly $1/p$ for every i. Thus after one generation the population will be differentiated and the expected number of individuals $\mathsf{r}^{i,k-1}$ is $\alpha|P| = |P|/p$. Thus except with probability $2e^{-\delta^2\alpha|P|/12}$, the fraction of $\mathsf{r}^{i,k-1}$, say $\tilde{f}_i$, satisfies $\alpha/(1+\delta) \leq \tilde{f}_i \leq (1+\delta)\alpha$. Taking union bound, this holds for all i with probability at least $1 - 2pe^{-\delta^2\alpha|P|/12}$.

Thus after one evolutionary step, each member of the population is of the form $\mathsf{r}^{i,k-1}$, and assume that $\tilde{f}_i$ satisfies $\alpha/(1+\delta) \leq \tilde{f}_i \leq \alpha(1+\delta)$ (allowing the simulation to fail with probability $2pe^{-\delta^2\alpha|P|/12}$). Next we show that $\mathsf{r}^{i,k-1}_{b,\sigma_b^{i,k-1}}$ is a feasible descendant of $\mathsf{r}^{i,k-1}$ and $\mathsf{r}^{i,k-1}$ if and only if b is a valid response of a τ-ZCSQ oracle to the query $\phi_z^{i,k-1}$. The evolutionary algorithm uses a $(\tau/4N)$-$\widehat{\mathrm{Perf}}_{f,D}$ oracle ($4N/\tau$ is polynomially bounded). Suppose $b = 1$ and $\mathsf{r}^{i,k-1}_{1,\sigma_1^{i,k-1}}$ is feasible. Then,

$$\frac{\tau}{4N}\text{-}\widehat{\mathrm{Perf}}_{f,D}(\mathsf{r}^{i,k-1}_{1,\sigma_1^{i,k-1}}) \geq \frac{\tau}{4N}\text{-}\widehat{\mathrm{Perf}}_{f,D}(\mathsf{r}^{i,k-1}) - \frac{\tau}{2N}$$

Because a $\tau/4N$-$\widehat{\mathrm{Perf}}_{f,D}$ oracle returns the value of $\mathrm{Perf}_{f,D}$ accurately up to an additive factor $\tau/4N$, this implies that,

$$\begin{aligned}\mathrm{Perf}_{f,D}(\mathsf{r}^{i,k-1}_{1,\sigma_1^{i,k-1}}) - \mathrm{Perf}_{f,D}(\mathsf{r}^{i,k-1}) &\geq -\frac{\tau}{N}\\ \frac{1}{N}\mathbb{E}_{x\sim D}[\phi_z^{i,k-1}(x)f(x)] &\geq -\frac{\tau}{N}\\ \mathbb{E}_{x\sim D}[\phi_z^{i,k-1}(x)f(x)] &\geq -\tau\end{aligned}$$

Here f is the ideal (target) function. But the last statement implies that $b = 1$ is a valid response to the query $\phi_z^{i,k-1}$ by a τ-ZCSQ oracle. The case when $b = 0$ is a feasible variant is similar.

The statement about the fraction of representations of each type now follows immediately using the Chernoff-Hoeffding bound, since the expected number of representations of the form $\mathsf{r}^{i,k-1}_{b,\sigma_b^{i,k-1}}$ (given $\tilde{f}_{i'}$ for all i') is $(\tilde{f}_i^2/\sum_{i'}\tilde{f}_{i'}^2)|P|$. Note that $\alpha/(1+\delta)^4 \leq \tilde{f}_i^2/\sum_{i'}\tilde{f}_{i'}^2 \leq \alpha(1+\delta)^4$. Since $\tilde{f}_i^2/\sum_{i'}\tilde{f}_{i'}^2 \geq \alpha/2$ for the value of δ in the statement of the Claim, except with probability $2pe^{-\delta^2\alpha|P|/24}$, for all i, it holds that $\alpha/(1+\delta)^5 \leq f_i \leq \alpha(1+\delta)^5$. (Note that when both $b = 0$ and $b = 1$ are valid answer to a query the total fraction of the population with either of these responses is still that indicated in the statement of the claim.) The assertion then follows by taking a union bound over failure events of the two evolutionary steps. ∎

Thus at the end of 2 evolutionary steps in phase k, we have a population P such that every member of the population is of the form $\mathsf{r}^{i,k-1}_{b,\sigma_b^{i,k-1}}$ where b represents a valid response to the query made by processor i during parallel step k. We now define the action of a recombinator[2] on these new representations. We need to define a few more intermediate representations. Let $\psi_0^{i,k-1} \equiv r^{i,k-1}_{b,\sigma_b^{i,k-1}}$. Then, for $t = 1, \ldots, \log(p)$ and for $1 \leq i \leq p/2^t$ define $\psi_t^{i,k-1} = \psi_{t-1}^{2i-1,k-1} + \psi_{t-1}^{2i,k-1} - \mathsf{r}^{k-1}$. We now define the recombinator operator on these variants.

1) For representations $\psi_t^{2i-1,k-1}$, $\psi_t^{2i,k-1}$, for $t = 0, \ldots \log(p) - 1$, $\mathrm{Desc}(\psi_t^{2i-1,k-1}, \psi_t^{2i,k-1}, \epsilon) = \{\psi_{t+1}^{i,k-1}\}$. Here, we assume that message passing also occurs so that $\psi_{t+1}^{i,k-1}$ captures the combined state of both its parents.
2) For any other pairs of representations r_1, r_2, $\mathrm{Desc}(r_1, r_2, \epsilon) = \emptyset$.

Finally, we define tolerance function $t(r_1, r_2, \epsilon) = 2\tau pT/N$, if r_1, r_2 are of the form $\psi_{t'}^{i',k-1}$ for some i' and t'. We choose N to be large enough so that $t(r_1, r_2, \epsilon)$ is a valid tolerance function (i.e. $t(r_1, r_2, \epsilon)$ is sandwiched between inverses of polynomially related polynomials). Also let $\mathsf{r}^k \equiv \psi_{\log(p)}^{i,k-1}$, then notice that

$$\mathsf{r}^k = \frac{1}{N}\sum_{\substack{1\leq i\leq p\\ 1\leq j\leq k}} s_{z^{j-1}}^{i,j-1}\phi_{z^{j-1}}^{i,j-1}$$

where z^{j-1} is the state of the parallel algorithm after completion of $j-1$ parallel steps, and $\phi_{z^{j-1}}^{i,j-1}$ is the query made by processor i during parallel step j and $s_{z^{j-1}}^{i,j-1}$ is 1 if the query response encoded is 1 and -1 if the query response

[2]Note that in addition to making queries and passing message, processors in a parallel algorithm also perform polynomially bounded computation. This is simulated by the recombinator because the representation encodes the state of all processors.

encoded is 0. Thus r^k is a linear combination of all queries made by all parallel processors in k parallel steps (and in addition also includes query responses and state information obtained by message passing).

Claim 2. *Suppose we start with a population that satisfies the assertion of Claim 1. Suppose $\alpha = 1/p$, then for every $0 \leq \delta \leq \ln(2)/(5p^3)$, after $\log(p)$ recombination steps, with probability at least $1 - 2p\log(p)e^{-\delta^2\alpha|P|/24}$ the population becomes identically r^k.*

Remark 1: There is a subtle point that needs to be made here. For some of the queries both $b = 0$ and $b = 1$ are valid responses. Thus technically the population at the end of claim 2 may not be identical. However, it is the case that each member of the population represents a valid simulation of k parallel steps of the parallel ZCSQ algorithm $\mathcal{A}$, which is all that is needed for our purposes.

Remark 2: With regard to the above remark, note that because it is not the case that population r^{k-1} is exactly identical, but only that each member represents a valid simulation, while defining recombination we need to be more careful. We have defined $\psi_{j+1}^{i,k-1} = \psi_j^{2i-1,k-1} + \psi_j^{2i,k-1} - \mathsf{r}^{k-1}$. In the case that the representations $\psi_j^{2i-1,k-1}$ and $\psi_j^{2i,k-1}$ contain possibly different versions of r^{k-1}, both valid simulations, then any one of the may be removed to keep the representation bounded in $[-1, 1]$. For the reduction to hold, all we need is that the state (including query responses) of any valid simulation of the the parallel-ZCSQ algorithm $\mathcal{A}$ up to step $k-1$ is carried forward.

Proof of Claim 2: The main idea is that we combine information of two processors at a time and thus after $\log(p)$ steps, there are individuals who have information of simulations of all processors and the communication is complete. For simplicity we assume that p is a power of 2, so that $\log(p)$ is an integer.

We claim that after j evolutionary steps, with probability at least $1 - 2jpe^{-\delta^2\alpha|P|/12}$ the following hold:

1) Each member of the population is of the form $\psi_j^{i,k-1}$, $i = 1, \ldots, 2^{\log(p)-j}$
2) If f_i^j is the fraction of the population of the form $\psi_j^{i,k-1}$ with then $2^j\alpha/(1+\delta)^{a_j} \leq f_i^j \leq 2^j\alpha(1+\delta)^{a_j}$.

where a_j is defined by the recurrence, $a_0 = 5$, $a_t = 4a_{t-1} + 1$. We show this assertion by induction. It clearly holds when $j = 0$ (because it is the assertion of Claim 1). Suppose it holds for j, then consider the corresponding assertion for $j+1$. Consider an individual $\psi_{j+1}^{i,k-1}$, that can be produced by the action of the recombinator on $\psi_j^{2i-1,k-1}$ and $\psi_j^{2i,k-1}$. The number of $(\psi_j^{2i-1,k}, \psi_j^{2i,k})$ pairs is $f_{2i-1}^j f_{2i}^j |P|^2$ and we know that,

$$\frac{(2^j\alpha|P|)^2}{(1+\delta)^{2a_j}} \leq f_{2i-1}^j f_{2i}^j |P|^2 \leq (2^j\alpha|P|)^2(1+\delta)^{2a_j}$$

On the other hand the total number of possible feasible mating pairs is $\sum_{l=1}^{2^{\log(p)-(j+1)}} f_{2l-1}^j f_{2l}^j |P|^2$, which satisfies,

$$\frac{p}{2^{j+1}}\frac{(2^j\alpha|P|)^2}{(1+\delta)^{2a_j}} \leq \sum_{l=1}^{2^{\log(p)-(j+1)}} f_{2l-1}^j f_{2l}^j |P|^2$$

and

$$\sum_{l=1}^{2^{\log(p)-(j+1)}} f_{2l-1}^j f_{2l}^j |P|^2 \leq \frac{p}{2^{j+1}}(2^j\alpha|P|)^2(1+\delta)^{2a_j}$$

Thus the probability for an individual in the $(j+1)^{\text{th}}$ generation, that the parents from the j^{th} generation are $\psi_j^{2i-1,k}$ and $\psi_j^{2i,k}$ is at least $2^{j+1}\alpha/(1+\delta)^{4a_j}$ and at most $2^{j+1}\alpha(1+\delta)^{4a_j}$. Thus by the Chernoff-Hoeffding bound, except with a further probability loss of $2pe^{-\delta^2 f_i^{j+1}|P|/12}$, we have $2^{j+1}\alpha/(1+\delta)^{4a_j+1} \leq f_i^{j+1} \leq 2^{j+1}\alpha(1+\delta)^{4a_j+1}$.

Notice that $a_t \leq 5^{t+1}$ for all t, and hence for the value of δ in the statement of the claim, the values f_i^j for all i, j are at least $\alpha/2$. Thus the statement holds by induction. However, after $\log(p)$ steps there is only one variant in the population $\psi_{\log(p)}^{1,k-1} \equiv \mathsf{r}^k$.

It still remains to be proved that $\psi_{j+1}^{i,k-1}$ is a *feasible* descendant of $\psi_j^{2i-1,k-1}$ and $\psi_j^{2i,k-1}$. Note that since all query responses are valid (by assertion of Claim 1), the term $(-1)^{b+1}\phi_z^{i,k-1}$ satisfies $\mathbb{E}[(-1)^{b+1}\phi_z^{i,k-1} f] \geq -\tau$, where f is the target function. There can be at most pT total queries and this can reduce the total performance by at most $pT\tau/N$ for any representation $\psi_{j+1}^{i,k-1}$. Thus $\mathrm{Perf}(\psi_{j+1}^{i,k-1}) \geq \mathrm{Perf}(\psi_j^{2i-1,k-1}) - (pT\tau)/N$. Thus when $t(r_1, r_2, \epsilon) = 2pT\tau/N$ and by using a $\tau/4N$-$\widehat{\mathrm{Perf}}$ oracle the descendant $\psi_{j+1}^{i,k-1}$ of $\psi_j^{2i-1,k}$ and $\psi_j^{2i,k}$ is feasible. ■

Using Claims 1 and 2, we can now prove Proposition 1.

Proof (of Proposition 1): We start with a population in which each member is identical, r^0. Each parallel step of the parallel ZCSQ learning algorithm $\mathcal{A}$ can be simulated using $\log(p) + 2$ evolutionary steps. Thus a (τ, T)-parallel ZCSQ learning algorithm can be simulated using an evolutionary algorithm in $T(\log(p) + 2)$ generations, at the end of which all representations know the end state of a valid simulation of $\mathcal{A}$, that can then output a good hypothesis h ($\mathrm{Perf}_{f,D}(h) \geq 1 - \epsilon$).

In the simulation, any representation that we use is an average of at most Tp (randomized) boolean functions. Thus setting $N = pT$ ensures that all the representations are valid (randomized) boolean functions. But if necessary we may set N to be larger (but polynomially bounded) to make the tolerance functions valid (sandwiched by polynomially related polynomials e.g. $N = pT\tau n/\epsilon$). Finally we need to add one more evolutionary step. For r^T, define $\mathrm{Desc}(\mathsf{r}^T, \mathsf{r}^T, \epsilon) = \{\mathsf{r}^T, h_{\mathsf{r}^T}, \ldots, h_{\mathsf{r}^T}\}$, where h_{r^T} is the output of the parallel ZCSQ algorithm and occurs M times in the multi-set $\mathrm{Desc}(\mathsf{r}^T, \mathsf{r}^T, \epsilon)$. Note that if $\mathrm{Perf}_{f,D}(\mathsf{r}^T) \geq 1 - \epsilon + \tau/(2N)$, then r^T itself is a good hypothesis. Otherwise the representation h_{r^T} is a feasible variant. (We can set

$\epsilon = \epsilon/2$ and $\tau/N < \epsilon/2$, to ensure that $h_{\mathbf{r}T}$ is a feasible variant.) We set M large enough so that $h_{\mathbf{r}T}$ is chosen with high probability. In order to ensure that the total failure probability is not more than ϵ, we set $|P| = O^*(p^7 T)$ where $O^*(\cdot)$ hides logarithmic factors. ■

Remark 3: We have only defined Desc for representations that the evolutionary algorithm that simulates the parallel ZCSQ algorithm $\mathcal{A}$ reaches with high probability. However, the definition needs to be made for all representations. For the remaining states, this may be made arbitrarily.

Remark 4: The representations we have defined implicitly depend on ϵ. To define a representation class independent of ϵ, we can define the representations for $\epsilon = 1, 1/2, 1/4, \ldots, 1/2^n$ and start from a special representation $\perp$. The first evolutionary step takes the population to a representation encoding the correct ϵ (cf. [2], [4]).

V. Applications

In this section we show that our general reduction from Section IV can be applied to the question of evolvability of conjunctions and halfspaces under certain distributions.

A. Evolving Conjunctions

Valiant showed that the class of monotone conjunctions is evolvable in $O(n \log(n/\epsilon))$ generations [1] under the uniform distribution. Diochnos and Turán [20] showed that Valiant's algorithm [1] can be improved and in fact showed that monotone conjunctions under the uniform distribution can be evolved in $O(\log(1/\epsilon))$ steps. This holds even in Valiant's model, i.e. without recombination. Their method however uses properties unique to the uniform distribution and monotone conjunctions and it is not clear whether this can be generalized to general conjunctions or other distributions. Feldman's reduction from CSQ learning [2] implies that conjunctions can be evolved in $O^*(n)$ generations in Valiant's model with respect to any fixed distribution.

Our reduction shows that under evolution with recombination, the class of conjunctions can be evolved in $O((\log(n)/\epsilon)^2)$ generations. This follows from a simple constant-time parallel CSQ algorithm for learning conjunctions. This algorithm is non-uniform, i.e. it requires advice capturing certain information about the distribution (cf. [2]). Details are provided in the full version of this paper.

B. Evolving Halfspaces

Kanade, Valiant and Vaughan gave an algorithm for learning homogeneous halfspaces under radially symmetric distributions and product normal distributions. Their evolution algorithm requires $O(n/\epsilon)$ (in fact $O(n/\epsilon^6)$ in the case of product normal distributions) generations. Our reduction shows that under evolution with recombination, homogeneous halfspaces under radially symmetric distribution and product normal distributions can be evolved in $O((\log(n/\epsilon))^2)$ generations. Details are provided in the full version of this paper.

VI. Discussion and Future Work

Our work suggests that recombination can potentially parallelize and hence speed-up evolution in Valiant's evolution model. Our work should be understood as demonstrating that for certain definitions of selection and recombination, polylogarithmic generations are enough for evolution whenever a suitable parallel CSQ algorithm exists. Here, we discuss in some detail the choices made in our model and also open questions.

In our model we have defined a recombinator operator, that simultaneously captures both mutation and recombination. In a sexually reproducing species only mutations that occur on the germ line matter and natural selection never acts on mutations alone, but simultaneously also on recombination. We have defined *feasible*[3] variants as those whose performance is at most some inverse polynomial lower than a variant that produced it. This definition is different from those common in population genetics where survival probability is determined by relative fitness, thus capturing competition between members of the species. However, the mapping from performance (in the sense of agreement with a target function) may translate to survival probability in several different ways. Although sharp thresholds are somewhat unnatural, they are robust to any monotone transformation between performance and survival probability. We believe that our choice of definition of feasible variants makes sense when the population is large and distributed over a large area and selection is weak and roughly equal for all beneficial mutations. However, if selection is much stronger on some mutations than others, our definition of feasibility may be unrealistic because weakly beneficial mutations may not stand any chance of survival. Similar assumptions are often used in population genetics models in the context of conditions under which sexual reproduction accelerates evolution (cf. [7], [16]). Maynard Smith [7] has a detailed discussion on this topic and our model is a plausible one in the conditions described therein.

One *generation* according to our model may correspond to several biological generations. The number of biological generations equivalent to one generation in our model (or Valiant's) can be thought of as the number of generations required for a beneficial mutation (or a particular recombination) to be established in the population. Thus the speed-up proposed by our model over Valiant's only makes sense when the rate of beneficial mutations is small compared to the time required for a beneficial mutation to be established in the population. This does seem to be the case in nature.

We believe that our reduction will generalize to different performance measures such as those used by Feldman [3] in

[3]Unlike Valiant, we do not define beneficial and neutral mutations separately. This is mainly for two reasons: first to keep the model relatively simple, and second, since our results are most relevant when selection is weak it makes less sense to differentiate explicitly between beneficial and neutral mutations.

a straightforward manner. A question that remains open is whether evolution under recombination would be equivalent if the tolerance function were required to be *fixed*. Currently, our model allows tolerance functions to depend on the representations (similar to Valiant [1]). In our reduction, we use different values of tolerance depending on the representations concerned. We suspect that this is an artifact of our reduction and in fact evolution with recombination would remain equivalent with fixed tolerance. Another interesting question is to find a "natural" evolutionary algorithm using recombination for some simple concept class, such as conjunctions. Such algorithms may shed more light on the advantage of recombination.

Finally, there remains the question of showing lower bounds on the number of generations required for evolution in Valiant's model. Feldman [2] showed that the correlational query complexity of a class is an upper bound. However, since the simulation of an evolutionary algorithm in Valiant's model can make several queries in each generation, a lower bound that separates Valiant's model from ours, in terms of number of generations required for evolution, would likely require different techniques.

Acknowledgments

I benefited greatly from several discussions with Leslie Valiant. I thank Vitaly Feldman, Adam Kalai and Leslie Valiant for their comments on an earlier version of the manuscript. I am grateful to Dipti Nayak and Gireeja Ranade for pointing me to useful references in the biology literature. This work was supported in part by NSF grants CCF-04-27129 and CCF-09-64401.

References

[1] L. G. Valiant, "Evolvability," *Journal of the ACM*, vol. 56, no. 1, pp. 1–21, 2009, (Earlier version appeared in the Proceedings of the 32nd International Symposium on Mathematical Foundations of Computer Science, LNCS, Vol 4708, (2007) Springer-Verlag, 22-43.).

[2] V. Feldman, "Evolvability from learning algorithms," in *Proceedings of ACM STOC 40*. New York, NY, USA: ACM, 2008, pp. 619–628.

[3] ——, "Robustness of evolvability," in *Conference on Learning Theory*, 2009.

[4] V. Kanade, L. G. Valiant, and J. W. Vaughan, "Evolution with drifting targets," in *Conference on Learning Theory*, 2010.

[5] R. A. Fisher, *The genetical theory of natural selection*. Clarendon Press, 1930.

[6] H. J. Muller, "Some genetic aspects of sex," *The American Naturalist*, vol. 66, no. 703, pp. 118–138, 1932.

[7] J. Maynard Smith, "What use is sex?" *Journal of Theoretical Biology*, vol. 30, no. 2, pp. 319–335, 1971.

[8] M. Kearns, "Efficient noise-tolerant learning from statistical queries," *JACM*, vol. 45, no. 6, pp. 983–1006, 1998.

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

[10] V. Feldman, "Distribution-independent evolvability of linear threshold functions," 2011, conference on Learning Theory.

[11] L. Michael, "Evolvability via the Fourier transform," *Theoretical Computer Science*, 2010, to appear.

[12] V. Feldman, "A complete characterization of statistical query learning with applications to evolvability," in *Proceedings of the IEEE Symposium on Foundation of Computer Science*. Washington, DC, USA: IEEE Computer Science, 2009, pp. 375–384.

[13] P. Valiant, "Distribution free evolvability of real functions over all convex loss functions," 2011, eCCC-TR-11-089.

[14] A. Livnat, C. Papadimitriou, J. Dushoff, and M. W. Feldman, "A mixability theory for the role of sex in evolution," *PNAS*, vol. 105, no. 50, pp. 19 803–19 808, 2008.

[15] A. Livnat, C. Papadimitriou, N. Pippenger, and M. W. Feldman, "Sex, mixability, and modularity," *PNAS*, vol. 107, no. 4, pp. 1452–1457, 2010.

[16] J. Maynard Smith, *The Evolution of Sex*. Cambridge: Cambridge University Press, 1978.

[17] ——, "The evolution of recombination," *The evolution of sex: an examination of current ideas (edited by R. E. Michod and B. R. Levin)*, pp. 106–125, 1988.

[18] N. H. Barton and B. Charlesworth, "Why sex and recombination?" *Science*, vol. 281, pp. 1986–1990, 1998.

[19] S. Wright, "Evolution in mendelian populations," *Genetics*, vol. 16, pp. 97–159, 1931.

[20] D. I. Diochnos and G. Turán, "On evolvability: The swapping algorithm, product distributions, and covariance," in *Fifth Symposium on Stochastic Algorithms, Foundations and Applications*. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 74–88.