Journal of Machine Learning Research 8 (2007) 1835-1865 Submitted 10/06; Revised 4/07; Published 8/07

Behavioral Shaping for Geometric Concepts

Manu Chhabra MCHHABRA@CS.ROCHESTER.EDU
Department of Computer Science
University of Rochester
Rochester, NY 14627, USA

Robert A. Jacobs ROBBIE@BCS.ROCHESTER.EDU
Department of Brain and Cognitive Sciences
University of Rochester
Rochester, NY 14627, USA

Daniel Štefankovič STEFANKO@CS.ROCHESTER.EDU
Department of Computer Science
University of Rochester
Rochester, NY 14627, USA

Editor: Manfred K. Warmuth

Abstract

In a search problem, an agent uses the membership oracle of a target concept to find a positive example of the concept. In a *shaped search problem* the agent is aided by a sequence of increasingly restrictive concepts leading to the target concept (analogous to behavioral shaping). The concepts are given by membership oracles, and the agent has to find a positive example of the target concept while minimizing the total number of oracle queries. We show that for the concept class of intervals on the real line an agent using a bounded number of queries *per oracle* exists. In contrast, for the concept class of unions of two intervals on the real line no agent with a bounded number of queries per oracle exists. We then investigate the (amortized) number of queries per oracle needed for the shaped search problem over other concept classes. We explore the following methods to obtain efficient agents. For axis-parallel rectangles we use a bootstrapping technique to obtain gradually better approximations of the target concept. We show that given rectangles $R \subseteq A \subseteq \mathbb{R}^d$ one can obtain a rectangle $A' \supseteq R$ with $\mathrm{vol}(A')/\mathrm{vol}(R) \leq 2$, using only $O(d \cdot \mathrm{vol}(A)/\mathrm{vol}(R))$ random samples from A. For ellipsoids of bounded eccentricity in $\mathbb{R}^d$ we analyze a deterministic ray-shooting process which uses a sequence of rays to get close to the centroid. Finally, we use algorithms for generating random points in convex bodies (Dyer et al., 1991; Kannan et al., 1997) to give a randomized agent for the concept class of convex bodies. In the final section, we explore connections between our bootstrapping method and active learning. Specifically, we use the bootstrapping technique for axis-parallel rectangles to active learn axis-parallel rectangles under the uniform distribution in $O(d\ln(1/\varepsilon))$ labeled samples.

Keywords: computational learning theory, behavioral shaping, active learning

1. Introduction

Computational models of learning are often inspired by human cognitive phenomena. For example, the PAC model of Valiant (1984) is a model of our ability to learn concepts from positive and negative examples generated at random by the world. Human learning, however, demonstrates a richer variety of learning phenomena such as that of behavioral shaping. Behavioral shaping is a training procedure commonly used to teach complex behaviors. Using this procedure, a complex task is taught to a learner in an incremental manner. The learner is initially rewarded for performing an easy task that coarsely resembles the target task that the teacher wants the learner to perform. Over time, the learner is rewarded for performing more difficult tasks that monotonically provide better approximations to the target task. At the end of the training sequence, the learner is rewarded only for performing the target task. Shaping was first proposed by B. F. Skinner in the 1930s (Skinner, 1938). In addition to training humans and other organisms, behavioral shaping has also been used in artificial intelligence applications such as those that arise in robotics or reinforcement learning (Dorigo and Colombetti, 1994; Mataric, 1994; Randløv and Alstrøm, 1998; Konidaris and Barto, 2006; Ng et al., 1999).

The goal of this paper is to mathematically formalize the notion of shaping, and to show that shaping makes certain tasks easier to learn. We specifically study shaping in the context of search problems (for a learning theoretic analysis of a similar search problem, see, e.g., Fine and Mansour, 2006). In a search problem, the task is to find one positive example of a concept given a membership oracle of the concept using as few queries to the oracle as possible. If a shaping sequence, which is a sequence of nested concepts, is available, it might be possible to solve the search problem with a smaller number of oracle queries. When concepts are standard geometrical concepts like rectangles, balls, ellipsoids, and convex bodies in high dimensions, we show efficient algorithms to solve the search problem using a shaping sequence.

"Reward shaping" of Ng et al. (1999) and quasi-convex optimization are related to our work. Ng et al. (1999) gave conditions under which "reward shaping" works in a reinforcement learning setting. In their framework, shaping is a transformation of the reward function and the goal is to formalize conditions under which this transformation preserves value of the underlying policies. Our framework is different from Ng et al. (1999) in at least two ways. First, we have a sequence of reward functions as compared to their one step reward transform. Second, our reward functions are binary whereas they allow general, real valued rewards.

A weaker version of the shaped search problem in which the concepts are convex and all the oracles are available to the agent simultaneously can be viewed as an instance of a quasi-convex optimization problem. Also, we cannot apply the algorithms from this area since they usually rely on an oracle (so called separation oracle) stronger than the membership oracle that we use. What makes our setting different is that the oracles are available in a fixed order, and we only have membership oracles. Our framework is motivated by behavioral shaping (Skinner, 1938), as well as practical robotics (Dorigo and Colombetti, 1994).

Our work is similar in spirit to the idea of using a helpful teacher (Goldman and Kearns, 1991; Goldman et al., 1993; Goldman and Mathias, 1993; Hegedűs, 1994). For example, Goldman and Kearns (1991) considered a model where a helpful teacher chose examples to allow a learner to uniquely identify any concept in a given concept class. Our model differs from Goldman and Kearns (1991) in the following aspects. First, the teacher in Goldman and Kearns (1991) is "active" (it is directly providing the learner with "good" examples), whereas in our model the burden of choosing good queries is on the learner. Second, the learner in Goldman and Kearns (1991) is faced with a more general task of identifying the concept, whereas our learner is solving a search problem.

The rest of the paper is organized as follows. In Section 2 we define the shaped search problem and summarize the results of the paper. In Section 3 we illustrate the shaped search problem with algorithms for the concept classes of intervals and axis-parallel rectangles. In Section 4 we use a bootstrapping algorithm to give an improved algorithm for axis-parallel rectangles. Section 5 explains and analyzes the center-point algorithm which is used in Section 6 to solve the shaped search problem for bounded eccentricity ellipsoids. Section 7 uses the techniques on sampling random points from a convex body to solve the problem for general convex bodies. In Section 8 we define the problem of one-sided active learning, and show that the bootstrapping algorithm given in Section 4 can be used to active-learn the concept class of axis-parallel rectangles with $O(d\ln(1/\varepsilon))$ labeled samples. Finally, Section 9 wraps up the paper with a discussion and possible future directions.

2. A Formal Model of Behavioral Shaping

A *search problem* is defined by a pair of sets (S,T) such that $T \subseteq S$ (S is the starting set and T is the target set). A search agent "knows" S and has to find a point $y \in T$. The set T will be given by a membership oracle. The goal of the agent is to minimize the number of queries to the oracle.

Of course without any conditions on (S,T) the agent could be searching for a needle in a haystack and require an unbounded number of queries. To make the problem reasonable we will assume that S and T come from some concept class $\mathcal{C}$ (e. g., S,T could be intervals in $\mathbb{R}$), and that the volume $\mathrm{vol}(T)$ is not too small compared to the volume $\mathrm{vol}(S)$ (i. e., T is not a needle in a haystack S).

Before formally defining a search problem we need the following standard notions from learning theory (see, e.g., Anthony and Biggs, 1992; Kearns and Vazirani, 1994). Let $(X,\mathcal{B})$ be a measurable space and let vol be a measure on $(X,\mathcal{B})$. Let $\mathcal{C} \subseteq \mathcal{B}$. The set $\mathcal{C}$ is called a *concept class* and its members are called *concepts*. We will assume that $\mathcal{C}$ comes with a *representation scheme* (see Kearns and Vazirani, 1994, Chapter 2). Examples of concept classes that we study include intervals in $\mathbb{R}$, axis-parallel rectangles in $\mathbb{R}^d$, ellipsoids in $\mathbb{R}^d$, and convex bodies in $\mathbb{R}^d$. We will restrict our attention to the Lebesgue measure on $\mathbb{R}^d$.

Definition 2.1 Let $\mathcal{C}$ be a concept class. A *search problem* is defined by a pair of concepts (S,T) such that $T \subseteq S$ and $S,T \in \mathcal{C}$. The agent has a representation of S and has to find a point in T using a membership oracle for T.

Note that for any concept class there is a natural "oblivious" randomized algorithm for the search problem: query independent uniform random points from S until you find a point in T. The expected number of queries of the algorithm is $\mathrm{vol}(S)/\mathrm{vol}(T)$. For sufficiently complicated concept classes (e. g., finite unions of intervals) the use of randomness might be inevitable—a deterministic algorithm with bounded number of queries need not exist (the question of deterministic search is related to the concept of hitting sets, see, e.g., Linial et al., 1993).

For concept classes we will consider one can find $\Omega(n)$ disjoint concepts, each of volume $\Omega(1/n)$. The following observation implies that the trivial algorithm is the best possible (up to a constant factor).

Observation 2.1 *Suppose that there exist disjoint concepts $T_1,\dots,T_n \subseteq S$. Let i be uniformly random from $[n]$. The expected (over the choice of i) number of queries made by any (randomized) agent on (S,T_i) is $\Omega(n)$.*

In a *shaped search problem* the agent's search task will be aided by a *shaping sequence*, which is a sequence of nested sets between the S and T. The sets in the shaping sequence will be gradually shrinking concepts from the underlying concept class $\mathcal{C}$. The rate of shrinking will be controlled by a parameter, denoted γ.

Definition 2.2 Let $\gamma \in (0,1)$. Let $\mathcal{C}$ be a concept class and let (S,T) be a search problem over $\mathcal{C}$. A sequence of concepts $S_0 \supseteq S_1 \supseteq \cdots \supseteq S_k$ is called a γ-*shaping sequence* if $S_0 = S$, $S_k = T$, and $\mathrm{vol}(S_{t+1}) \geq \gamma \mathrm{vol}(S_t)$ for all $t \in \{0,\dots,k-1\}$.

A search agent in a shaped search problem only has access to the membership oracles of $S_1,\dots,S_k$. However, if the agent makes a query to S_t, it can no longer make a query to S_j with $j < t$. In other words, the oracles S_t are presented to the agent in k iterations, with the agent making (zero or more) queries to the oracle S_t at iteration t. The agent successfully solves the shaped search problem if at the end of the process it outputs $x \in T$. We assume that the agent knows the value of γ and does not know the value of k. However, before the last iteration the agent is informed that it is accessing the last oracle.

We will evaluate the performance of the agent by the amortized number of membership queries per oracle (i. e., the total number of queries divided by k). We will also consider randomized agents, which are zero-error probability (Las Vegas) algorithms (i. e., they are always successful). For a randomized agent the performance is measured by the expected number of membership queries per oracle, where the expectation is taken over the coin flips of the agent. This is formalized in the following definition.

Definition 2.3 Let $\mathcal{C}$ be a concept class. Let $\mathcal{A}$ be a search agent. We say that the agent $\mathcal{A}$ solves a γ-shaped search problem using *q queries per oracle*, if for every $S,T \in \mathcal{C}$, every k, and every γ-shaping sequence $S_0,\dots,S_k \in \mathcal{C}$ the total number of queries made by the agent is bounded by kq. If the agent is randomized we require the expected total number of queries to be bounded by kq.

Note that for $\gamma > \gamma'$ any γ-shaping sequence is a γ'-shaping sequence. Thus as $\gamma \to 1$ the shaped search problem becomes easier. We will study how γ affects the complexity of the shaped search problem.

2.1 Our Results

In order to introduce the shaped search problem, we start with a positive and a negative result for two simple concept classes (the proofs are in Section 3). First, we show that $O(1/\gamma)$ queries per oracle suffice to solve the γ-shaped search problem for the concept class of closed intervals in $\mathbb{R}$.

Proposition 2.4 *Let $\mathcal{C}$ be the concept class of closed intervals in $\mathbb{R}$. There exists a deterministic agent which for any γ uses $O(1/\gamma)$ queries per oracle to solve γ-shaped search problem.*

Next, we contrast the Proposition 2.4 by showing that for the class of "unions of two closed intervals in $\mathbb{R}$" there exists no agent that solves the γ-shaped search problem using bounded number of queries per oracle.

Proposition 2.5 *Let $\mathcal{C}$ be the concept class of unions of two closed intervals in $\mathbb{R}$. Let $\gamma \in (0,1)$. For every (randomized) agent $\mathcal{A}$ and every number q there exists a search problem (S,T), k, and a γ-shaping sequence $S_1,\dots,S_k$ such that $\mathcal{A}$ makes more than q queries per oracle (in expectation).*

We understand the concept class of intervals completely as Proposition 2.4 can be strengthened as follows.

Proposition 2.6 *Let $\mathcal{C}$ be the concept class of closed intervals in $\mathbb{R}$. Let $f(\gamma) = 1/\gamma$ for $\gamma \leq 1/2$, and $f(\gamma) = \ln(1/\gamma)$ for $\gamma > 1/2$. There exists a deterministic agent which for any $\gamma \in (0,1)$ uses $O(f(\gamma))$ queries per oracle to solve γ-shaped search problem. On the other hand, for any $\gamma \in (0,1)$, any (randomized) agent makes at least $\Omega(f(\gamma))$ queries per oracle.*

The shaped search problem for axis-parallel rectangles in $\mathbb{R}^d$ turns out to be more complicated. Here we do not understand the dependence of the complexity of the γ-shaped search problem on γ. We present three algorithms; each algorithm works better than the other two for a certain range of γ.

We say that a randomized agent is *oblivious* if for every oracle S_t the queries to S_t which lie in S_t are independent and uniformly random in S_t.

Theorem 2.7 *Let $\mathcal{C}$ be the concept class of axis-parallel rectangles in $\mathbb{R}^d$.*

1. *For any γ there exists a randomized agent using $O(\frac{1}{\gamma} + (d + \ln \frac{1}{\gamma}) \ln d)$ queries per oracle.*
2. *For any γ there exists an (oblivious) randomized agent using $O(d/\gamma)$ queries per oracle.*
3. *For any constant $\gamma > 1/2$ there exists a deterministic agent using $O(\ln d)$ queries per oracle.*

The following table compares the number of queries used by the algorithms for various values of γ.

	Alg. 1.	Alg. 2.	Alg. 3.
$\gamma = 3/4$	$O(d \ln d)$	$O(d)$	$O(\ln d)$
$\gamma = 1/4$	$O(d \ln d)$	$O(d)$	N/A
$\gamma = 1/\ln d$	$O(d \ln d)$	$O(d \ln d)$	N/A
$\gamma = 1/d$	$O(d \ln d)$	$O(d^2)$	N/A

Note that the deterministic algorithm for part 3. uses less queries than the randomized algorithm for part 2., but it only works in a very restricted range of γ. It relies on the following fact: the centroid of an axis-parallel rectangle of volume 1 is contained in every axis-parallel sub-rectangle of volume $\geq 1/2$. It would be interesting to know whether the logarithmic dependence on d could be extended for constants $\gamma \leq 1/2$, or, perhaps, a lower bound could be shown.

Question 1 *Are $\Omega(d)$ queries per oracle needed for $\gamma < 1/2$?*

The simple agent for the part 1) of Theorem 2.7 is described in Section 3.

In Section 4 we introduce the concept of "bootstrap-learning algorithm". A bootstrap-learning algorithm, given an approximation A_1 of an unknown concept $C \in \mathcal{C}$ and a membership oracle for C, outputs a better approximation A_2 of C. We show an efficient bootstrap-learning algorithm for the concept class of axis-parallel rectangles and use it to prove part 2) of Theorem 2.7.

Part 3) of Theorem 2.7 is proved in Section 5. We show how an approximate center of an axis-parallel rectangle can be maintained using only $O(\ln d)$ (amortized) queries per oracle. If γ is not too small, the center of S_t will remain inside S_{t+1} and can be "recalibrated".

The results of Section 5 suggest that maintaining an approximate centroid of the S_t is a useful approach for solving the shaped search problem. For a centrally symmetric convex body K the following process can be used to get close to a centroid of K: pick a line ℓ through the current point, move the current point to the center of $\ell \cap K$ and repeat. If ℓ is uniformly random the process converges to the centroid of K. It would be interesting to know what parameters influence the convergence rate of this process.

Question 2 *How many iterations of the random ray-shooting are needed to get ε-close to the centroid of a (isotropic), centrally symmetric convex body K?*

We will consider the following deterministic version of the ray-shooting approach: shoot the rays in the axis directions $e_1,\dots,e_d$, in a round-robin fashion.

Question 3 *How many iterations of the deterministic round-robin ray-shooting are needed to get ε-close to the centroid of a (isotropic), centrally symmetric convex body K?*

In Section 6 we study a variant of the deterministic ray-shooting which moves to a weighted average of the current point and the center of $K \cap \ell$. We analyze the process for the class of ellipsoids of bounded eccentricity. As a consequence we obtain:

Theorem 2.8 *Let $\mathcal{C}$ be the concept class of ellipsoids with eccentricity bounded by L. Let $\gamma > 1/2$. The γ-shaped search problem can be solved by a deterministic agent using $O(L^2 \cdot d^{3/2} \ln d)$ queries per ray-shooting oracle (a ray-shooting oracle returns the points of intersection of K with a line through a point $x \in K$)*

The requirement $\gamma > 1/2$ in Theorem 2.8 can be relaxed. Similarly, as in the axis-parallel rectangle case, we need a condition on the volume of a sub-ellipsoid of an ellipsoid E which guarantees that the sub-ellipsoid contains the centroid of E. We do not determine this condition (which is a function of L and d).

To prove Theorem 2.8 we need the following interesting technical result.

Lemma 2.9 *Let $v_1,\dots,v_d \in \mathbb{R}^d$ be orthogonal vectors. Let $\alpha \in (0,2)$ and $L \geq 1$. Let D be an $d \times d$ diagonal matrix with diagonal entries from the interval $[1/L,1]$. Let*

$$M(\alpha) = \prod_{i=1}^{n} \left(I - \alpha \cdot \frac{D v_i v_i^{\mathrm{T}} D}{v_i^{\mathrm{T}} D^2 v_i} \right).$$

Then

$$\|M(1/\sqrt{d})\|_2^2 \leq 1 - \frac{1}{5L^2\sqrt{d}}.$$

Using random walks and approximating ellipsoids (Bertsimas and Vempala, 2004; Kannan et al., 1997; Grötschel et al., 1988; Lovász, 1986) we can show that convexity makes the shaped search problem solvable. We obtain the following result (a sketch of the proof is in Section 7):

Theorem 2.10 *Let $\mathcal{C}$ be the concept class of compact convex bodies in $\mathbb{R}^d$. For any $\gamma \in (0,1)$ there exists a randomized agent for the γ-shaped search problem using $O(\mathrm{poly}(d, 1/\gamma))$ queries per oracle.*

3. Basic Results for Intervals and Axis-parallel Rectangles

Now we show that for intervals $O(1/\gamma)$ queries per oracle are sufficient to solve the γ-shaped search problem. For each S_t we will compute an interval $[a_t,b_t]$ containing S_t such that the length of $[a_t,b_t]$ is at most three times longer than the length of S_t. By querying S_{t+1} on a uniform set of $O(1/\gamma)$ points in $[a_t,b_t]$ we will obtain $[a_{t+1},b_{t+1}]$.

Proof of Proposition 2.4:
The agent will compute an interval approximating S_t for $t=0,\dots,k$. More precisely it will compute a_t and b_t such that $S_t \subseteq [a_t,b_t]$ and $\mathrm{vol}(S_t) \geq (b_t-a_t)/3$. Initially we have $S=S_0=:[a_0,b_0]$ and $\mathrm{vol}(S_0)=(b_0-a_0)\geq(b_0-a_0)/3$.

Suppose that $S_t \subseteq [a_t,b_t]$ and $\mathrm{vol}(S_t)\geq(b_t-a_t)/3$. Using an affine transformation we can, w.l.o.g., assume $a_t=0$ and $b_t=1$. Thus $\mathrm{vol}(S_t)\geq 1/3$ and $\mathrm{vol}(S_{t+1})\geq\gamma/3$.

Let $Q_0=\{0,1\}$, $Q_1=\{0,1/2,1\}$, …, $Q_i=\{j/2^i \mid j=0,\dots,2^i\}$. The agent will query S_{t+1} on all points in Q_i, $i=0,1,\dots$, until it finds the smallest j such that $|Q_j\cap S_{t+1}|\geq 2$. Choose a_{t+1} and b_{t+1} such that

$$Q_j\cap S_{t+1}=\{a_{t+1}+2^{-j},\dots,b_{t+1}-2^{-j}\}.$$

For this a_{t+1},b_{t+1} we have $S_{t+1}\subseteq[a_{t+1},b_{t+1}]$ and $\mathrm{vol}(S_{t+1})\geq(b_{t+1}-a_{t+1})/3$.

Note that if $2^{-i}\leq\gamma/6$ then $|A_i\cap S_{t+1}|\geq 2$. By the minimality of j we have $2^j\leq 12/\gamma$ and hence the total number of queries per oracle is $O(1/\gamma)$. ■

Proof of Proposition 2.6:
First we show the upper bound of $O(\ln(1/\gamma))$ for $\gamma>1/2$. Let $\ell=\lceil -\frac{\ln 2}{\ln\gamma}\rceil$. Note that $\gamma^\ell\geq 1/4$. Now we use the agent from Proposition 2.4 on oracles $S_0,S_\ell,S_{2\ell},\dots$, and we do not query the rest of the oracles at all. The number of queries per oracle is $O(1/\ell)=O(\ln(1/\gamma))$.

Next we show the lower bound of $\Omega(1/\gamma)$ for $\gamma<1/2$. We take a shaping sequence of length $k=1$. Note that there exist $\lfloor 1/\gamma\rfloor$ disjoint intervals of length γ in $[0,1]$ and hence, by Observation 2.1, the agent needs to make $\Omega(\lfloor 1/\gamma\rfloor)$ queries (per oracle).

Finally, the lower bound of $\Omega(\ln(1/\gamma))$ will follow by an information-theoretic argument. Assume that the agent is deterministic. Fix an integer k. There exist $\Omega(1/\gamma^k)$ disjoint intervals of length γ^k. For each of these intervals there exists a shaping sequence of length k ending with that interval. We will randomly pick one of these shaping sequences and present it to the agent. The agent, using Q queries, identifies which interval (out of the $\Omega(1/\gamma^k)$ intervals) we chose. This implies $E[Q]=\Omega(k\ln(1/\gamma))$, and hence the number of queries per oracle is $\Omega(\ln(1/\gamma))$. The lower bound for a randomized agent follows by Yao's min-max lemma (see, e.g., Motwani and Raghavan, 1995, Chapter 2). ■

For unions of two intervals the agent can be forced to make many queries per oracle. If one of the intervals is large and one is small then the small interval can abruptly shrink. We will use this to "hide" the target T. Then we will shrink the large interval until it disappears. Now the agent is forced to find the hidden target T, which requires a large number of queries.

Proof of Proposition 2.5:
Let $\gamma\in(0,1)$. Let n be the smallest positive integer such that $(1+\gamma^n)\gamma\leq 1$. Let ℓ be a positive integer such that $\gamma^\ell<1/2$. Let T be a random interval of length $\gamma^{n+\ell}$ in $[0,\gamma^n]$. Let $S=[-1,1]$. The

γ-shaping sequence will be the following:

$$S_t = \begin{cases} [-1,0] \cup [0,\gamma^t] & \text{for } t = 0,\dots,n, \\ [-\gamma^{t-n-1},0] \cup T & \text{for } t = n+1,\dots,3n+\ell+1, \\ T & \text{for } t = 3n+\ell+2. \end{cases}$$

Note that S_t is always a union of two intervals. In the first $n+1$ iterations, the right hand-side interval is shrinking. When the right-hand side interval is sufficiently small we can replace it by the "hidden" interval T. After that we shrink the left-hand side until we make it disappear.

For the sake of the lower bound argument, we will help the agent by telling it the general shape of the shaping sequence, but we will keep the location of T secret. Now, w.l.o.g, we can assume that the agent only queries points in $[0,\gamma^n]$ on the oracle for T (because for all the other queries the agent can figure the answer himself). By Observation 2.1 the agent needs $\Omega(1/\gamma^\ell)$ queries to find a point in T. Hence the number of queries per oracle is

$$\Omega\left(\frac{1/\gamma^\ell}{3n+\ell+2}\right).$$

Letting $\ell \to \infty$ we obtain that the number of queries per oracle is unbounded. ■

Now we describe an agent for axis-parallel rectangles. Let A_t be a tight rectangle containing S_t (i. e., a rectangle such that $\mathrm{vol}(A_t) \le C \cdot \mathrm{vol}(S_t)$ for some constant C). We will sample random points in A_t until we get a point y inside S_{t+1}. Then we will shoot rays from y in the axis-parallel directions to find the approximate boundary of S_{t+1}. From this we will obtain a tight rectangle A_{t+1} containing S_{t+1}.

Proof of the part 1) of Theorem 2.7:
We will compute axis-parallel rectangles $A_0,\dots,A_k$ such that $S_t \subseteq A_t$ and $\mathrm{vol}(A_t) \le \mathrm{e}\,\mathrm{vol}(S_t)$ (for $t = 0,\dots,k$). Clearly we can take $A_0 = S_0$.

Suppose that we have A_t such that $S_t \subseteq A_t$, and $\mathrm{vol}(A_t) \le \mathrm{e}\,\mathrm{vol}(S_t)$. Using an affine transformation we can, w.l.o.g, assume $A_t = [0,1]^d$. Since the S_t form a γ-shaping sequence we have $\mathrm{vol}(S_{t+1}) \ge \gamma/\mathrm{e}$. We will sample random points from A_t, until we get a point x inside S_{t+1}. In expectation we will need to query at most e/γ points to find x.

Now that we have x we will try to approximate S_{t+1} in each dimension separately. We will find the smallest $j \ge 0$ such that $x + 2^{-j}e_1 \in S_{t+1}$, or $x - 2^{-j}e_1 \in S_{t+1}$. Only $O(-\ln w_1(S_{t+1}))$ queries are needed for this step (where $w_1(S_{t+1})$ is the width of S_{t+1} in the 1-st dimension).

Then using binary search on $[0,2^{1-j}]$ we will find y such that $x + ye_1 \in S_{t+1}$ and $x + (y + 2^{-j-1}/d)e_1 \notin S_{t+1}$. This step uses only $O(\ln d)$ queries. Similarly we will find $z \in [0,2^{1-j}]$ such that $x - ze_1 \in S_{t+1}$ and $x - (z + 2^{-j-1}/d)e_1 \notin S_{t+1}$. Note that

$$I_1 := [x - (z + 2^{-j-1}/d), x + (y + 2^{-j-1}/d)],$$

contains the projection of S_{t+1} into the 1-st dimension, and the the length of I_1 is at most $(1 + 1/d)w_1(S)$.

Analogously we compute the I_i for $i = 1,\dots,d$. The total number of queries is

$$O(d\ln d) + O\left(-\sum_{i=1}^{d} \ln w_1(S_{t+1})\right) = O\left(d\ln d + \ln\frac{1}{\gamma}\right).$$

Let $A_{t+1} = I_1 \times \cdots \times I_d$. We have $S_{t+1} \subseteq A_{t+1}$, and $\mathrm{vol}(A_{t+1}) \le (1+1/d)^d \le \mathrm{e}$. ■

4. (α,β)-bootstrap Learning Algorithms

In this section we prove the part 2) of Theorem 2.7. We cast the proof in a general setting that we call "bootstrap-learning algorithms".

Definition 4.1 Let $\mathcal{C}$, $\mathcal{C}_A$ be concept classes. Let $\alpha > \beta \geq 1$. An (α,β)-*bootstrap learning algorithm for* $\mathcal{C}$ *using* $\mathcal{C}_A$ takes as an input a representation of a concept $A_1 \in \mathcal{C}_A$ and an oracle for a concept $R \in \mathcal{C}$. The concepts A_1 and R are guaranteed to satisfy $R \subseteq A_1$ and $\mathrm{vol}(A_1) \leq \alpha \cdot \mathrm{vol}(R)$. The algorithm outputs a representation of a concept $A_2 \in \mathcal{C}_A$ such that $R \subseteq A_2$ and $\mathrm{vol}(A_2) \leq \beta \cdot \mathrm{vol}(R)$. The efficiency of the algorithm is measured by the worst-case (expected) number T of oracle queries to R (i. e., we take the worst A_1 and R from $\mathcal{C}$).

If an efficient (α,β)-bootstrap learning algorithm exists for a concept class $\mathcal{C}$ then it can be used for the shaped search problem as follows.

Lemma 4.2 *Let* $\mathcal{C}$*,* $\mathcal{C}_A$ *be concept classes. Let* $\alpha > \beta \geq 1$*. Assume that there exists an* (α,β)*-bootstrap learning algorithm for* $\mathcal{C}$ *using* $\mathcal{C}_A$ *using* T *queries. Suppose that for every* $C \in \mathcal{C}$ *there exists* $A \in \mathcal{C}_A$ *such that* $C \subseteq A$ *and* $\mathrm{vol}(A) \leq \beta \cdot \mathrm{vol}(C)$*. Then there exists an agent which for any* $\gamma \geq \beta/\alpha$ *solves the* γ*-shaped search problem (over* $\mathcal{C}$*) using* T *queries per oracle.*

Proof :
We will compute $A_0,\ldots,A_k \in \mathcal{C}_A$ such that $S_t \subseteq A_t$ and $\mathrm{vol}(A_t) \leq \beta \cdot \mathrm{vol}(S_t)$. By the assumption of the lemma the first A_0 exists. (The starting concept S in a shaped search problem is known in advance and hence the agent can pre-compute A_0.)

Suppose that we have A_t. Then $S_{t+1} \subseteq A_t$, and $\mathrm{vol}(A_t) \leq (\beta/\gamma)\mathrm{vol}(S_{t+1}) \leq \alpha \mathrm{vol}(S_{t+1})$. Hence using the (α,β) boot-strap algorithm we can find A_{t+1}, using only T queries. ■

If one uses Lemma 4.2 to obtain an agent for the shaped search problem for $\mathcal{C}$, one should choose $\mathcal{C}_A$ which allows for an efficient (α,β)-bootstrap algorithm. Later in this section we will show that for axis-parallel rectangles one can take $\mathcal{C}_A = \mathcal{C}$, and obtain an efficient (α,β)-bootstrap algorithm. Are there concept classes for which it is advantageous to choose $\mathcal{C}_A \neq \mathcal{C}$? More generally one can ask:

Question 4 *For given concept class* $\mathcal{C}$*, and* $\alpha > \beta \geq 1$*, which concept classes* $\mathcal{C}_A$ *allow for an efficient* (α,β)*-bootstrap learning algorithm?*

We will study the following algorithm for the (α,β)-bootstrap learning problem.

input : a representation of $A_1 \in \mathcal{C}_A$ and an oracle for $R \in \mathcal{C}$
assume : $R \subseteq A_1$ and $\mathrm{vol}(A_1) \leq \alpha \cdot \mathrm{vol}(R)$.
output : a representation of $A_2 \in \mathcal{C}_A$, such that
$R \subseteq A_2$ and $\mathrm{vol}(A_2) \leq \beta \cdot \mathrm{vol}(R)$.

$S^+ \leftarrow \emptyset$, $S^- \leftarrow \emptyset$
repeat
  pick a random point $p \in A_1$
  if $p \in R$ **then** $S^+ \leftarrow S^+ \cup \{p\}$ **else** $S^- \leftarrow S^- \cup \{p\}$ **fi**
  $\text{Possible}_R \leftarrow \{C \in \mathcal{C} \mid S^+ \subseteq C \subseteq A_1 \setminus S^-\}$
  $v \leftarrow$ the minimal volume of a concept in Possible_R
  $A_2 \leftarrow$ a concept of minimal volume in $\mathcal{C}_A$ containing all $C \in \text{Possible}_R$
until $\mathrm{vol}(A_2) \leq \beta \cdot v$
output a representation of A_2

Algorithm 1: Inner-Outer algorithm for (α, β)-bootstrap learning

Note that the set Possible_R contains R and hence $R \subseteq A_2$, and $v \leq \mathrm{vol}(R)$. Thus when the algorithm terminates we have $\mathrm{vol}(A_2) \leq \beta \cdot v \leq \beta \cdot \mathrm{vol}(R)$. Thus we have the following observation.

Proposition 4.3 *The Inner-Outer algorithm is an (α, β)-bootstrap learning algorithm.*

Now we analyze the Inner-Outer algorithm for the concept class of axis-parallel rectangles with $\mathcal{C}_A = \mathcal{C}$. We will need the following technical results (the proofs are in the appendix).

Lemma 4.4 *Let $X_1, \dots, X_n$ be i.i.d. uniformly random in the interval $[0,1]$. Then*

$$E\left[-\ln\left(\max_i X_i - \min_i X_i\right)\right] = \frac{2n-1}{n(n-1)} \leq \frac{2}{n-1} \leq \frac{4}{n}.$$

Lemma 4.5 *Let K be from the binomial distribution $B(n,p)$. Let $X_1, \dots, X_K$ be i.i.d. uniformly random in the interval $[0,1]$. Then*

$$E[\min\{1, X_1, \dots, X_K\}] = \frac{1-(1-p)^{n+1}}{(n+1)p} \leq \frac{1}{np}.$$

Lemma 4.6 *Let $\mathcal{C}$ be the set of axis-parallel rectangles in $\mathbb{R}^d$. Let $\mathcal{C}_A = \mathcal{C}$. The expected number of oracle calls made by the Inner-Outer algorithm is bounded by $8 + 320d\alpha/\ln\beta$.*

As an immediate corollary we obtain:

Proof of the part 2) of Theorem 2.7:
Immediate from Lemma 4.6 and Lemma 4.2. ■

Proof of Lemma 4.6:
W.l.o.g. we can assume $A_1 = [0,1]^d$. Let $R = [a_1, b_1] \times \cdots \times [a_d, b_d]$, where $0 \leq a_i \leq b_i \leq 1$, for $i = 1, \dots, d$.

For the purpose of the analysis of the Inner-Outer algorithm we will split the algorithm into two phases of length n_1 and n_2, respectively. We will then show that with probability $1/4$ the algorithm stops after these two phases. From this it will follow that the expected number of samples used by the algorithm is at most $4(n_1 + n_2)$.

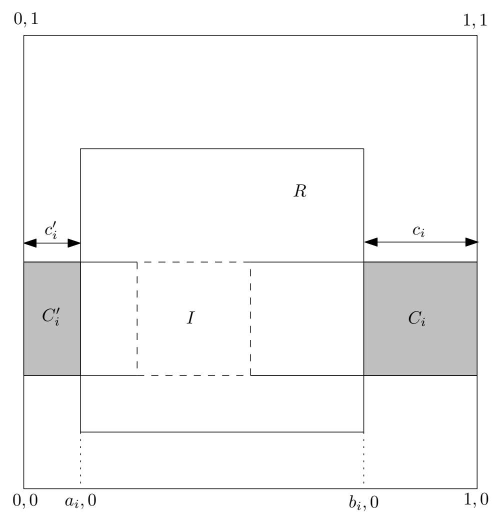

Figure 1: Schematic drawing for the proof of Lemma 4.6.

In the first phase n_1 i.i.d. random points from A_1 are sampled. The expected number of points that fall inside R is at least n_1/α. By Chernoff bound with probability at least $3/4$ we get at least $n_1/(2\alpha)$ points inside R. With probability at most $1/4$ the algorithm "fails".

From now on we assume that the algorithm did not "fail", i. e., at least $n_1/(2\alpha)$ points are inside R. Let I be the smallest rectangle containing these points. We will choose $n_1 > 4\alpha$ and hence, by Lemma 4.4, the expected logarithm of the width of I in the i-th dimension satisfies

$$E\left[-\ln\frac{w_i(I)}{b_i - a_i}\right] \leq \frac{8\alpha}{n_1}. \tag{1}$$

Summing the (1) for $i \in [d]$ we obtain (using the linearity of expectation)

$$E\left[\ln\frac{\mathrm{vol}(R)}{\mathrm{vol}(I)}\right] \leq \frac{8d\alpha}{n_1}. \tag{2}$$

Markov inequality applied to (2) yields that with probability at least $3/4$ we have

$$\ln\frac{\mathrm{vol}(R)}{\mathrm{vol}(I)} \leq \frac{32d\alpha}{n_1}. \tag{3}$$

If (3) is not satisfied we say that the algorithm "failed".

From now on we assume that (3) is true, i. e., the algorithm did not fail. Thus

$$\mathrm{vol}(I) \geq \mathrm{vol}(R)\cdot\exp\left(-\frac{32d\alpha}{n_1}\right). \tag{4}$$

Let I_i be obtained from I by stretching the i-th dimension to $[0,1]$ (for $i \in [d]$). Note that R cuts I_i into three parts. Call the parts of I_i that are outside of R, C_i and C_i' (see Figure 1). Let c_i be the width of C_i in the i-th dimension, and c_i' be the width of C_i' in the i-th dimension.

Now we sample n_2 i.i.d. random points from A_1. A point falls inside C_i with probability $\mathrm{vol}(C_i) = c_i\mathrm{vol}(I)/(b_i - a_i)$. The expected distance of the point in C_i closest to R, by Lemma 4.5, is bounded by

$$c_i \cdot \frac{1}{n_2 \cdot c_i\mathrm{vol}(I)/(b_i - a_i)} = \frac{b_i - a_i}{n_2\mathrm{vol}(I)}. \tag{5}$$

Note that A_2 determined by the closest points in the C_i and C_i' contains $\mathrm{Possible}_R$. By (5) the expected width of A_2 in the i-th dimension satisfies $E[w_i(A_2)] \le (b_i - a_i)(1 + 2/(n_2\mathrm{vol}(I)))$. By Jensen's inequality

$$E\left[\ln \frac{w_i(A_2)}{b_i - a_i}\right] \le \ln\left(1 + \frac{2}{n_2\mathrm{vol}(I)}\right) \le \frac{2}{n_2\mathrm{vol}(I)}.$$

By the linearity of expectation

$$E\left[\ln \frac{\mathrm{vol}(A_2)}{\mathrm{vol}(R)}\right] \le \frac{2d}{n_2\mathrm{vol}(I)}.$$

By Markov inequality with probability at least $3/4$ we have

$$\ln \frac{\mathrm{vol}(A_2)}{\mathrm{vol}(R)} \le \frac{8d}{n_2\mathrm{vol}(I)} \le \frac{8d\alpha}{n_2}, \tag{6}$$

and hence

$$\mathrm{vol}(R) \ge \mathrm{vol}(A_2) \cdot \exp\left(-\frac{8d\alpha}{n_2}\right). \tag{7}$$

Again, if (6) is false we say that the algorithm "failed". Note that the algorithm "failed" (in any of the three possible ways) with probability at most $3/4$. Thus with probability $1/4$ we have that (4) and (7) are true and hence

$$\mathrm{vol}(A_2) \le \mathrm{vol}(I) \cdot \exp\left(\frac{8d\alpha}{n_2} + \frac{32d\alpha}{n_1}\right).$$

For $n_1 = \lceil 64d\alpha/\ln\beta \rceil$ and $n_2 = \lceil 16d\alpha/\ln\beta \rceil$ the right hand side is bounded by β and hence the algorithm will terminate.

Note that additional points in S^+ and S^- do not "hurt" the algorithm (removal of a point cannot increase v, nor can it decrease $\mathrm{vol}(A_2)$). Thus if the algorithm does not terminate, the next run of length $n_1 + n_2$ terminates with probability $\ge 1/4$, etc. Hence in expectation at most $4(n_1 + n_2)$ oracle queries suffice. ∎

5. Center-point Algorithm

In this section we show how an approximate center-point of a rectangle can be maintained using only $O(\ln d)$ queries per oracle. As a corollary we obtain a proof of the part 3) of Theorem 2.7.

Given a vector $v \in \mathbb{R}^d$, let $P_i v$ be the projection of v to the i-th dimension (i. e., the vector obtained by zeroing out all the entries of v except the i-th coordinate). Let ∂S denote the boundary of the S.

Definition 5.1 Let S be an axis-parallel rectangle in $\mathbb{R}^d$. Let $\varepsilon \in (0,1)$. Let x be a point in S and let $v \in \mathbb{R}^d$, $v \ge 0$. Let $\alpha_i, \beta_i \ge 0$ be determined by $x + \alpha_i(P_i v) \in \partial S$ and $x - \beta_i(P_i v) \in \partial S$, for $i \in [d]$. We say that v is an ε-*approximate distance vector* for (S,x) if $1 - \varepsilon \le \alpha_i \le 1$ and $1 - \varepsilon \le \beta_i \le 1$ for $i \in [d]$. We say that x is an ε-*approximate center-point* of S if there exists an ε-approximate distance vector v for (S,x).

Note that if v is an ε-approximate distance vector for (S,x) then we have

$$\prod_{i=1}^{d}(2v_i) \geq \prod_{i=1}^{d}((\alpha_i+\beta_i)v_i) = \mathrm{vol}(S). \tag{8}$$

If $S' \subseteq S$ and the volume of S' is not much smaller than the volume of S then an approximate center-point x of S should be contained in S'. The next lemma formalizes this intuition.

Lemma 5.2 *Let $S' \subseteq S$ be axis-parallel rectangles. Assume that $\mathrm{vol}(S)/\mathrm{vol}(S') \leq 2-2\varepsilon$. Let $x \in S$ be an ε-approximate center-point of S. Then $x \in S'$. Moreover, for $\alpha_i', \beta_i' \geq 0$ determined by $x+\alpha_i'(P_i v) \in \partial S'$ and $x-\beta_i'(P_i v) \in \partial S'$ we have $\alpha_i'+\beta_i' \geq 1$, and $\alpha_i' \geq \varepsilon/2$, and $\beta_i' \geq \varepsilon/2$.*

Now we give an algorithm which "recalibrates" an ε-approximate center-point. Note that the first two steps of the algorithm rely on the fact that $x \in S'$ (which is guaranteed by Lemma 5.2).

input : $x \in S$, and an ε-approximate distance vector v for (S,x).
assume : $S' \subseteq S$ and $\mathrm{vol}(S) \leq (2-2\varepsilon)\mathrm{vol}(S')$
output : $x' \in S'$, and an ε-approximate distance vector v' for (S',x').

1 find $\delta^+ > 0$ such that $x+(\delta^+ + \varepsilon/8)v \notin S'$ and $x+\delta^+ v \in S'$
2 find $\delta^- > 0$ such that $x-(\delta^- + \varepsilon/8)v \notin S'$ and $x-\delta^- v \in S'$
3 let $\delta = \min\{\delta^+, \delta^-\}$, let $s=+1$ if $\delta = \delta+$ and $s=-1$ otherwise
4 **if** $\delta < 1-\varepsilon$ **then**
5 find $j \in [d]$ such that $x+s(\delta+\varepsilon/8)(P_j v) \notin S'$ and $x+s\delta(P_j v) \in S'$
6 find $\alpha > 0$ such that $x+(\alpha+\varepsilon/8)(P_j v) \notin S'$, and $x+\alpha(P_j v) \in S'$
7 find $\beta > 0$ such that $x-(\beta+\varepsilon/8)(P_j v) \notin S'$, and $x-\beta(P_j v) \in S'$
8 update $x_j \leftarrow x_j + v_j(\alpha-\beta)/2$
9 update $v_j \leftarrow (1+\varepsilon/4)((\alpha+\beta)/2)v_j$
10 go to step 1

11 return x, v

Algorithm 2: Center-point algorithm

Proof of Lemma 5.2:
Let v be an ε-approximate distance vector for (S,x). Suppose $x \notin S'$. Then there exists a coordinate $i \in [d]$ such that S' lies on one side of the hyperplane $\{z \in \mathbb{R}^d \,|\, z_i = x_i\}$. Thus the width of S' in the i-th dimension satisfies $w_i(S') \leq v_i$. On the other hand $w_i(S) \geq w_i(S') + (1-\varepsilon)v_i$. Hence $w_i(S)/w_i(S') \geq 2-\varepsilon$ which implies $\mathrm{vol}(S)/\mathrm{vol}(S') \geq 2-\varepsilon$, a contradiction. We proved $x \in S'$.

For any $i \in [d]$, using $\alpha_i, \beta_i \geq 1-\varepsilon$, we obtain

$$2-2\varepsilon \geq \frac{\mathrm{vol}(S)}{\mathrm{vol}(S')} \geq \frac{\alpha_i+\beta_i}{\alpha_i'+\beta_i'} \geq \frac{2-2\varepsilon}{\alpha_i'+\beta_i'},$$

and hence $\alpha_i'+\beta_i' \geq 1$.

Similarly, for any $i \in [d]$, using $\beta_i' \leq \beta_i \leq 1$, $\alpha_i' \leq \alpha_i$, we obtain

$$2-2\varepsilon \geq \frac{\mathrm{vol}(S)}{\mathrm{vol}(S')} \geq \frac{\alpha_i+\beta_i}{\alpha_i'+\beta_i'} \geq \frac{\alpha_i+\beta_i}{\alpha_i'+\beta_i} \geq \frac{\alpha_i+1}{\alpha_i'+1} \geq \frac{2-\varepsilon}{\alpha_i'+1}.$$

This implies $\alpha_i' \geq \varepsilon/2$. The proof of $\beta_i' \geq \varepsilon/2$ is the same. ■

By the assumptions on the input the α_i, and β_i for x, v, S are bounded by 1 from above. Hence the α_i and β_i for x, v, S' are bounded by 1 from above. Later we will show that during the execution of the algorithm the α_i and β_i for x, v, S' always remain bounded by 1 from above.

When $|\delta| \geq 1-\varepsilon$ on step 4 then $x+(1-\varepsilon)v \in S'$ and $x-(1-\varepsilon)v \in S'$, and hence the α_i and β_i are bounded by $1-\varepsilon$ from below. Thus the final v is ε-approximate distance vector for (S',x).

It remains to show that during the algorithm the α_i and β_i for x,v,S' always remain bounded by 1 from above. Let $x'_j = x_j + v_j(\alpha-\beta)/2$ and $v'_j = (1+\varepsilon/4)((\alpha+\beta)/2)v_j$, i. e., x'_j and v'_j are the new values assigned on lines 8 and 9. We have

$$x'_j + v'_j = x_j + v_j\left(\alpha + \frac{\varepsilon(\alpha+\beta)}{8}\right) \geq x_j + v_j(\alpha+\varepsilon/8), \tag{9}$$

and

$$x'_j + (1-\varepsilon)v'_j = x_j + v_j\left(\frac{\alpha-\beta}{2} + (1+\varepsilon/4)(1-\varepsilon)\frac{\alpha+\beta}{2}\right) \leq x_j + \alpha v_j. \tag{10}$$

From (9) and (10) and our choice of α on line 6 it follows that on line 10 the value of α_j for the new x and v satisfies $1-\varepsilon \leq \alpha_j \leq 1$. Similar argument establishes $1-\varepsilon \leq \beta_j \leq 1$. Note that

$$\frac{v'_j}{v_j} = (1+\varepsilon/4)\frac{\alpha+\beta}{2} \leq (1+\varepsilon/4)(1-\varepsilon/2) \leq 1-\varepsilon/4. \tag{11}$$

We will use (11) to bound the amortized number of steps we spend in our application of the center-point algorithm.

Proof of the part 3) of Theorem 2.7:
W.l.o.g., assume $S = S_0 = [0,1]^d$. Let $x^{(0)} = v^{(0)} = (1/2,\ldots,1/2)$. Note that $v^{(0)}$ is an ε-approximate distance vector for $(S_0, x^{(0)})$. We will use the center-point algorithm to compute $x^{(t)}, v^{(t)}$ such that $v^{(t)}$ is an ε-approximate distance vector for $(S_t, x^{(t)})$.

Before we start analyzing the algorithm let us emphasize that we defined "queries per oracle" to be an "amortized" quantity (as opposed to a "worst-case" quantity). Thus it is fine if the algorithm makes $\Theta(d)$ queries going from S_t to S_{t+1}, as long as the average number of queries per oracle is $O(\ln d)$.

Now we analyze the number of queries per oracle. Let

$$\Phi_t = \frac{\prod_{i=1}^d (2v_i^{(t)})}{\mathrm{vol}(S_t)}.$$

Note that $\Phi_0 = 1$, and, by (8), $\Phi_t \geq 1$ for every $t = 0,\ldots,k$.

From (11) it follows that every time the step on line 10 is executed, the value of Φ_t decreases by a factor of $(1-\varepsilon/4)$. The denominators can contribute a factor at most $(1/\gamma)^k$ to $\Phi_k \geq 1$. Thus the step 10 is executed at most $\ln\gamma^k/\ln(1-\varepsilon/4)$ times. Therefore the steps 1-9 are executed at most $k(1+\ln\gamma/\ln(1-\varepsilon/4))$ times. The steps $1,2,6,7$ use a binary search on $[0,1]$ with precision $\varepsilon/8$ and hence use $O(\ln 1/\varepsilon)$ queries.

Now we will argue that step 5 can be implemented using $O(\ln d)$ queries using binary search. Let $v = v_1 + v_2$, where the last $\lfloor d/2 \rfloor$ coordinates of v_1 are zero, and the first $\lceil d/2 \rceil$ coordinates of v_2 are zero. We know that $x + s\delta v_1 \in S'$ and $x + s\delta v_2 \in S'$ (we used the fact that S' is an axis-parallel rectangle). We also know that $x + s(\delta+\varepsilon/8)v_1 \notin S'$ or $x + s(\delta+\varepsilon/8)v_2 \notin S'$. If $x + s(\delta+\varepsilon/8)v_1 \notin S'$ then we proceed with binary search on v_1, otherwise we proceed with binary search on v_2.

Thus the total number of queries is

$$O\left(\left(\ln\frac{1}{\varepsilon} + \ln d\right) k \left(1 + \frac{\ln\gamma}{\ln(1-\varepsilon/4)}\right)\right) = O(k\ln d),$$

since $\gamma < 1/2$ is a constant, and we can take $\varepsilon = (1/2-\gamma)/4$. ■

6. Bounded-Eccentricity Ellipsoids

For a bounded-eccentricity ellipsoid K the following process converges to the centroid of K: pick a line ℓ through the current point, move the current point to the center of $\ell \cap K$ and repeat. We analyze the process in the case when the lines ℓ are chosen in axis-parallel directions in a round-robin fashion.

We say that an ellipsoid E has eccentricity bounded by L if the ratio of its axis is bounded by L. Let A be a positive definite matrix with eigenvalues from $[1/L^2, 1]$. Let E be the ellipsoid given by $x^{\mathrm{T}}Ax = 1$ (note that E has eccentricity at most L). If the current point is x and the chosen line is $\ell = \{x + \beta y \,|\, \beta \in \mathbb{R}\}$ then the midpoint of $E \cap \ell$ is

$$x' = \left(I - \frac{yy^{\mathrm{T}}A}{y^{\mathrm{T}}Ay}\right)x.$$

The process described above moves from x to x'. A more cautious process would move somewhere between x and x'. The point $y = (1-\alpha)x + \alpha x'$ is given by

$$y = \left(I - \alpha\frac{yy^{\mathrm{T}}A}{y^{\mathrm{T}}Ay}\right)x.$$

Thus one d-step round of the cautious process takes point x to the point $S(\alpha)x$, where

$$S(\alpha) = \prod_{i=1}^{d}\left(I - \alpha\frac{e_ie_i^{\mathrm{T}}A}{A_{ii}}\right) = A^{-1/2}\left(\prod_{i=1}^{d}\left(I - \alpha\frac{A^{1/2}e_ie_i^{\mathrm{T}}A^{1/2}}{e_i^{\mathrm{T}}Ae_i}\right)\right)A^{1/2}.$$

We will consider the following quantity as a measure of “distance” from the centroid: $\|A^{1/2}x\|_2^2$. After the move we have

$$\|A^{1/2}S(\alpha)x\|_2^2 = \left\|\left(\prod_{i=1}^{d}\left(I - \alpha\frac{A^{1/2}e_ie_i^{\mathrm{T}}A^{1/2}}{e_i^{\mathrm{T}}Ae_i}\right)\right)(A^{1/2}x)\right\|_2^2. \tag{12}$$

Let $A^{1/2} = V^{\mathrm{T}}DV$, where V is orthogonal. Note that the entries of D are between $1/L$ and 1. Let $v_i = Ve_i$. Now we can apply Lemma 2.9 on (12), and obtain that for $\alpha = 1/\sqrt{d}$ we have

$$\frac{\|A^{1/2}S(\alpha)x\|_2^2}{\|A^{1/2}x\|_2^2} \le 1 - \frac{1}{5L^2\sqrt{d}}. \tag{13}$$

Now we use (13) to prove Theorem 2.8.

Proof of Theorem 2.8:

The agent will compute a sequence of points $x_0, \ldots, x_k$ such that $x_t \in S_t$.

Suppose that we have $x_t \in S_t$. The ray-shooting process is invariant under translations and uniform stretching and hence, w.l.o.g., we can assume that the centroid of S_t is at 0, and $S_t = \{y \,|\, y^{\mathrm{T}}Ay \le 1\}$, where the eigenvalues of A are from $[1/L^2, 1]$. Let $\alpha = 1/\sqrt{d}$. From (13) it follows that if we apply the ray shooting process $5L^2\sqrt{d} \cdot c$ times we obtain a point z such that $\|A^{1/2}z\|_2^2 \le \mathrm{e}^{-c}$. We choose c so that $\mathrm{e}^{-c/2} \le (\gamma - 1/2)/2$.

Now we apply affine transformation such that S_t becomes a unit ball and z becomes a point at distance at most $\mathrm{e}^{-c/2}$ from the center of S_t. Since

$$\mathrm{vol}(S_{t+1}) \geq \gamma \mathrm{vol}(S_t) \geq \left(\frac{1}{2} + 2\mathrm{e}^{-c/2}\right) \mathrm{vol}(S_t),$$

it follows that z is inside S_{t+1}, and we can take $x_{t+1} = z$. ■

Remark 1 Somewhat surprisingly the cautious process with $\alpha = 1/\sqrt{d}$ can get closer to the centroid than the original process (i. e., the one with $\alpha = 1$), see Observation A.3.

7. General Convex Bodies

In this section we show that the shaped search problem can be solved for general convex bodies. The algorithm is a simple combination of known sophisticated algorithms (e. g., ball-walk algorithm, and shallow-cut ellipsoid algorithm).

We start with an informal description of the algorithm. The agent will keep two pieces of information:

1. a collection of independent nearly-uniform random points in S_t, and
2. a weak Löwner-John pair (E, E') of ellipsoids, $E \subseteq S_t \subseteq E'$.

The random points in S_t will be so abundant that with high probability many of them will fall inside S_{t+1}. In the unlikely event that only few (or none) of the points fall inside S_{t+1} we will use E' to generate further random points in S_{t+1} (this will be very costly but unlikely).

Then we will use the random points in S_{t+1} to find an affine transformation which brings S_{t+1} into a near-isotropic position. As a consequence we will obtain a centering of S_{t+1} and we can use the shallow-cut ellipsoid algorithm (with just a membership oracle) to find a weak Löwner-John pair of ellipsoids for S_{t+1}. Finally, we will use the ball-walk algorithm (see, e.g., Kannan et al., 1997) to generate independent nearly-uniform random points inside S_{t+1}.

We will need the following definitions and results. As usual, $B(c, r)$ denotes the ball with center c and radius r.

Algorithms which deal with convex bodies given by membership oracles often require the body to be sandwiched between balls, in the following precise sense.

Definition 7.1 A (r_1, r_2)-*centered convex set* is a convex set $K \subseteq \mathbb{R}^d$ together with a point $c \in K$ such that $B(c, r_1) \subseteq K \subseteq B(c, r_2)$.

Not every convex body can be efficiently centered (e. g., if it is thin in some direction). However when we allow affine transformations of balls (i. e., ellipsoids), every convex body can be efficiently sandwiched. We will use the following relaxed notion of sandwiching.

Definition 7.2 A pair of ellipsoids (E, E') is called a *weak Löwner-John pair* for a convex body K, if $E \subseteq K \subseteq E'$, the centers of E and E' coincide, and E is obtained from E' by shrinking by a factor of $1/((d+1)\sqrt{d})$.

The following property is useful for understanding when an efficient centering is possible.

Definition 7.3 A convex set K is in *near-isotropic position* if the eigenvalues of the covariance matrix of the uniform distribution over K are from $[1/2, 3/2]$.

Our algorithm will need random samples from the uniform distribution over a convex body K. Unfortunately, uniform distribution can be difficult to achieve. The total variation distance will be used to measure the distance from uniformity.

Definition 7.4 The total *variation distance* between distributions π and μ is

$$d_{\mathrm{TV}}(\pi,\mu) = \sup_{A \subseteq \Omega} (\pi(A) - \mu(A)).$$

We will say that a distribution μ is δ-*nearly-uniform* in K if the total variation between μ and the uniform distribution on K is bounded by δ.

Some subroutines used in our algorithm require a centered convex body on their input. To be able to use these subroutines we need to find an affine transformation which makes a centering possible. Sufficiently many random points immediately will give such a transformation. The theorem below is a restatement of Corollary 11 in Bertsimas and Vempala (2004), which is based on Bourgain (1999), Rudelson (1999) and Kannan et al. (1997).

Theorem 7.5 *Using $s = O((d \ln d) \ln^2(1/\delta))$ independent samples from a δ-nearly-uniform distribution in K, one can find an affine transformation A such that $A(K)$ is in nearly-isotropic position, with probability at least $1 - s\delta$.*

Once we have the convex body in a nearly isotropic position we immediately obtain a centering. The following result is Corollary 5.2 (with $\vartheta = 1/4$) in Kannan et al. (1997).

Theorem 7.6 *Assume that K is in nearly isotropic position. Then*

$$B(0, 1/2) \subseteq K \subseteq B(0, 2(d+1)).$$

Once the convex body K is centered we can use shallow-cut ellipsoid algorithm to sandwich K between ellipsoids. The following is Theorem 2.4.1 in Lovász (1986) (combined with Theorem 2.2.14 in Lovász, 1986).

Theorem 7.7 *Let K be a (r_1, r_2)-centered convex body given by a membership oracle. A weak Löwner-John pair for K can be found in time polynomial in d and r_2/r_1.*

Finally, we will need to be able to generate random points from convex bodies. We will use the ball-walk algorithm (see Kannan et al., 1997, Theorem 2.2).

Theorem 7.8 *Let K be a (r_1, r_2)-centered convex body. A random point from a distribution ε-close to uniform can be found in time polynomial is d, r_2/r_1, and $\ln(1/\varepsilon)$.*

Now we describe a Las Vegas algorithm for one step of the shaped search problem. The input of the algorithm is a set W of d^8/γ independent δ-nearly-uniform random set of points in S_t, and a weak Löwner-John pair (E, E') for S_t. The algorithm has access to a membership oracle of S_{t+1}, where $S_{t+1} \subseteq S_t$ and $\mathrm{vol}(S_{t+1}) \geq \gamma \mathrm{vol}(S_t)$. The output is a set of d^7/γ independent δ-nearly-uniform random set of points in S_{t+1}, and a weak Löwner-John pair (F,F') for S_{t+1}. The algorithm runs in expected polynomial time.

We will use following objects in the algorithm. Let $Z \subseteq \mathbb{R}^d$ be the $2d$ points in which $B(0,1/2)$ intersects the axis of $\mathbb{R}^d$, i. e.,

$$Z = \{(1/2,0,\ldots,0),(-1/2,0,\ldots,0),\ldots,(0,\ldots,0,-1/2),(0,\ldots,0,1/2)\}.$$

Let r_1 be the radius of the largest ball contained in the convex hull of Z (i. e., $r_1 = 1/\sqrt{4d}$).

Finally, let $\delta = \exp(-\Theta(d^2))$.

1 $W' \leftarrow W \cap S_{t+1}$
2 Find an affine transformation A of Theorem 7.5, using points from W'. If W' does not contain enough points, let A be the identity.
3 Let r_2 be the radius of the smallest ball containing $A(E')$.
4 **if** Z is not inside $A(S_{t+1})$ or $r_2 > 4(d+1)\sqrt{d}$ **then**
5   generate independent uniformly random points from E' until we obtain d^6 random samples from S_{t+1}, let W' be the set of these new points. Go to step 2)
6 Use Theorem 7.7 to find a weak Löwner-John pair (F,F') for S_{t+1}, using centering $B(0,r_1) \subseteq A(S_{t+1}) \subseteq B(0,r_2)$.
7 Use Theorem 7.8 to find d^8/γ independent δ-nearly-uniform random points in S_{t+1}.

Algorithm 3: One step in the shaped-search algorithm for general convex bodies.

Theorem 7.9 *The algorithm 3 is correct, and its expected running-time is bounded by a polynomial in d.*

Proof :
Once we are on line 6 of the algorithm, we have

$$B(0,r_1) \subseteq Z \subseteq A(S_{t+1}) \subseteq A(E') \subseteq B(0,r_2),$$

and $r_2/r_1 \leq 8(d+1)d$. Thus $A(S_{t+1})$ is centered and the shallow-cut ellipsoid algorithm finds a weak Löwner-John pair (F,F') for S_{t+1}. Similarly the ball-walk algorithm gives δ-nearly-uniform samples from S_{t+1}. It remains to analyze lines 1-5 of the algorithm.

We enter line 5 only if $A(S_{t+1})$ is not nearly-isotropic. This can happen for two reasons: the number of points in W' is smaller than d^6, or the algorithm of Theorem 7.5 fails. Both these events have probability bounded by $\exp(-\Omega(d^2))$. The cost per sample on line 5 is $\text{vol}(E')/\text{vol}(S_{t+1}) = \exp(O(d \ln d))$. Hence the total contribution of line 5 to the total number of queries is $O(1)$. ∎

8. Active Learning

One of the earliest works in which the framework allows the learner to choose examples is by Eisenberg and Rivest (1990). In this work, the learner does not have access to unlabeled samples, but is allowed to query the membership oracle with any instance of its choice. (see also Angluin, 1988; Bshouty and Eiron, 2003; Jackson, 1997, for a similar setting). They showed a negative result stating that there are certain concept classes which are "dense in themselves", meaning that a small number of queries (even if chosen by the learner) are not enough to determine the target concept well. This result gave rise to a further line of work, the query by committee algorithm of Freund et al. (1997), in which the learner has access to an oracle of unlabeled samples also. Further, the learner is allowed to selectively query labels of some of these samples generated from the unlabeled oracle. Under this setting, it was shown that certain "dense in themselves" classes, for example homogeneous perceptrons under the uniform distribution, are efficiently learnable using a small number of labeled queries. Much modern active learning work uses this framework of having an unlabeled oracle and a membership oracle that can answer queries from examples generated from the unlabeled oracle. For example, Dasgupta et al. (2005) showed again (using a simpler method) that homogeneous perceptrons are learnable using only $O^*(d\ln(1/\varepsilon))$ labeled queries. Several other results are presented in Castro et al. (2006) and Dasgupta (2006). An exception to this framework is the recent work on active sampling by Fine and Mansour (2006). In their task, the learner has access to the oracle of a multi-valued function and has to find at least one instance of each example. Their work is related to our work in at least two ways. First, like us, they do not want to learn the concept, rather have just one positive example of the concept. Second, they allow the learner to choose its own examples.

The concept class of rectangles has been popular in the machine learning literature as rectangles are geometrically simple objects and also yield excellent results experimentally (Dietterich et al., 1997). Several theoretical results also exist for rectangles. For example, Auer et al. (1998) give an algorithm to PAC learn rectangles in $O(d/\varepsilon)$ queries, which matches the lower bound up to a multiplicative factor. Goldberg et al. (1994) give algorithms to learn the more complicated class of union of rectangles.

In this section, we show that rectangles are active learnable by using a variant of the bootstrap algorithm (Algorithm 1) in $O(d\ln(1/\varepsilon))$ labeled queries. We adopt the standard active learning framework of having an oracle that generates random samples and another oracle that can label these samples on request. Note that our current bootstrap algorithm does not use this flexibility and gets labeled samples uniformly at random from inside the outer body A_1 (see Algorithm 1). However, it clearly gives a (weak) upper bound to active learning the concept class of rectangles in $O(d/\varepsilon)$ labeled samples under the uniform distribution. In this section, we give a variant of the bootstrapping algorithm and show how it can be repeatedly used to active learn rectangles using both labeled and unlabeled oracles with only $O(d\ln(1/\varepsilon))$ labeled queries. Our algorithm for active learning rectangles is a one-sided active learning algorithm, that is, it outputs a hypothesis which is a superset of the target concept. We now define one-sided active learning.

Definition 8.1 A concept class $\mathcal{C}$ is *one-sided active learnable* under the uniform distribution over the instance space X if there is an algorithm, that for any concept $c \in \mathcal{C}$ and $0 < \varepsilon < 1$, gets $O(1/\varepsilon)$ samples from the uniform distribution on X and uses the membership oracle of c to label $O(\ln(1/\varepsilon))$ of these samples, and outputs a concept h such that $c \subseteq h$, and $P(h(x) \neq c(x)) < \varepsilon$.

Observation 8.1 *The concept class of axis-parallel rectangles inside the d-dimensional cube* $[0,1]^d$ *is not one-sided active learnable.*

Consider a rectangle with volume ε. Then we need, in expectation, $O(1/\varepsilon)$ labeled samples just to find one point inside the rectangle. Thus we are making exponentially more queries to the membership oracle than desired. Note that this is going to be a problem in learning *any* concept class which has concepts which have a small measure. For example, Dasgupta (2005) pointed out that learning non-homogeneous perceptrons when X is a unit sphere requires $\Omega(1/\varepsilon)$ labeled samples. They overcame this problem by restricting the class to only homogeneous perceptrons (passing through the center of the sphere).

In the same spirit, we assume that our concept class has rectangles which are larger than some constant value and show that active learning is possible in this case.

Definition 8.2 The concept class $\mathcal{C}$ of *big rectangles* is a set of axis-parallel rectangles R such that $R \subset [0,1]^d$ and $\mathrm{vol}(R) > 1/2$.

8.1 The Concept Class of Big Rectangles is One-sided Active Learnable

Throughout this section $\mathcal{C}$ denotes the concept class of axis-parallel rectangles. Note that the bootstrapping algorithm (Algorithm 1) for $\mathcal{C}$ takes as input an outer rectangle A_1 and two parameters α and β such that $\mathrm{vol}(A_1) < \alpha \cdot \mathrm{vol}(R)$ and outputs a rectangle A_2 such that $\mathrm{vol}(A_2) < \beta \cdot \mathrm{vol}(R)$. The algorithm samples $O(d\alpha/\ln\beta)$ points from the membership oracle of R. Notice that the algorithm actually constructs the minimal volume rectangle (call it B_2) containing all positive samples and guarantees that $\mathrm{vol}(A_2) < \beta \cdot \mathrm{vol}(B_2)$. We make use of this inner approximation in the active learning algorithm.

input : A representation $A_1 \in \mathcal{C}$ and $B_1 \in \mathcal{C}$. An oracle for $R \in \mathcal{C}$. A sampler S which samples uniformly at random points from A_1. A number $\beta > 1$.
assume : $B_1 \subseteq R \subseteq A_1$.
output : a representation of $B_2 \in \mathcal{C}$ and $A_2 \in \mathcal{C}$, such that $B_2 \subseteq R \subseteq A_2$ and $\mathrm{vol}(A_2) \leq \beta \cdot \mathrm{vol}(B_2)$.

1 $S^+ \leftarrow \emptyset$, $S^- \leftarrow \emptyset$
2 **repeat**
3 pick a random point $p \in A_1$ using S;
4 **if** $p \notin A_1 - B_1$ **then goto** step 3
5 **if** $p \in R$ **then** $S^+ \leftarrow S^+ \cup \{p\}$ **else** $S^- \leftarrow S^- \cup \{p\}$ **fi**
6 $\mathrm{Possible}_R \leftarrow \{C \in \mathcal{C} \mid S^+ \subseteq C \subseteq A_1 \setminus S^-\}$
7 $B_2 \leftarrow$ the smallest axis-parallel rectangle containing S^+
8 $A_2 \leftarrow$ the axis-parallel rectangle of minimal volume containing all $C \in \mathrm{Possible}_R$
9 **until** $\mathrm{vol}(A_2) \leq \beta \cdot \mathrm{vol}(B_2)$
10 output a representation of A_2 and B_2

Algorithm 4: Modified Inner-Outer algorithm used for active learning

Lemma 8.3 *Let $\alpha > \beta > 1$. Let E be the expected number of membership-oracle calls of the modified Inner-outer algorithm on input $B_1 \subseteq R \subseteq A_1$ and β.*

1. *If* $\mathrm{vol}(A_1) \leq \alpha \cdot \mathrm{vol}(R)$ *then* $E = O(d\alpha/\ln\beta)$.
2. *If* $\mathrm{vol}(A_1) \leq \alpha \cdot \mathrm{vol}(B_1)$ *then* $E = O(d(\alpha - 1)/\ln\beta)$.

Proof :
By direct application of Lemma 4.6, the expected number of random points picked in step 3 of the algorithms is bounded by $8 + d\alpha/\ln\beta$. This proves part 1.

For part 2., note the modification made in step 4 of algorithm. We only query points which lie in the region between A_1 and B_1, thus ignoring at least $1/\alpha$ fraction of the region A_1 (as $\mathrm{vol}(A_1) \leq \alpha \cdot \mathrm{vol}(B_1)$). Hence the expected number of queries to the membership oracle is $(1 - 1/\alpha)(8 +$

$d\alpha/\ln\beta)$, which is $O(1+d(\alpha-1)/\ln\beta) = O(d(\alpha-1)/\ln\beta)$ (in the last containment we used $d(\alpha-1)/\ln\beta \geq d(\alpha-1)/\ln\alpha \geq d = \Omega(1))$. ■

The algorithm above requires a sampler S that samples uniformly random points from A_1. This sampler can be easily obtained from the unlabeled sampler of the instance space X using rejection sampling. This increases the number of samples needed by a constant factor, as $\text{vol}(A_1) \geq \text{vol}(R) > 1/2$.

We now repeatedly apply this bootstrapping procedure to do active learning.

input : An oracle O to generate unlabeled samples from the instance space $X = [0,1]^d$. The membership oracle of an axis-pallalel rectangle R. A parameter $\varepsilon > 0$.
assume : $\text{vol(R)} \geq 1/2$.
output : a representation of an axis-parallel rectangle A, such that $R \subseteq A$ and $\text{vol}(A) < (1+\varepsilon)\text{vol}(R)$

1 $(A,B) \leftarrow$ output of Algorithm 4 with $A_1 = X$, $B_1 = \emptyset$, and $\beta = 1+\varepsilon 2^{\lceil \log_2 1/\varepsilon \rceil}$
2 **for** $i \leftarrow \lceil \log_2 1/\varepsilon \rceil$ **to** 1 **do**
3 $\quad$ $(A,B) \leftarrow$ output of Algorithm 4 with $A_1 = A$, $B_1 = B$, $\beta = 1+\varepsilon 2^{i-1}$
4 output a representation of A

Algorithm 5: Algorithm to do one-sided active learning

Theorem 8.4 *The expected number membership-queries made by Algorithm 5 is bounded by* $O(d\ln(1/\varepsilon))$.

Proof :
By Lemma 8.3, part 1., the number of membership queries at step 1. is bounded by $O(d)$.

By Lemma 8.3, part 2., at each iteration the number membership queries by Algorithm 4 on step 3. is bounded by

$$O(d(1+2^i\varepsilon-1)/\ln(1+2^{i-1}\varepsilon)) = O(d),$$

where in the last step we used the fact that $\ln(1+x) \geq x - x^2/2$ for $x \geq 0$. The total number of iterations is $\lceil \log_2(1/\varepsilon) \rceil$. Hence the total number of calls to the membership oracle is $O(d\ln(1/\varepsilon))$. ■

At the end of learning, $\text{vol}(A) \leq (1+\varepsilon)\cdot\text{vol}(R)$. Hence, $\text{vol}(A) - \text{vol}(R) < \varepsilon\cdot\text{vol}(R) < \varepsilon$. Further, $R \subset A$. Hence, big rectangles are one-sided active learnable.

Note that a trivial algorithm to learn big rectangles would be to learn each face at a time. This can be done by doing binary search starting from $(1/2,\ldots 1/2)$. As there are d faces, this will take $O(d\ln\frac{d}{\varepsilon})$ labeled samples (since precision ε/d is required along each dimension).

9. Discussion and Future Work

In this paper, we introduced a new framework of learning using oracles of increasingly restrictive concepts. Our framework has been inspired from the biological phenomenon of behavioral shaping in which a target behavior is taught to a subject by teaching it successively better approximations of the behavior, eventually converging to the target behavior. Analogous to behavioral shaping, in a shaped search problem, the learner is given access to a sequence of membership oracles of increasingly restrictive concepts and the learner is required to output one sample from the target concept.

We gave efficient algorithms to solve the shaped search problem for the concept class of intervals, axis-parallel rectangles, bounded eccentricity ellipsoids, and general convex bodies. While we have matching lower and upper bounds for the concept class of intervals, for other concept classes we do not understand the complexity of the shaped search problem (i. e., our lower and upper bounds do not match). The concept class of axis-parallel rectangles is a natural question to consider next.

Question 5 *Let $\mathcal{C}$ be the concept class of axis-parallel rectangles in $\mathbb{R}^d$. What is the complexity of the shaped search problem?*

The bootstrapping technique of Section 4 was useful for both the shaped search problem and active learning for axis-parallel rectangles. Whether efficient bootstrapping algorithms exist for other concept classes is an interesting problem.

Question 6 *For which concept classes is bootstrapping possible?*

Another technique that was useful in our setting was a deterministic ray shooting algorithm. By keeping track of the centroid we were able to solve the shaped search problem for bounded eccentricity ellipsoids. One can imagine that such an approach might work for any centrally symmetric convex body.

Question 7 *Let K be a centrally symmetric convex body given by a membership oracle. Can the centroid of K be found by an efficient deterministic algorithm?*

Our solution of the shaped search problem for general convex bodies samples random points from convex bodies and hence heavily relies on randomization. Is the use of randomness inevitable?

Question 8 *Let $\mathcal{C}$ be the concept class of (centrally symmetric) convex bodies in $\mathbb{R}^d$. Does there exist a deterministic agent for the γ-shaped search problem, using $O(\mathrm{poly}(d, 1/\gamma))$ queries per oracle?*

Another interesting direction of future work is to apply this model to active learning. Active learning, in general, does not provide any advantage for concepts which have a "small volume". For example, it was observed in Dasgupta (2005) that when the instance space is a unit ball, the concept class of non-homogeneous perceptrons is not active learnable (in the sense that any active learning scheme requires a large number of labeled samples). This is because a perceptron, in general, can pass through the sphere in such a way that it leaves a very small "cap" of the ball on one side, and just sampling one example from this cap might require a large number of labeled samples. Our shaping model can be seen as a way of directing search to such events of low probability. One way to remedy this problem is to consider a sequence of perceptrons which divide the ball into increasingly asymmetric parts and ultimately lead to the final perceptron. Whether non-homogeneous perceptrons are learnable under this framework is an interesting direction.

As our model has been inspired from behavioral shaping, we restrict the oracles to be presented in a temporal fashion. One could consider a framework in which all the oracles are present simultaneously and the agent can query any oracle at any time. This simultaneous oracle model can be viewed as a special case of the "reward shaping" of Ng et al. (1999). Under what conditions are these models equivalent?

Acknowledgments

The authors thank to the anonymous referees for many helpful corrections, and suggestions.

The authors are listed in alphabetical order.

Appendix A.

In this section we prove Lemmas 4.4 and 4.5 (in Subsection A.1), and Lemma 2.9 (in Subsection A.2). Finally we comment on the optimality of Lemma 2.9 (in Subsection A.3).

A.1 Technical Results about Maxima and Minima of Random Variables

Proof of Lemma 4.4:
Let Y be the minimum and Z be the maximum of the X_i. For $0 \leq y \leq z \leq 1$, the density function of (Y,Z) is

$$-\frac{\partial}{\partial y}\frac{\partial}{\partial z}(z-y)^n,$$

and hence

$$E[-\ln(Z-Y)] = \int_0^1 \int_y^1 \ln(z-y)\frac{\partial}{\partial y}\frac{\partial}{\partial z}(z-y)^n \, \mathrm{d}z\,\mathrm{d}y = \frac{2n-1}{n(n-1)}.$$

■

Proof of Lemma 4.5:
Conditioning on the value of K we obtain

$$E[\min\{1,X_1,\ldots,X_K\} \,|\, K=k] = \int_0^1 (1-x)\frac{\partial}{\partial x}x^k \, \mathrm{d}x = \frac{1}{k+1},$$

and hence

$$E[\min\{1,X_1,\ldots,X_K\}] = \sum_{k=0}^{n} \binom{n}{k} p^k (1-p)^{n-k} \frac{1}{k+1} = \frac{1-(1-p)^{n+1}}{(n+1)p}.$$

■

A.2 Bounding the 2-norm of the Ray-shooting Matrix

In this section we prove Lemma 2.9. Our goal is to understand the square of the 2-norm of

$$M(\alpha) = \prod_{i=1}^{n}\left(I - \alpha \cdot \frac{Dv_i v_i^{\mathrm{T}} D}{v_i^{\mathrm{T}} D^2 v_i}\right)$$

as a function of α (the 2-norm of $M(\alpha)$ measures how much closer to the centroid does a point get in the ray-shooting algorithm of Theorem 2.8).

We can understand the value of $\|M(\alpha)\|_2^2$ for $\alpha = 0$ and $\alpha = 1$ but this does not give us much information about $\|M(\alpha)\|_2^2$ for other values of α. In order to obtain this information, we are going show $\|M(\alpha)\|_2^2 \leq 1 - L^2\alpha(2-\alpha)/||G(\alpha)||_2^2$, where the entries of $G(\alpha)$ are linear functions of α (Equations 16 and 17). It will turn out that for $G(\alpha)$ we can understand $\|G(\alpha)\|_2^2$ for $\alpha = 0$ and

$\alpha = 1$. Now, since dependence of $G(\alpha)$ on α is much simpler than the dependence of $M(\alpha)$ on α, we will be able to obtain an upper bound on $\|G(\alpha)\|_2^2$ for values of $\alpha \in [0,1]$, which, in turn, will imply an upper bound on $\|M(\alpha)\|_2^2$.

Note that scaling the v_i does not change M and hence we will, w.l.o.g., assume $\|v_i\|_2 = 1$. Let $\gamma_{ij} = v_i^{\mathrm{T}} D^2 v_j$. Let $G(\alpha)$ be the upper triangular $d \times d$ matrix defined by

$$G_{ij} = \begin{cases} \sqrt{\gamma_{jj}} & \text{for } i = j, \\ (\gamma_{ij}/\sqrt{\gamma_{jj}}) \cdot \alpha & \text{for } i < j, \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

For $\alpha \in (0,2)$ we let

$$\Gamma(\alpha) = G(\alpha)/\sqrt{\alpha(2-\alpha)}. \tag{15}$$

Let V be the $d \times d$ matrix with columns $v_1, \ldots, v_d$. To prove Lemma 2.9 we will need the following two auxiliary results.

Lemma A.1 *Let $\Gamma = \Gamma(1/\sqrt{d})$ be the matrix defined by* (15) *(with $\alpha = 1/\sqrt{d}$). Then*

$$\|\Gamma\|_2^2 \le 5\sqrt{d}.$$

Lemma A.2 *For M, D, V, Γ defined above $\|Mx\|_2^2 = \|x\|_2^2 - \|\Gamma^{-1} V^{\mathrm{T}} D x\|_2^2$. Moreover*

$$\|M\|_2^2 = 1 - \frac{1}{\lambda_{\max}(\Gamma^{\mathrm{T}} V^{\mathrm{T}} D^{-2} V \Gamma)}, \tag{16}$$

(where $\lambda_{\max}(A)$ is the largest eigenvalue of A).

We postpone the proof of Lemma's A.1 and A.2 after the proof of Lemma 2.9.

Proof of Lemma 2.9:
Since $D^{-2} \preceq L^2 \cdot I$ we have

$$\lambda_{\max}(\Gamma^{\mathrm{T}} V^{\mathrm{T}} D^{-2} V \Gamma) \le L^2 \cdot \lambda_{\max}(\Gamma^{\mathrm{T}} V^{\mathrm{T}} I V \Gamma) = L^2 \cdot \lambda_{\max}(\Gamma^{\mathrm{T}} \Gamma) = L^2 \cdot \|\Gamma\|_2^2. \tag{17}$$

Now applying Lemma A.1 (with $\alpha = 1/\sqrt{d}$) we get

$$\lambda_{\max}(\Gamma^{\mathrm{T}} V^{\mathrm{T}} D^{-2} V \Gamma) \le 5L^2\sqrt{d}.$$

Now, using Lemma A.2 we obtain the result. ■

Proof of Lemma A.2:
Let Γ_k be the $k \times k$ top-left submatrix of Γ. Let V_k be the $d \times k$ matrix consisting of the first k columns of V. Let

$$M_k = \prod_{i=1}^{k} \left(I - \alpha \cdot \frac{D v_i v_i^{\mathrm{T}} D}{v_i^{\mathrm{T}} D^2 v_i} \right).$$

By induction on k we will show that for any x

$$\|M_k x\|_2^2 = \|x\|_2^2 - \|\Gamma_k^{-1} V_k^{\mathrm{T}} D x\|_2^2. \tag{18}$$

For $k = 1$ we have

$$M_1 = I - \alpha \frac{Dv_1 v_1^{\mathrm{T}} D}{v_1^{\mathrm{T}} D^2 v_1}, \quad M_1 x = x - \alpha \frac{(v_1^{\mathrm{T}} Dx)}{v_1^{\mathrm{T}} D^2 v_1} Dv_1, \quad \|M_1 x\|_2^2 = \|x\|_2^2 - \alpha(2-\alpha)\frac{\left(v_1^{\mathrm{T}} Dx\right)^2}{v_1^{\mathrm{T}} D^2 v_1}.$$

Moreover $V_1^{\mathrm{T}} = v_1^{\mathrm{T}}$, $\Gamma_1^{-2} = \alpha(2-\alpha)/(v_1^{\mathrm{T}} D^2 v_1)$, and hence

$$\|\Gamma_1^{-1} V_1^{\mathrm{T}} Dx\|_2^2 = \frac{\alpha(2-\alpha)}{v_1^{\mathrm{T}} D^2 v_1}(v_1^{\mathrm{T}} Dx)^2.$$

We showed that (18) holds for $k = 1$. Now we assume $k > 1$.

If $v_k^{\mathrm{T}} Dx = 0$ then

$$\left(I - \alpha \cdot \frac{Dv_k v_k^{\mathrm{T}} D}{v_k^{\mathrm{T}} D^2 v_k}\right) x = x,$$

and hence $M_{k-1}x = M_k x$. Moreover the last entry of $V_k^{\mathrm{T}} Dx$ is zero and hence $\|\Gamma_{k-1}^{-1} V_{k-1}^{\mathrm{T}} Dx\|_2 = \|\Gamma_k^{-1} V_k^{\mathrm{T}} Dx\|_2$. Thus (18) holds by the induction hypothesis.

Now we assume $v_k^{\mathrm{T}} Dx \neq 0$. W.l.o.g., we can assume $v_k^{\mathrm{T}} Dx = 1$. Let

$$x' = \left(I - \alpha \cdot \frac{Dv_k v_k^{\mathrm{T}} D}{v_k^{\mathrm{T}} D^2 v_k}\right) x = x - \alpha \cdot \frac{Dv_k}{\gamma_{kk}}. \tag{19}$$

We have $\|x'\|_2^2 = \|x\|_2^2 - \alpha(2-\alpha)/\gamma_{kk}$.

Let $b = V_k^{\mathrm{T}} Dx$ and $b' = V_{k-1}^{\mathrm{T}} Dx'$. If we show

$$\|\Gamma_k^{-1} b\|_2^2 = \frac{\alpha(2-\alpha)}{\gamma_{kk}} + \|\Gamma_{k-1}^{-1} b'\|_2^2 \tag{20}$$

then

$$\|M_k x\|_2^2 = \|M_{k-1} x'\|_2^2 = \|x'\|_2^2 - \|\Gamma_{k-1}^{-1} b'\|_2^2 = \|x\|_2^2 - \frac{\alpha(2-\alpha)}{\gamma_{kk}} - \|\Gamma_{k-1}^{-1} b'\|_2^2 = \|x\|_2^2 - \|\Gamma_k^{-1} b\|,$$

and we are done. Thus it remains to show (20).

From (19) we have for $i = 1, \ldots, k-1$,

$$b_i' = b_i - \alpha \frac{\gamma_{ik}}{\gamma_{kk}}.$$

Let Z be the $(k-1) \times k$ matrix taking b to b', i. e.,

$$Z = \begin{cases} 1 & \text{for } i = j, \\ -\alpha(\gamma_{ik}/\gamma_{kk}) & \text{for } j = k, \\ 0 & \text{otherwise.} \end{cases}$$

Let $y = \Gamma_k^{-1} b$ and $y' = \Gamma_{k-1}^{-1} b'$. Note that $y' = \Gamma_{k-1}^{-1} Z \Gamma_k y$.

If the last coordinate of y is zero then the last coordinate of $\Gamma_k y$ is zero as well and hence Z acts as identity on $\Gamma_k y$. Thus

$$\begin{aligned}\left(\Gamma_{k-1}^{-1} Z \Gamma_k\right)(y_1, \ldots, y_{k-1}, 0)^{\mathrm{T}} = \Gamma_{k-1}^{-1} I \Gamma_k (y_1, \ldots, y_{k-1}, 0)^{\mathrm{T}} = \\ \Gamma_{k-1}^{-1} I \Gamma_{k-1} (y_1, \ldots, y_{k-1})^{\mathrm{T}} = (y_1, \ldots, y_{k-1})^{\mathrm{T}}.\end{aligned}$$

Thus the left $(k-1)\times(k-1)$ submatrix of $\Gamma_{k-1}^{-1}Z\Gamma_k$ is the identity matrix.

For any $i=1,\ldots,k-1$ we have

$$e_i^{\mathrm{T}}Z\Gamma_k e_k = \left(e_i^{\mathrm{T}} - \alpha\frac{\gamma_{ik}}{\gamma_{kk}}e_k^{\mathrm{T}}\right)\Gamma_k e_k = \frac{\gamma_{ik}}{\sqrt{\gamma_{kk}}}\sqrt{\alpha/(2-\alpha)} - \alpha\frac{\gamma_{ik}}{\gamma_{kk}}\sqrt{\gamma_{kk}}/\sqrt{\alpha(2-\alpha)} = 0.$$

Thus the last column of $\Gamma_{k-1}^{-1}Z\Gamma_k$ is zero. Hence

$$\|y'\|_2^2 = \|y\|_2^2 - y_k^2. \tag{21}$$

Since $y=\Gamma_k^{-1}b$, $b_k=1$, and Γ_k is upper triangular matrix we have that

$$y_k = (\Gamma_k^{-1})_{kk} = \frac{1}{(\Gamma_k)_{kk}} = \sqrt{\frac{\alpha(2-\alpha)}{\gamma_{kk}}}.$$

Plugging y_k into (21) we obtain (20). We completed the induction step and hence (18) is true for all $k=1,\ldots,n$. We proved the first part of the lemma.

To show the second part of the lemma we observe

$$\|M\|_2^2 = \max_{\|x\|_2=1}\|Mx\|_2^2 = 1 - \min_{\|x\|_2=1}\|\Gamma^{-1}V^{\mathrm{T}}Dx\|_2^2 = 1-\lambda_{\min}\left(D^{-1}V\Gamma^{-T}\Gamma^{-1}V^{\mathrm{T}}D\right).$$

Let $A=\Gamma^{-1}V^{\mathrm{T}}D=(D^{-1}V\Gamma)^{-1}$. We have

$$\lambda_{\min}(A^{\mathrm{T}}A) = \lambda_{\min}(AA^{\mathrm{T}}) = \frac{1}{\lambda_{\max}(A^{-T}A^{-1})} = \frac{1}{\lambda_{\max}(\Gamma^{\mathrm{T}}V^{\mathrm{T}}D^{-2}V\Gamma)}.$$

■

Lemma A.3 *Let A be an $d\times d$ symmetric matrix such that $0\preceq A\preceq I$ (i. e., A is positive semi-definite and all its eigenvalues are ≤ 1). Then*

$$\sum_{i=1}^{d}\sum_{j=1}^{d}\frac{A_{ij}^2}{A_{jj}} \leq d. \tag{22}$$

Proof :
The eigenvalues of A are between 0 and 1 and hence we have $A^2\preceq A$. Thus

$$\sum_{i=1}^{d}A_{ij}^2 = (A^2)_{jj} \leq A_{jj}. \tag{23}$$

Dividing both sides of (23) by A_{jj} and summing over $j\in[d]$ we obtain the result. ■

Consider $G(1)$ defined by (14) with $\alpha=1$. We can bound $\|G(1)\|_F^2$ (the square of the Frobenius norm) by (22) where we take $A=V^{\mathrm{T}}DV=(\gamma_{ij})_{i,j=1}^d$. Using $\|A\|_2\leq\|A\|_F$ we obtain the following bound.

Corollary A.4 *Let $G(1)$ be defined by (14) with $\alpha=1$. Then*

$$\|G(1)\|_2 \leq \sqrt{d}. \tag{24}$$

For $\alpha = 0$ the matrix G is diagonal with all the diagonal entries ≤ 1. Hence we have:

Observation A.1 *Let $G(0)$ be defined by* (14) *with $\alpha = 0$. Then*

$$\|G(0)\|_2 \leq 1. \tag{25}$$

Let $F(\alpha) = G(\alpha)^{\mathrm{T}} G(\alpha)$. Then

$$\frac{\mathrm{d}}{\mathrm{d}\alpha} F(0) = \begin{cases} \gamma_{ij}\sqrt{\gamma_{ii}}/\sqrt{\gamma_{jj}} & \text{if } i < j, \\ \gamma_{ij}\sqrt{\gamma_{jj}}/\sqrt{\gamma_{ii}} & \text{if } i > j, \\ 0 & \text{if } i = j. \end{cases}$$

Note that $\gamma_{ii} \leq 1$ for $i \in [n]$. Hence, using Lemma A.3 we obtain the following bound.

Observation A.2

$$\left\| \frac{\mathrm{d}}{\mathrm{d}\alpha} F(0) \right\|_2^2 \leq \left\| \frac{\mathrm{d}}{\mathrm{d}\alpha} F(0) \right\|_F^2 \leq 2n. \tag{26}$$

Now we can finally prove Lemma A.1.

Proof of Lemma A.1:
Let x be such that $\|x\|_2 = 1$. Let $f(\alpha) = x^{\mathrm{T}} G(\alpha)^{\mathrm{T}} G(\alpha) x$. Note that f is a quadratic function of α. Let

$$f(\alpha) = a\alpha^2 + b\alpha + c.$$

From (24), (25), (26) we get that

$$a + b + c = f(1) \leq d,$$

$$c = f(0) \leq 1, \quad \text{and}$$

$$|b| = |f'(0)| \leq \sqrt{2d}.$$

Hence

$$f(1/\sqrt{d}) \leq (d + \sqrt{2d}) \cdot \left(\frac{1}{\sqrt{d}} \right)^2 + \sqrt{2d} \frac{1}{\sqrt{d}} + 1 \leq 5.$$

Let $\alpha = 1/\sqrt{d}$, and $G = G(\alpha)$. Since x was arbitrary we get

$$\|G\|_2^2 = \max_{\|x\|_2 = 1} x^{\mathrm{T}} G^{\mathrm{T}} G x \leq 5.$$

Finally,

$$\|\Gamma\|_2^2 = \frac{\|G\|_2^2}{\alpha(2 - \alpha)} \leq 5\sqrt{d}.$$

∎

A.3 Optimality of Lemma 2.9

Now we show that Lemma 2.9 cannot be improved (up to a constant factor).

Let D be a diagonal matrix with $D_{11} = 1$ and $D_{ii} = \varepsilon$ for $i = 2, \ldots, d$. Let

$$v_1 \propto (\sqrt{1/2}, \sqrt{\varepsilon}, \ldots, \sqrt{\varepsilon}, \sqrt{1/2}),$$

and let $v_1, \ldots, v_d$ be orthogonal. Let V have columns $v_1, \ldots, v_d$. Then $V^{\mathrm{T}} D^2 V = (1-\varepsilon^2) v_1 v_1^{\mathrm{T}} + \varepsilon^2 I$. From the definition (15) we immediately obtain

$$\Gamma = \Gamma(\alpha) = \frac{1}{\sqrt{\alpha(2-\alpha)}} \left(B + O(\varepsilon^{1/2}) \right),$$

where

$$B_{ij} = \begin{cases} \sqrt{1/2} & \text{if } i = j = 1 \text{ or } i = j = d, \\ \alpha\sqrt{1/2} & \text{if } i = 1 \text{ and } j \geq 2, \\ 0 & \text{otherwise.} \end{cases}$$

A short calculation yields

$$\Gamma^{\mathrm{T}} V^{\mathrm{T}} D^{-2} V \Gamma = \frac{1}{4\alpha(2-\alpha)} \cdot \frac{1}{\varepsilon^2} \cdot \left(w w^{\mathrm{T}} + O(\varepsilon) \right),$$

where $w = (1, -\alpha, \ldots, -\alpha, \alpha - 1)$, and hence

$$\lambda_{\max}(\Gamma^{\mathrm{T}} V^{\mathrm{T}} D^{-2} V) = \frac{1}{4\varepsilon^2} \left(\frac{(d-1)\alpha^2 - 2\alpha + 2}{\alpha(2-\alpha)} + O(\varepsilon) \right)$$

The minimum of $\frac{(d-1)\alpha^2 - 2\alpha + 2}{\alpha(2-\alpha)}$ occurs at $\alpha = (-1 + \sqrt{2d-3})/(d-2)$. For this value of α we have

$$\frac{(d-1)\alpha^2 - 2\alpha + 2}{\alpha(2-\alpha)} \sim \sqrt{d/8}$$

as $d \to \infty$. Thus we have the following.

Observation A.3 *For any* $\alpha \in (0, 2)$,

$$\lambda_{\max}(\Gamma^{\mathrm{T}} V^{\mathrm{T}} D^{-2} V) \gtrapprox \frac{\sqrt{d/8}}{4\varepsilon^2}.$$

For $\alpha = 1$

$$\lambda_{\max}(\Gamma^{\mathrm{T}} V^{\mathrm{T}} D^{-2} V) \approx \frac{d}{4\varepsilon^2}.$$

References

Dana Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, 1988.

Martin Anthony and Norman Biggs. *Computational Learning Theory: An Introduction*. Cambridge University Press, New York, NY, USA, 1992.

Peter Auer, Philip M. Long, and Aravind Srinivasan. Approximating hyper-rectangles: Learning and pseudorandom sets. *Journal of Computer and System Sciences*, 57(3):376–388, 1998.

Dimitris Bertsimas and Santosh Vempala. Solving convex programs by random walks. *Journal of the ACM*, 51(4):540–556, July 2004. doi: 10.1145/1008731.1008733.

Jean Bourgain. *Random Points in Isotropic Convex Sets. Convex Geometric Analysis*, pages 53–58. Cambridge University Press, Cambridge, 1999.

Nader H. Bshouty and Nadav Eiron. Learning monotone DNF from a teacher that almost does not answer membership queries. *Journal of Machine Learning Research*, 3:49–57, 2003.

Rui Castro, Rebecca Willett, and Robert Nowak. Faster rates in regression via active learning. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 179–186. MIT Press, Cambridge, MA, 2006.

Sanjoy Dasgupta. Analysis of a greedy active learning strategy. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 337–344. MIT Press, Cambridge, MA, 2005.

Sanjoy Dasgupta. Coarse sample complexity bounds for active learning. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 235–242. MIT Press, Cambridge, MA, 2006.

Sanjoy Dasgupta, Adam Tauman Kalai, and Claire Monteleoni. Analysis of perceptron-based active learning. In *Proceedings of the eighteenth Annual Conference on Learning Theory*, pages 249–263, 2005.

Thomas G. Dietterich, Richard H. Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1-2):31–71, 1997. ISSN 0004-3702.

Marco Dorigo and Marco Colombetti. Robot shaping: Developing autonomous agents through learning. *Artificial Intelligence*, 71(2):321–370, 1994.

Martin Dyer, Alan Frieze, and Ravi Kannan. A random polynomial-time algorithm for approximating the volume of convex bodies. *Journal of the ACM*, 38(1):1–17, 1991.

Bonnie Eisenberg and Ronald L. Rivest. On the sample complexity of PAC-learning using random and chosen examples. In *Proceedings of the Third Annual Workshop on Computational Learning Theory*, pages 154–162, San Francisco, CA, USA, 1990. Morgan Kaufmann Publishers Inc.

Shai Fine and Yishay Mansour. Active sampling for multiple output identification. In *Proceedings of the Nineteenth Annual Conference on Learning Theory, COLT 2006, Pittsburgh, PA, USA, June 2006*, volume 4005 of *Lecture Notes in Artificial Intelligence*, pages 620–634. Springer, Berlin, 2006.

Yoav Freund, H. Sebastian Seung, Eli Shamir, and Naftali Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28(2-3):133–168, 1997.

Paul W. Goldberg, Sally A. Goldman, and H. David Mathias. Learning unions of boxes with membership and equivalence queries. In *COLT '94: Proceedings of the Seventh Annual Conference on Computational Learning Theory*, pages 198–207, New York, NY, USA, 1994. ACM Press. ISBN 0-89791-655-7.

Sally A. Goldman and Michael J. Kearns. On the complexity of teaching. In *COLT '91: Proceedings of the Fourth Annual Workshop on Computational Learning Theory*, pages 303–314, San Francisco, CA, USA, 1991. Morgan Kaufmann Publishers Inc. ISBN 1-55860-213-5.

Sally A. Goldman and H. David Mathias. Teaching a smart learner. In *COLT '93: Proceedings of the Sixth Annual Conference on Computational Learning Theory*, pages 67–76, New York, NY, USA, 1993. ACM Press. ISBN 0-89791-611-5.

Sally A. Goldman, Ronald L. Rivest, and Robert E. Schapire. Learning binary relations and total orders. *SIAM J. Comput.*, 22(5):1006–1034, 1993. ISSN 0097-5397.

Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, Berlin, 1988.

Tibor Hegedűs. Combinatorial results on the complexity of teaching and learning. In *MFCS '94: Proceedings of the 19th International Symposium on Mathematical Foundations of Computer Science 1994*, pages 393–402, London, UK, 1994. Springer-Verlag. ISBN 3-540-58338-6.

Jeffrey C. Jackson. An efficient membership-query algorithm for learning DNF with respect to the uniform distribution. *Journal of Computer and System Sciences*, 55(3):414–440, 1997. doi: http://dx.doi.org/10.1006/jcss.1997.1533.

Ravi Kannan, László Lovász, and Miklós Simonovits. Random walks and an $O^*(n^5)$ volume algorithm for convex bodies. *Random Structures and Algorithms*, 11(1):1–50, 1997.

Michael Kearns and Umesh Vazirani. *An Introduction to Computational Learning Theory*. MIT Press, Cambridge, MA, 1994.

George Konidaris and Andrew Barto. Autonomous shaping: Knowledge transfer in reinforcement learning. In *Proceedings of the Twenty-third International Conference on Machine Learning*, pages 489–496, New York, NY, USA, 2006. ACM Press.

Nati Linial, Michael Luby, Michael Saks, and David Zuckerman. Efficient construction of a small hitting set for combinatorial rectangles in high dimension. In *Proceedings of the Twenty-fifth Annual ACM Symposium on Theory of Computing*, pages 258–267, New York, NY, USA, 1993. ACM Press.

László Lovász. *An Algorithmic Theory of Numbers, Graphs and Convexity*, volume 50 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1986.

Maja J. Mataric. Reward functions for accelerated learning. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 181–189, 1994.

Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, Cambridge, 1995. ISBN 0-521-47465-5.

Andrew Y. Ng, Daishi Harada, and Stuart J. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning*, pages 278–287, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

Jette Randløv and Preben Alstrøm. Learning to drive a bicycle using reinforcement learning and shaping. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 463–471, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

Mark Rudelson. Random vectors in isotropic position. *Journal of Functional Analysis*, 164(1): 60–72, 1999.

Burrhus F. Skinner. *The Behavior of Organisms*. Appleton-Century-Crofts, New York, NY, USA, 1938.

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