Electronic Colloquium on Computational Complexity, Report No. 72 (2000)

Approximating Hyper-Rectangles: Learning and Pseudo-random Sets

Peter Auer[†] Philip M. Long[‡] Aravind Srinivasan[§]

Abstract

The PAC learning of rectangles has been studied because they have been found experimentally to yield excellent hypotheses for several applied learning problems. Also, pseudorandom sets for rectangles have been actively studied recently because (i) they are a subproblem common to the derandomization of depth-2 (DNF) circuits and derandomizing Randomized Logspace, and (ii) they approximate the distribution of n independent multivalued random variables. We present improved upper bounds for a class of such problems of "approximating" high-dimensional rectangles that arise in PAC learning and pseudorandomness.

Key words and phrases. Rectangles, machine learning, PAC learning, derandomization, pseudorandomness, multiple-instance learning, explicit constructions, Ramsey graphs, random graphs, sample complexity, approximations of distributions.

1 Introduction

A basic common theme of a large part of PAC learning and derandomization/computational pseudorandomness is to "approximate" a structure using a "small" number of representative random examples. A commonly studied such type of structure in both learning and derandomization are the n-dimensional combinatorial rectangles, i.e., sets of the form $S_1 \times S_2 \times \cdots \times S_n$, where each $S_i \subseteq \mathbf{R}$; an important special case is where each S_i is an interval, in which case we have the usual geometric (axis-aligned) rectangles. The PAC learning of rectangles has been studied because they have been found experimentally to yield excellent hypotheses for a variety of applied learning problems (see [36, 11]). Also, pseudorandom sets for rectangles have been actively studied recently [28, 4, 13, 24, 10, 19, 6] because (i) they are a subproblem common to the derandomization of depth-2 (DNF) circuits and derandomizing Randomized Logspace (RL), and (ii) they approximate the distribution of n independent multivalued random variables. In this work, we present improved (and in some cases optimal) upper bounds for a class of such "approximating rectangles" problems in learning and derandomization.

(a) Learning from multiple-instance examples. We describe and analyze a new algorithm for a practical learning problem, motivated by drug discovery, introduced by Dietterich, Lathrop and Lozano-Perez [11]. Their problem boils down to that of learning an axis-parallel rectangle B in $\mathbf{R}^n$ from multi-instance examples. An r-instance example consists of r elements of $\mathbf{R}^n$, together with a label indicating whether any of the instances of this example are in B. The idea is that each multi-instance example represents a molecule and the instances represent different shapes of this molecule. The molecule "works" if at least one of its shapes can bind to some site. This is possible if the measurements of the shape are in the target region B. After receiving a sample of multi-instance examples the learning algorithm is supposed to output a hypothesis $H \subseteq \mathbf{R}^n$ which is close to the target rectangle B, in the sense that it is likely to correctly classify another r-instance example as to whether any of its instances are in B.

This problem has previously been studied in Valiant's PAC framework [34]. In [25], it was proved that if all instances are drawn independently from a product distribution on $\mathbf{R}^n$, the target rectangle can be learned from r-instance examples in $O\left(\frac{n^5 r^{12}}{\epsilon^{20}} \log^2 \frac{nr}{\epsilon\delta}\right)$ time, where ϵ and δ are accuracy and confidence parameters.

In this paper we present a new learning algorithm which does not require that the distribution on $\mathbf{R}^n$ is a product distribution and which takes only $O\left(\frac{n^3 r^2}{\epsilon^2} \log \frac{nr\log(1/\delta)}{\epsilon} \log \frac{n}{\delta}\right)$ time. This algorithm can be modified slightly to achieve similar results in the statistical query model; applying the

[†] Supported by Esprit NeuroCOLT project No. 8556. Address: Institute for Theoretical Computer Science, Technische Universität Graz, Klosterwiesgasse 32/2, A-8010 Graz, Austria. Email: pauer@igi.tu-graz.ac.at.

[‡] Supported by National University of Singapore Academic Research Fund Grant RP960625. Some of the work reported in this paper was done while this author was at Duke University supported by Office of Naval Research grant N00014–94–1–0938. Address: Dept. of Information Systems and Computer Science, National University of Singapore, Singapore 119260, Republic of Singapore. Email: plong@iscs.nus.sg.

[§] Address: Dept. of Information Systems and Computer Science, National University of Singapore, Singapore 119260, Republic of Singapore. Supported in part by National University of Singapore Academic Research Fund Grants RP950662 and RP960620. Email: aravind@iscs.nus.sg.

ISSN 1433-8092

results of Kearns [20], this implies that it can be made robust against classification noise. Our algorithm is substantially different from those previously proposed for this problem [11, 25]. We believe that a variant of our algorithm will prove useful in practice. Initial empirical results [7] support this belief: a straightforward implementation of a variant of our algorithm performs competitively on datasets used in [11].

Our analysis still requires that all instances are drawn independently. We point out that r-instance examples generated by an arbitrary distribution on $(\mathbf{R}^n)^r$ yield a much harder learning problem. If rectangles could be learned in this model from multi-instance examples then DNF formulas could be learned in the original PAC model, a longstanding open problem. Furthermore, we show that a polynomial-time learning algorithm which outputs a rectangle as its hypothesis only exists if $\mathcal{NP} = \mathcal{RP}$.

(b) Learning from single-instance examples. We show that the "closure algorithm" [18], which takes time linear in the size of the sample, PAC learns axis-aligned rectangles in $\mathbf{R}^n$ in the original (one-instance) PAC model from $O\left(\frac{n+\log\frac{1}{\delta}}{\epsilon}\right)$ examples. This matches the lower bound of [12] to within a constant factor. This is the first example we know of an infinite concept class used in practice whose PAC learning sample complexity has been determined to within a constant factor. Our bound improves on the $O\left(\frac{n\log\frac{1}{\epsilon}+\log\frac{1}{\delta}}{\epsilon}\right)$ bound that follows from the general results of Blumer, Ehrenfeucht, Haussler and Warmuth [8], and on the bound of $O\left(\frac{n\log\frac{1}{\delta}}{\epsilon}\right)$ that follows from the general results of Haussler, Littlestone and Warmuth [18].

In our analysis, we bound the p-norm of the error of the algorithm's hypothesis as a function of the random sample it receives, where $p = \ln(1/\delta)$. A side effect of our analysis is that for all p, after m examples, this norm is at most $\frac{n+p}{m}$. The bound of [18] was obtained by analyzing the expected error (i.e. the 1-norm).

(c) Pseudorandom sets for combinatorial rectangles. One major goal in derandomization is to efficiently construct a discrete structure (e.g., constant-degree expanders (Lubotzky, Phillips and Sarnak[26]), dispersers and extractors (Nisan [29]), hash function families (Carter and Wegman [9])) that is usually easily shown to exist by a probabilistic argument. Converting randomized algorithms to deterministic ones is one of the many applications of such results. Another application is that a random construction (notably of hashing families and error-correcting codes) could require enormous storage, thus requiring a succinct explicit construction. This area was enriched by the key observation of Naor and Naor that randomized algorithms are usually robust to approximating the distribution of n i.i.d. unbiased random bits $X_1, X_2, \ldots, X_n$ [28]. A natural generalization of this is to allow the X_i to have arbitrary independent distributions on any finite set, and is formalized as the following problem of pseudorandom sets for combinatorial rectangles. Let $\mathcal{R}_m^n = \{S_1 \times \cdots \times S_n : \forall i,\ S_i \subseteq \{0, 1, \ldots, m-1\}\}$ be a set of n-dimensional combinatorial rectangles. Call a finite multiset $S \subseteq [0, m-1]^n$ an ϵ-approximation for $\mathcal{R}_m^n$, if for $\vec{X}$ sampled uniformly at random from S, we have for all $R = S_1 \times \cdots \times S_n \in \mathcal{R}_m^n$ that $|\Pr(\vec{X} \in R) - (\prod_i |S_i|)/m^n| \leq \epsilon$. That is, w.r.t. events in $\mathcal{R}_m^n$, a random sample from S should look roughly like a random sample from $[0, m-1]^n$.

The goal here is to construct a "small" such "pseudorandom" set S deterministically. (A multi-set of $O(mn/\epsilon)$ points chosen at random from $[0, m-1]^n$, forms an ϵ-approximation with high probability.) Call an ϵ-approximation S for $\mathcal{R}_m^n$ indexible if, given $\log_2 |S|$ bits chosen independently and uniformly at random, we can generate any desired bit of a uniformly random sample from S deterministically in time polylog$(m + n + |S|)$. The major open question here is to explicitly construct an indexible ϵ-sample S such that $\log|S| = O(\log m + \log n + \log(1/\epsilon))$. Progress on this has been made very recently in Armoni, Saks, Wigderson & Zhou [6], constructing an indexible S with $\log|S| = O(\log m + \log n + \log^2(1/\epsilon))$. Hence, the key problem is to improve the $O(\log^2(1/\epsilon))$ term, to the eventual target of $O(\log(1/\epsilon))$. Since this is an improvement in the number of random bits used, running-time improvement for the corresponding derandomized algorithms would be significant: poly$((1/\epsilon)^{\log(1/\epsilon)})$ will become poly$(1/\epsilon)$.

Another related direction for progress is as follows. Given $R = S_1 \times S_2 \times \cdots \times S_n \in \mathcal{R}_m^n$, we shall say R is trivial in dimension i iff $S_i = \{0, 1, \ldots, m-1\}$; R is nontrivial in dimension i otherwise. Given any integer $k \leq n$, let $\mathcal{R}_{m,k}^n$ contain the elements of $\mathcal{R}_m^n$ that are nontrivial in at most k dimensions. In analogy with k-wise and almost k-wise independence ([27, 3, 28, 4]), it is also an important open question to construct a good ϵ-approximation S_0 for $\mathcal{R}_{m,k}^n$ (note that $\mathcal{R}_m^n = \mathcal{R}_{m,n}^n$). "Indexibility" here is that given $\log_2|S_0|$ bits chosen independently and uniformly at random, we can generate any desired bit of a uniformly random sample from S_0 deterministically in time polylog$(m + k + \log n + |S_0|)$. The major goal here is to achieve $\log|S_0| = O(\log\log n + \log k + \log m + \log(1/\epsilon))$. This would be optimal to within a constant factor, and by setting $k = n$, we would also get the required construction for $\mathcal{R}_m^n$.

We first show how to convert the construction of [6] to get an indexible ϵ-approximation S for $\mathcal{R}_{m,k}^n$ with $\log|S| = O(\log\log n + \log k + \log m + \log^2(1/\epsilon))$. We then use some previous results with new ideas to construct an indexible ϵ-approximation S where

$$\log|S| = O(\log\log n + (\log m)\log(1/\epsilon) \\ + \log(1/\epsilon)\log(\lceil k/\log(1/\epsilon)\rceil))$$

To parse this, we note that this is better than the above-seen construction of [6] iff $\epsilon \leq k^{-c_1}$ and $\epsilon \leq m^{-c_2}$, for certain absolute constants $c_1, c_2 > 0$, i.e., if ϵ is sufficiently small. Often, ϵ is in fact as small as $\exp(-\Theta(k))$, e.g., in many applications of almost k-wise independent random bits. Thus, the main determining factor for comparing the construction of [6] with ours seems to be whether $\epsilon \leq m^{-c_2}$ or not. If ϵ is "small", e.g., if m is $O(\text{polylog}(1/\epsilon))$, then we get good improvements.

Of equal interest is the fact that our approach suggests a potential way of improving on it: Theorem 18 shows, e.g., that if there is an indexible ϵ-approximation S' for $\mathcal{R}_{m,k}^n$ with $\log|S'| = O(\log\log n + k + \log m + \log(1/\epsilon))$ (note that we just ask for linear dependence on k, while logarithmic dependence on k can be shown existentially), then we can improve our S to an S'', which satisfies

$$\log|S''| = O(\log\log n + \log k + \log m + \log(1/\epsilon) \\ + \log(1/\epsilon)\log(\lceil k/\log(1/\epsilon)\rceil)).$$

We see this as particularly promising, since: (i) $\log|S'|$ is allowed to be quite high as a function of k, and, indeed, such a construction has been achieved for axis-aligned rectangles [13, 10], and (ii) $\log|S''|$ would be optimal for the common situation of $k = O(\log(1/\epsilon))$. Even if k is, say, $O(\text{polylog}(1/\epsilon))$, we would have $\log|S''| = O(\log\log n + \log m + \log(1/\epsilon)\log\log(1/\epsilon))$, a significant improvement over the $O(\log\log n + \log m + \log^2(1/\epsilon))$ that we derive above from [6].

(d) Pseudorandom sets for axis-aligned rectangles, and constructions of Ramsey-type graphs. Call an undirected graph G an (s, t, n)-graph iff it has n vertices, has clique number $\omega(G) \leq s$, and independence number $\alpha(G) \leq t$. One of the first applications of the probabilistic method was the proof by Erdős that $(2\log_2 n, 2\log_2 n, n)$-graphs (also known as Ramsey graphs since they provide lower bounds for the graph Ramsey function) exist [14]. It is still an outstanding open question to explicitly construct such graphs; the current-best are the breakthroughs of Frankl and Wilson, who constructed $(2^{O(\sqrt{\log n \log\log n})}, 2^{O(\sqrt{\log n \log\log n})}, n)$-graphs [16, 17]. Similarly, while nonconstructive progress has been made on the ("off-diagonal" or "Ramsey-type") case of $s \neq t$ [5], where we assume w.l.o.g. that $s < t$, very few constructive results are known; see, e.g., [2] for a construction of $(2, O(n^{2/3}), n)$-graphs. (It is known that $(2, \Theta(\sqrt{n\log n}), n)$-graphs exist.)

We present a family of improved deterministic sequential and parallel (EREW PRAM) constructions for Ramsey-type graphs here, using ϵ-approximations for axis-aligned rectangles in $\{0, 1, \ldots, m-1\}^n$. For instance, we show parallel constructions of, e.g., $(2^{O(\sqrt{\log n})}, 2^{O(\sqrt{\log n})}, n)$-graphs using $n^{O((\log\log n)^2)}$ processors and polylog(n) time. (If we are willing to expend $n^{O(\log n)}$ processors, it is known that we can use "almost $(2\log^2 n)$-wise independent random bits" [28] to construct $(2\log_2 n, 2\log_2 n, n)$-graphs in polylog$(n)$ time.) At the other end of the spectrum, we show, e.g., that for arbitrarily small constants $\alpha, \beta > 0$, we can construct (c, n^α, n)-graphs using $\exp(n^\beta)$ processors and $O(n^\alpha)$ time, where $c = c(\alpha, \beta)$ is a constant. (Direct application of previous techniques such as "(almost) $n^{\Theta(1)}$-wise independent random variables" will have the processor complexity necessarily of the form $\exp(n^{f(\alpha)})$ for some function f, and not $\exp(n^\beta)$ for an arbitrarily small constant $\beta > 0$, as we have here.) To our knowledge, even the sequential counterparts of these parallel algorithms have not been presented before.

2 Definitions

Denote the reals by $\mathbf{R}$ and the positive integers by $\mathbf{N}$. We sometimes abbreviate "random variable" as "r.v.". We use the unit cost RAM model of computation.

For each $n \in \mathbf{N}$, for each $\vec{a}, \vec{b} \in \mathbf{R}^n$, let $B_{\vec{a},\vec{b}} = \prod_{i=1}^n [a_i, b_i]$, and let $B_{\vec{b}} = \prod_{i=1}^n (-\infty, b_i]$. Define $\text{BOXES}_n = \{B_{\vec{a},\vec{b}} : \vec{a}, \vec{b} \in \mathbf{R}^n\}$, $\text{BOXES}_{n,m} = \{B \cap \{0, ..., m-1\}^n : B \in \text{BOXES}_n\}$, and $\text{OBOXES}_n = \{B_{\vec{b}} : \vec{b} \in \mathbf{R}^n\}$. Finally, a class of combinatorial rectangles that we work with:

$$\mathcal{R}_m^n = \{\prod_{i=1}^n X_i : X_1, ..., X_n \subseteq \{0, ..., m-1\}\}.$$

3 Learning from multiple-instance examples

First we give a formal description of the learning problem.

Let $n, r \in \mathbf{N}$. A tuple $((\vec{x}_1, \ldots, \vec{x}_r), y)$ with each $\vec{x}_i \in \mathbf{R}^n$ and with $y \in \{0, 1\}$ is called an r-instance example. A sample is a sequence of r-instance examples. For a finite sequence $\sigma = \langle\langle \vec{x}_{s,i}\rangle_{i=1}^r\rangle_{s=1}^\ell$ of instances and a rectangle B, the sample generated by σ and B is

$$S_{B,\sigma} = ((\vec{x}_{1,1}, \ldots, \vec{x}_{1,r}), y_1), \ldots, ((\vec{x}_{\ell,1}, \ldots, \vec{x}_{\ell,r}), y_\ell),$$

where

$$y_s = \psi_B(\vec{x}_{s,1}, \ldots, \vec{x}_{s,r}) = \begin{cases} 1 & \text{if} \quad \exists i \in \{1, \ldots, r\} : \vec{x}_{s,i} \in B \\ 0 & \text{if} \quad \forall i \in \{1, \ldots, r\} : \vec{x}_{s,i} \notin B \end{cases}$$

for $s = 1, \ldots, \ell$.

A learning algorithm receives a sample S and an accuracy parameter ϵ and outputs a hypothesis $H(S, \epsilon) \subseteq \mathbf{R}^n$. The error of a hypothesis H in respect to a probability distribution D on $\mathbf{R}^n$ is measured by the probability that a random r-instance is misclassified, i.e. by $\mathbf{er}_{B,D}(H) = D^r\{(\vec{x}_1, \ldots, \vec{x}_r) : \psi_B(\vec{x}_1, \ldots, \vec{x}_r) \neq \psi_H(\vec{x}_1, \ldots, \vec{x}_r)\}$. Here and elsewhere D^r denotes the distribution obtained by sampling r times independently from D.

Definition 1 A learning algorithm learns BOXES_n from r-instance examples with sample complexity $\ell(n, r, \epsilon, \delta)$ if the learning algorithm calculates a hypothesis $H(S, \epsilon)$ such that for all $B \in \text{BOXES}_n$, for all distributions D on $\mathbf{R}^n$, for all $\epsilon, \delta > 0$, for all $\ell \geq \ell(n, r, \epsilon, \delta)$

$$D^{r\cdot\ell}\{\sigma : \mathbf{er}_{B,D}(H(S_{B,\sigma}, \epsilon)) \geq \epsilon\} < \delta.$$

The following is the main result of this section.

Theorem 2 There is a learning algorithm that learns BOXES_n from r-instance examples with sample complexity $\ell = O\left(\frac{n^2r^2}{\epsilon^2}\log\frac{n}{\delta}\right)$. The run-time of the algorithm is $O(n\ell\log\ell)$.

3.1 The algorithm

First, observe that $a_k \leq x_k \leq b_k \Leftrightarrow -x_k \leq -a_k \wedge x_k \leq b_k$. Thus BOXES_n can be learned if OBOXES_{2n} can be learned. (A similar trick was employed in [21, 23].) We therefore give an algorithm for learning OBOXES_n for each n.

The algorithm for learning OBOXES_n learns the faces $b_1, ..., b_n$ of the target $B = \prod_{k=1}^n(-\infty, b_k]$ independently. It tries to calculate an estimate $\hat{b}_k \leq b_k$ such that $D\{(x_1, \ldots, x_n) : (x_1, \ldots, x_n) \in B \text{ and } x_k > \hat{b}_k\}$ is small. If this can be done for all k then the hypothesis $\prod_{k=1}^n(-\infty, \hat{b}_k]$ is contained in the target rectangle and its error is small. Now the main observation is that, despite the fact that examples do not indicate whether individual instances lie in B, we can still estimate $\beta_k(b) = D\{(x_1, \ldots, x_n) : (x_1, \ldots, x_n) \in B \text{ and } x_k > b\}$ from the sample. Thus the algorithm just has to pick a $\hat{b}_k$ such that the estimate $\hat{\beta}_k(\hat{b}_k)$ is small while still guaranteeing that $\hat{b}_k \leq b_k$. Let $q = [1 - D(B)]^r$ (note that this is just the probability that an r-instance example is labelled 0) and $\alpha_k(b) = D\{(x_1, \ldots, x_r) : x_k > b\}$ so that if we define $\varphi_k(b) = D^r\{(\vec{x}_1, \ldots, \vec{x}_r) : \psi_B(\vec{x}_1, \ldots, \vec{x}_r) = 0 \text{ and } x_{1,k} > b\}$, then

$$\varphi_k(b) = [\alpha_k(b) - \beta_k(b)] \cdot [1 - D(B)]^{r-1} = [\alpha_k(b) - \beta_k(b)] \cdot q^{1-1/r}. \tag{1}$$

Since q, α_k, and φ_k can be estimated from a sample S, we get an estimate for β_k by solving (1) for β_k: we set $\hat{q} := |\{1 \leq s \leq \ell : y_s = 0\}|/\ell$, $\hat{\alpha}_k(b) := |\{1 \leq s \leq \ell : x_{s,1,k} > b\}|/\ell$, $\hat{\varphi}_k(b) := |\{1 \leq s \leq \ell : x_{s,1,k} > b \text{ and } y_s = 0\}|/\ell$, and $\hat{\beta}_k(b) := \hat{\alpha}_k(b) - \frac{\hat{\varphi}_k(b)}{\hat{q}^{1-1/r}}$. Finally, observe that $\mathbf{er}_{B,D}(\mathbf{R}^n) < \epsilon$ if $q < \epsilon$ and $\mathbf{er}_{B,D}(\emptyset) < \epsilon$ if $q > 1 - \epsilon$. We get the following algorithm.

```
Input: a sample $S$ of size $\ell$ and an accuracy parameter $\epsilon$.
If $\hat{q} < \epsilon/2$ then return $\mathbf{R}^n$ and halt, and if $\hat{q} > 1 - \epsilon/2$ then return $\emptyset$ and halt.
For $k = 1, \ldots, n$ do
    $\hat{b}_k := \min\{x_{s,1,k} : 1 \leq s \leq \ell \text{ and } \hat{\beta}(x_{s,1,k}) \leq \frac{\epsilon}{16nr\hat{q}^{1-1/r}}\}$.
Return $B_{(\hat{b}_1, \ldots, \hat{b}_n)}$.
```

For each k the $\hat{\beta}_k(x_{s,1,k})$ can be calculated incrementally after sorting the $x_{s,1,k}$. Thus the run-time of the algorithm is $O(n\ell \log \ell)$.

3.2 Analysis of the algorithm

It remains to show that the estimate $\hat{b}_k$ is sufficiently accurate that the error of the hypothesis of the algorithm is small enough.

The next lemma gives a sufficient condition on the accuracy of the estimates $\hat{q}$, $\hat{\alpha}_k$, and $\hat{\varphi}_k$. Theorem 2 is then proved by applying uniform convergence bounds.

Lemma 3 Let $0 < \epsilon < 1/2$ and let H be the hypothesis of the algorithm based on a sample $S_{B,\sigma}$. If for all $k \in \{1, \ldots, n\}$ and all $b \in \mathbf{R}$

$$\max\{|q - \hat{q}|, |\alpha_k(b) - \hat{\alpha}_k(b)|, |\varphi_k(b) - \hat{\varphi}_k(b)|\} \leq \frac{\epsilon}{256nr}, \tag{2}$$

then $\mathbf{er}_{B,D}(H) < \epsilon$.

Proof: We claim that for the non-trivial case $\frac{\epsilon}{2} \leq \hat{q} \leq 1 - \frac{\epsilon}{2}$, the following inequalities

$$\frac{\epsilon}{4} \leq q \leq 1 - \frac{\epsilon}{4}, \tag{3}$$

$$q^{1-1/r}/2 \leq \hat{q}^{1-1/r} \leq 2q^{1-1/r}, \tag{4}$$

$$|\beta_k(b) - \hat{\beta}_k(b)| \leq \frac{\epsilon}{32nrq^{1-1/r}} \tag{5}$$

hold for all k and b. Inequalities (3) and (4) are obvious; we now prove (5).

It is easy to show using calculus that for all $x, y > 0$

$$|x^{1-1/r} - y^{1-1/r}| \leq |x - y| \left(\frac{1}{\min\{x, y\}}\right)^{1/r}. \tag{6}$$

Fix k and b (and drop them from the subscripts of all variables). Expanding the definition of $\hat{\beta}$ and applying (1), we have

$$|\beta - \hat{\beta}| = \left| \left(\alpha - \frac{\varphi}{q^{1-1/r}}\right) - \left(\hat{\alpha} - \frac{\hat{\varphi}}{\hat{q}^{1-1/r}}\right) \right|.$$

Bounding $|\alpha - \hat{\alpha}|$ using (2) and simplifying the rest, we get

$$|\beta - \hat{\beta}| \leq \frac{\epsilon}{256rn} + \frac{|\varphi\hat{q}^{1-1/r} - \hat{\varphi}q^{1-1/r}|}{q^{1-1/r}\hat{q}^{1-1/r}}.$$

Applying (4), we get

$$\begin{aligned} |\beta - \hat{\beta}| &\leq \frac{\epsilon}{256rn} + \frac{2|\varphi\hat{q}^{1-1/r} - \hat{\varphi}q^{1-1/r}|}{q^{2-2/r}} \\ &= \frac{\epsilon}{256rn} + \frac{2|\varphi\hat{q}^{1-1/r} - (\varphi + (\hat{\varphi} - \varphi))q^{1-1/r}|}{q^{2-2/r}} \\ &\leq \frac{\epsilon}{256rn} + \frac{2|\hat{\varphi} - \varphi|}{q^{1-1/r}} + \frac{2|\varphi\hat{q}^{1-1/r} - \varphi q^{1-1/r}|}{q^{2-2/r}}. \end{aligned}$$

Applying (2) and the fact that $\varphi \leq q$, we have

$$|\beta - \hat{\beta}| \leq \frac{\epsilon}{256rn} + \frac{\epsilon}{128rnq^{1-1/r}} + \frac{2|\hat{q}^{1-1/r} - q^{1-1/r}|}{q^{1-2/r}}.$$

Applying (6), we get

$$|\beta - \hat{\beta}| \leq \frac{\epsilon}{256rn} + \frac{\epsilon}{128rnq^{1-1/r}} + \frac{2|\hat{q} - q|}{(\min\{q, \hat{q}\})^{1/r} q^{1-2/r}}. \tag{7}$$

Since $\hat{q} \in [\epsilon/2, 1 - \epsilon/2]$, (2) implies that $\hat{q} \geq q/2$, so (7) implies

$$|\beta - \hat{\beta}| \leq \frac{\epsilon}{256rn} + \frac{\epsilon}{128rnq^{1-1/r}} + \frac{4|\hat{q} - q|}{q^{1-1/r}}.$$

Applying (2) yields (5).

We claim that $\hat{b}_k \leq b_k$ and that

$$\beta_k(\hat{b}_k) \leq \frac{\epsilon}{4nrq^{1-1/r}}. \tag{8}$$

Observe that if $x_{s,1,k} > b_k$ for all $s \in \{1, \ldots, \ell\}$ then $\hat{\alpha}_k(b_k) = 1$ whereas $\alpha_k(b_k) \leq 1 - D(B) = q^{1/r} \leq (1 - \epsilon/4)^{1/r} \leq 1 - \frac{\epsilon}{4r}$ by (3) which contradicts (2). Hence there are some $x_{s,1,k} \leq b_k$. Let $c_k = \max\{x_{s,1,k} : 1 \leq s \leq \ell, x_{s,1,k} \leq b_k\}$. Then $\hat{\beta}_k(c_k) = \hat{\beta}_k(b_k) \leq \frac{\epsilon}{32nrq^{1-1/r}} \leq \frac{\epsilon}{16nr\hat{q}^{1-1/r}}$ by (5) and (4) since $\beta_k(b_k) = 0$. Thus $\hat{b}_k \leq b_k$. Furthermore,

$$\begin{aligned} \beta_k(\hat{b}_k) &\leq \hat{\beta}_k(\hat{b}_k) + \frac{\epsilon}{32nrq^{1-1/r}} \\ &\leq \frac{\epsilon}{16nr\hat{q}^{1-1/r}} + \frac{\epsilon}{32nrq^{1-1/r}} \\ &\leq \frac{\epsilon}{8nrq^{1-1/r}} + \frac{\epsilon}{32nrq^{1-1/r}} \\ &\leq \frac{\epsilon}{4nrq^{1-1/r}}, \end{aligned}$$

again by (5) and (4) and the choice of $\hat{b}_k$ in the algorithm.

Finally an r-instance is misclassified by $H = \prod_{k=1}^{n}(-\infty, \hat{b}_k]$ only if no instance is in H and at least one instance is in

B, i.e. at least one coordinate of this instance is in $(\hat{b}_k, b_k]$ for the corresponding k. The probability of drawing such an instance is at most $\beta_k(\hat{b}_k) - \beta_k(b_k) = \beta_k(\hat{b}_k)$. To bound the probability that a random instance is not in H we find

$$\begin{aligned} 1 - D(H) &\leq 1 - D(B) + \sum_{k=1}^{n} \beta_k(\hat{b}_k) \\ &\leq q^{1/r} + \frac{\epsilon}{4rq^{1-1/r}} \\ &\leq q^{1/r}(1 + 1/r) \end{aligned}$$

by (8) and (3). Hence

$$\begin{aligned} \mathbf{er}_{B,D}(H) &\leq r \sum_{k=1}^{n} \beta_k(\hat{b}_k) \cdot [1 - D(H)]^{r-1} \\ &\leq nr \frac{\epsilon}{4nrq^{1-1/r}} [q^{1/r}(1 + 1/r)]^{r-1} \\ &\leq \epsilon \end{aligned}$$

by (8). □

Now we turn to analyzing the number of examples required to ensure (2). This analysis uses standard techniques, but we include it in an appendix for completeness.

Lemma 4 There is a constant c such that for all $0 < \epsilon, \delta < 1/2$, if $\ell \geq \frac{cn^2r^2}{\epsilon^2} \log \frac{n}{\delta}$, then

$$\Pr(|q - \hat{q}| > \frac{\epsilon}{256nr}) \leq \frac{\delta}{n+1} \tag{9}$$

and for each $k \in \{1, ..., n\}$,

$$\begin{aligned} &\Pr(\exists b,\ \max\{|\alpha_k(b) - \hat{\alpha}_k(b)|, |\varphi_k(b) - \hat{\varphi}_k(b)|\} > \tfrac{\epsilon}{256nr}) \\ &\qquad \leq \tfrac{\delta}{n+1}. \end{aligned} \tag{10}$$

Proof: In Appendix A. □

Proof of Theorem 2: Combine Lemmas 3 and 4.

3.3 The hardness of learning from dependent multiple-instance examples

Definition 5 A learning algorithm learns BOXES$_n$ from dependent r-instance examples if the learning algorithm calculates a hypothesis $H(S, \epsilon)$ such that for all $B \in$ BOXES$_n$, for all distributions D on $(\mathbf{R}^n)^r$, for all $\epsilon, \delta > 0$, for all $\ell \geq \ell(n, r, \epsilon, \delta)$, $D^\ell\{\sigma : \mathbf{er}_{B,D}(H(S_{B,\sigma}, \epsilon)) \geq \epsilon\} < \delta$.

Theorem 6 If there is a poly(n,r,$1/\epsilon$,$1/\delta$)-time algorithm A for learning BOXES$_n$ from dependent r-instance examples, then there is a poly(n,r,$1/\epsilon$,$1/\delta$)-time algorithm A' for learning r-term DNF formulas over n variables (from 1-instance examples).

If A in addition outputs axis-aligned rectangles as its hypotheses then $\mathcal{RP} = \mathcal{NP}$.

Proof: We reduce learning an r-term DNF $f = C_1 + ... + C_r$ over n variables $x_1, ..., x_n$ to learning a rectangle in $\mathbf{R}^{nr}$ from r-instance examples. For each truth setting $\vec{v} \in \{0,1\}^n$, let $\varphi(\vec{v})$ be the r instances $(v_1, ..., v_n, 1/2, ..., 1/2), ..., (1/2, ..., 1/2, v_1, ..., v_n) \in \{0, 1/2, 1\}^{nr}$. We associate f with $B_{\vec{a},\vec{b}}$ where for each $0 \leq i < r, 1 \leq j \leq n$,

$$\begin{array}{ll} a_{in+j} = 1/2 \text{ and } b_{in+j} = 1 & \text{if } x_j \in C_{i+1} \\ a_{in+j} = 0 \text{ and } b_{in+j} = 1/2 & \text{if } \bar{x}_j \in C_{i+1} \\ a_{in+j} = 0, b_{in+j} = 1 & \text{otherwise.} \end{array}$$

Then $\varphi(\vec{v})$ is classified 1 by $B_{\vec{a},\vec{b}}$ iff $\vec{v}$ satisfies f.

Suppose A learns BOXES$_n$ from dependent r-instance examples in poly(n,r,$1/\epsilon$,$1/\delta$) time. Consider the DNF learning algorithm A' that, for each example $(\vec{v}, y)$, gives $(\varphi(\vec{v}), y)$ to A, and, given the hypothesis H_A output by A, constructs $H_{A'}$ by letting $\vec{v} \in H_{A'} \Leftrightarrow \varphi(\vec{v}) \in H_A$. It is easily verified (see [31]), that A' learns r-term DNF formulas over n variables in poly(n,r,$1/\epsilon$,$1/\delta$) time.

Next, we claim that if H_A labels collections of r instances according to an axis-aligned hyperrectangle, then $H_{A'}$ can be expressed as an r-term DNF. Applying the result of Pitt and Valiant [30], that r-term DNF are not learnable using r-term DNF as hypotheses in polynomial time unless $\mathcal{RP} = \mathcal{NP}$, will complete the proof of the second statement.

Suppose H_A is $\psi_{B_{(\hat{a}_1,...,\hat{a}_{rn}),(\hat{b}_1,...,\hat{b}_{rn})}}$.

- If there exist distinct $i, i' \in \{0, ..., r-1\}$ such that there are $j, j' \in \{1, ..., n\}$ with $1/2 \notin [\hat{a}_{in+j}, \hat{b}_{in+j}]$ and $1/2 \notin [\hat{a}_{i'n+j'}, \hat{b}_{i'n+j'}]$ then the definition of φ implies that $H_{A'} = \emptyset$, trivially expressed as an r-term DNF.
- If there is a single $i \in \{0, ..., r-1\}$ such that there exists j with $1/2 \notin [\hat{a}_{in+j}, \hat{b}_{in+j}]$, then $H_{A'}$ can be expressed by the single clause $\hat{C}$, where $x_j \in \hat{C} \Leftrightarrow 0 \notin [\hat{a}_{in+j}, \hat{b}_{in+j}]$ and $\bar{x}_j \in \hat{C} \Leftrightarrow 1 \notin [\hat{a}_{in+j}, \hat{b}_{in+j}]$.
- Otherwise, it is easily verified that $H_{A'}$ can be expressed as the r-term DNF $\hat{C}_1 + ... + \hat{C}_r$ obtained by including x_j in $\hat{C}_{i+1}$ iff $0 \notin [\hat{a}_{in+j}, \hat{b}_{in+j}]$ and including $\bar{x}_j$ in $\hat{C}_{i+1}$ iff $1 \notin [\hat{a}_{in+j}, \hat{b}_{in+j}]$.

□

4 Learning from single-instance examples

Valiant's PAC model can be obtained from the model described in Section 3 by fixing the number r of instances to be 1. The following is our main result about this model.

Theorem 7 There is a learning algorithm A such that, for all $\epsilon, \delta > 0, n \in \mathbf{N}$, Algorithm A (ϵ, δ)-learns BOXES$_n$ from $\frac{e(2n+\lceil \ln \frac{1}{\delta} \rceil)}{\epsilon}$ examples in $O\left(\frac{n(n+\ln \frac{1}{\delta})}{\epsilon}\right)$ time.

As discussed in Section 3, it is sufficient to consider OBOXES$_n$. Consider the algorithm A that, given a sample $((\vec{x}_1, y_1), ..., (\vec{x}_\ell, y_\ell))$, sets each $\hat{b}_k = \max\{x_{i,k} : y_i = 1\}$ and outputs $H = B_{(\hat{b}_1,...,\hat{b}_n)}$. This algorithm is known as the "closure algorithm", because it outputs the unique smallest element of OBOXES$_n$ consistent with the sample.

We begin by characterizing the pth moment of the error of A's hypothesis. For some sequence $\sigma = \vec{x}_1, ..., \vec{x}_\ell$, define $\mathbf{er}_{A,B,D}(\sigma)$ to be the error of A's hypothesis H when given $S_{B,\sigma}$, i.e. $\mathbf{er}_{B,D}(H(S_{B,\sigma}), \epsilon)$.

Lemma 8 Choose $n, \ell, p \in \mathbf{N}$, $B \in \mathrm{OBOXES}_n$ and a probability distribution D over $\mathbf{R}^n$. Then

$$\mathbf{E}_{\sigma \in D^\ell}((\mathbf{er}_{A,B,D}(\sigma))^p)$$

is equal to the probability, if

- we draw $\vec{x}_1, ..., \vec{x}_{\ell+p}$ independently at random according to D,
- give $S_{B,(\vec{x}_1,...,\vec{x}_\ell)}$ to A (only the first ℓ draws are used here), and
- call A's resulting hypothesis H,

that H is incorrect about each of $\vec{x}_{\ell+1}, ..., \vec{x}_{\ell+p}$. That is, that

$$\{\vec{x}_{\ell+1}, ..., \vec{x}_{\ell+p}\} \subseteq H \Delta B,$$

where Δ denotes the symmetric difference.

Proof: For each $\sigma \in (\mathbf{R}^n)^\ell$, define $H_\sigma = H(S_{B,\sigma})$. Expanding the definition yields

$$\mathbf{E}_{\sigma \in D^\ell}((\mathbf{er}_{A,B,D}(\sigma))^p) = \int (\Pr_{\vec{z} \in D}(\vec{z} \in H_\sigma \Delta B))^p \, dD^\ell(\sigma).$$

Changing the names of the variables in the p factors of $(\Pr_{\vec{z} \in D}(\vec{z} \in H_\sigma \Delta B))^p$ from $\vec{z}$ to $\vec{x}_{\ell+1}, ..., \vec{x}_{\ell+p}$ respectively, we get

$$\begin{aligned}&\mathbf{E}_{\sigma \in D^\ell}((\mathbf{er}_{A,B,D}(\sigma))^p) \\ &= \int \prod_{i=1}^p \Pr_{\vec{x}_{\ell+i} \in D}(\vec{x}_{\ell+i} \in H_\sigma \Delta B) \, dD^\ell(\sigma)\end{aligned}$$

which immediately implies

$$\begin{aligned}&\mathbf{E}_{\sigma \in D^\ell}((\mathbf{er}_{A,B,D}(\sigma))^p) \\ &= \int \prod_{i=1}^p \Pr_{\vec{x}_{\ell+1},...,\vec{x}_{\ell+p} \in D^p}(\vec{x}_{\ell+i} \in H_\sigma \Delta B) \, dD^\ell(\sigma).\end{aligned}$$

Applying the definition of independence, we get

$$\begin{aligned}&\mathbf{E}_{\sigma \in D^\ell}((\mathbf{er}_{A,B,D}(\sigma))^p) \\ &= \int \Pr_{\vec{x}_{\ell+1},...,\vec{x}_{\ell+p} \in D^p}(\textstyle\bigwedge_{i=1}^p \vec{x}_{\ell+i} \in H_\sigma \Delta B) \, dD^\ell(\sigma).\end{aligned} \tag{11}$$

Define $\varphi : (\mathbf{R}^n)^{\ell+p} \to \{0, 1\}$ by

$$\varphi(\vec{x}_1, ..., \vec{x}_{\ell+p}) = \begin{cases} 1 & \text{if } \bigwedge_{i=1}^p \vec{x}_{\ell+i} \in H_{(\vec{x}_1,...,\vec{x}_\ell)} \Delta B \\ 0 & \text{otherwise.} \end{cases}$$

Then rewriting (11), we get

$$\begin{aligned}&\mathbf{E}_{\sigma \in D^\ell}((\mathbf{er}_{A,B,D}(\sigma))^p) \\ &= \int \int \varphi(\vec{x}_1, ..., \vec{x}_{\ell+p}) \, dD^p(\vec{x}_{\ell+1}, ..., \vec{x}_{\ell+p}) \, dD^\ell(\vec{x}_1, ..., \vec{x}_\ell).\end{aligned}$$

Applying Fubini's Theorem (see [15, volume 2, page 120]) completes the proof. □

The proof of Lemma 8 did not use anything specific about A or OBOXES_n; therefore, the lemma can trivially be generalized to any algorithm and concept class.

Next, we record a well-known lemma whose application is commonly known as the "permutation trick".

Lemma 9 (see [18]) Choose a set X, $m \in \mathbf{N}$, a distribution D on X, and a random variable φ defined on X^m. Let U be the uniform distribution on the permutations of $\{1, ..., m\}$. Then

$$\int \varphi(x) D^m(x) \le \sup_{(x_1,...,x_m) \in X^m} \int \varphi(x_{\sigma(1)}, ..., x_{\sigma(m)}) U(\sigma).$$

Now we are ready to bound the pth moment of the error.

Lemma 10 Choose $n, p, \ell \in \mathbf{N}$. For Algorithm A (the Closure Algorithm), for any $B \in \mathrm{OBOXES}_n$, and for any probability distribution D over $\mathbf{R}^n$, $\mathbf{E}_{\sigma \in D^\ell}((\mathbf{er}_{A,B,D}(\sigma))^p) \le \frac{\binom{n+p-1}{p}}{\binom{\ell+p}{p}}$.

Proof: Lemma 8 implies that $\mathbf{E}_{\sigma \in D^\ell}((\mathbf{er}_{A,B,D}(\sigma))^p)$ is equal to the probability, if (a) we draw $\vec{x}_1, ..., \vec{x}_{\ell+p}$ independently at random from D (b) give $S_{B,(\vec{x}_1,...,\vec{x}_\ell)}$ to A (only the first ℓ draws are used here), and (c) call the resulting hypothesis H, that each of $\vec{x}_{\ell+1}, ..., \vec{x}_{\ell+p}$ fall in the symmetric difference of H and B.

Applying Lemma 9, the above probability is at most the supremum, over $\vec{x}_1, ..., \vec{x}_{\ell+p}$, of the probability of the same event with respect to a random permutation of this particular sequence. Since the hypothesis H does not depend on the relative order of the first ℓ elements, and the "test" does not depend on the order of the last p elements, we can instead evaluate the probability of the same event with respect to a random choice of which p elements occur last.

Let us call a set of p elements, which, if occurring last, all fall in $B \Delta H$, a "bad set". We claim that there is a mapping ψ from the set of sequences of n nonnegative integers summing to p onto the set of bad sets. The fact that there are known to be only $\binom{n+p-1}{p}$ such sequences will then complete the proof.

Since the Closure Algorithm outputs the unique smallest hypothesis (call it $\prod_k (-\infty, \hat{b}_k]$) containing the examples given to it that are in B, each element $\vec{x}$ of a bad set must be in $B - \prod_k (-\infty, \hat{b}_k]$. For convenience, we "blame" $\vec{x}$'s misclassification on the least k such that $x_k > \hat{b}_k$.

Define a map ψ from the set of sequences $\vec{i}$ of n nonnegative integers summing to p to subsets of $\{1, ..., \ell + p\}$ by the following procedure.

```
$T := \{t : \vec{x}_t \in B\}$;
$U := \emptyset$;
for $k := 1$ to $n$ do
        move the $i_k$ elements $t$ of $T$ with the largest values
            of $x_{t,k}$ from $T$ to $U$;
output $U$;
```

Choose a bad subset S. Let $\prod(-\infty, \hat{b}_k]$ be the hypothesis output by the closure algorithm when given the sample generated from the instances with indices not in S. For each $k \in \{1, ..., n\}$, define C_k to be the those elements of S whose misclassification was blamed on dimension k. We claim that $S = \psi(|C_1|, ..., |C_n|)$.

Imagine a run of the procedure defining ψ with its input $\vec{i}$ set to $(|C_1|, ..., |C_n|)$. For each $k \in \{1, ..., n\}$, let T_k be the value of T before the kth time through the loop. Define U_k similarly for U.

We wish to show that, each time through the loop, the elements of C_k are moved from T to U. We prove this by induction by proving the nominally stronger statement that for all k, $U_k = \cup_{j<k} C_j$. The base case is trivial. Assume the IH holds before some time k through the loop. Since $C_1, ..., C_n$ are disjoint, the inductive hypothesis implies that none of the elements of C_k have been moved from T to U before the kth time through the loop.

We claim that the elements of C_k are the elements of T_k with the largest kth components. Assume without loss of generality that $C_k \neq \emptyset$ and $C_k \neq T_k$. Choose $s \in C_k, t \in T_k - C_k$.

- If $t \notin S$, then $\vec{x}_t$ was one of the elements of B given to the closure algorithm, and therefore, $x_{t,k} \leq \hat{b}_k$.
- Suppose $t \in S$. By the inductive hypothesis, t was not blamed on an index less than k, since otherwise it would have been moved to U before the kth time through the loop. Therefore, t must be blamed on an index greater than k. Thus $x_{t,k} \leq \hat{b}_k$.

So in either case, $x_{t,k} \leq \hat{b}_k$. But the fact that index k was blamed for the misclassification of $\vec{x}_s$ implies that $x_{s,k} > \hat{b}_k$, and therefore, that $x_{t,k} < x_{s,k}$. Since s and t were chosen arbitrarily, the elements of C_k are the elements t of T_k with the largest values of $x_{t,k}$. Thus, they are the elements moved from T to U during the kth time through the loop, completing the proof of the inductive step. This implies that the output of the algorithm is $\cup_{k=1}^{d} C_k = S$. Since S was chosen arbitrarily, this implies that ψ is onto. As described above, this completes the proof. □

Proof of Theorem 7: Applying Markov's inequality together with Lemma 10 implies that

$$\begin{aligned} \Pr_{\kappa \in D^\ell}(\mathbf{er}_{A,B,D}(\kappa) > \epsilon) &= \Pr_{\kappa \in D^\ell}((\mathbf{er}_{A,B,D}(\kappa))^p > \epsilon^p) \\ &\leq \left(\frac{n+p}{\epsilon\ell}\right)^p \\ &\leq \delta \end{aligned}$$

for $p = \lceil \ln(1/\delta) \rceil$ and $\ell \geq \frac{e\left(n+\lceil \ln \frac{1}{\delta} \rceil\right)}{\epsilon}$. The fact that learning BOXES$_n$ reduces to learning OBOXES$_{2n}$ completes the proof. □

5 Pseudorandom sets for combinatorial rectangles

We refer the reader to the introduction for the motivation, notation, and history of this problem.

We will make use of an approximation result for BOXES. Let BOXES$_{k,n,m} \subseteq$ BOXES$_{n,m}$ be the set of axis-parallel rectangles in $\{0, 1, \ldots, m-1\}^n$ that are nontrivial in at most k dimensions, analogously to $\mathcal{R}^n_{m,k}$. Once again, BOXES$_{n,n,m}$ = BOXES$_{n,m}$. Furthermore, ϵ-approximations for BOXES$_{k,n,m}$ are defined analogously to ϵ-approximations of $\mathcal{R}^n_{m,k}$. An explicit family $\{S_{m,k,n,\epsilon} \subseteq [0, m-1]^n : k, m, n \in \mathbf{N}, k \leq n\}$, where $S_{m,k,n,\epsilon}$ is an indexible ϵ-approximation for BOXES$_{k,n,m}$, was presented in the full version of [10] with

$$\begin{aligned} \log |S_{m,k,n,\epsilon}| = O(&\log\log n + \log k + \log(1/\epsilon) \\ &+ \log(1/\epsilon)\log(\lceil k/\log(1/\epsilon)\rceil)). \end{aligned} \tag{12}$$

This builds on some ideas from [13], and improves on all three constructions of [13]. (Though it may look surprising that the bound of (12) is independent of m, it is shown in [13] that in the case of axis-parallel rectangles, we can effectively reduce the problem to the case where $m \leq \lceil 4k/\epsilon \rceil$ and where ϵ is replaced by $\epsilon/2$; hence the independence from m.)

In the following we first show how ϵ-approximations for $\mathcal{R}^{n'}_m$ easily lead to ϵ-approximations for $\mathcal{R}^n_{m,k}$ when $n' \geq 2k^2/\epsilon$ (Section 5.1). Then we show our main ϵ-approximation construction for $\mathcal{R}^n_{m,k}$ (Section 5.2).

5.1 Reducing n

Theorem 11 Let $k \leq n$ be any positive integers, $\epsilon \in (0,1)$, and $n' = \lceil 2k^2/\epsilon \rceil$. Suppose S' is an explicit indexible $(\epsilon/2)$-approximation for $\mathcal{R}^{n'}_m$. Then we can explicitly construct an indexible ϵ-approximation S'' for $\mathcal{R}^n_{m,k}$, with $\log|S''| = \log|S'| + O(\log\log n + \log k + \log(1/\epsilon))$.

We start with a simple hashing lemma.

Lemma 12 Suppose k and $n \geq k$ are positive integers. Let $\delta \in (0,1)$, $\ell = \lceil k^2/\delta \rceil$, and $\rho = \delta/(k^2\ell)$. Suppose $\vec{Y} = (Y_1, Y_2, \ldots, Y_n)$ is sampled uniformly at random from $S_{\ell,2,n,\rho}$. Then for any $\{a_1, a_2, \ldots, a_s\} \subseteq [n]$ with $s \leq k$, $\Pr(Y_{a_1}, Y_{a_2}, \ldots Y_{a_s}$ are all pairwise distinct$) \geq 1 - \delta$.

Proof: By the definition of $S_{\ell,2,n,\rho}$ we have, for any $1 \leq i < j \leq s$ and for any $p \in \{0, 1, \ldots, \ell - 1\}$, that $\Pr(Y_{a_i} = Y_{a_j} = p) \leq 1/\ell^2 + \rho$. Thus, $\Pr(Y_{a_i} = Y_{a_j}) \leq 1/\ell + \ell\rho = 1/\ell + \delta/k^2$. Hence,

$$\begin{aligned} &\Pr(Y_{a_1}, Y_{a_2}, \ldots Y_{a_s} \text{ not all distinct}) \\ &\qquad \leq \textstyle\sum_{i<j} \Pr(Y_{a_i} = Y_{a_j}) < (k^2/2)(1/\ell + \delta/k^2), \end{aligned}$$

which is at most δ by our choice of ℓ. □

Proof of Theorem 11: We first describe S'' by saying how to generate a uniformly random sample $\vec{X} = (X_1, X_2, \ldots, X_n)$ from it. Let $\delta = \epsilon/2$ and $\rho = \delta/(k^2 n')$. Sample $\vec{Y} = (Y_1, Y_2, \ldots, Y_n)$ uniformly at random from $S_{n',2,n,\rho}$, and define $f(i) = Y_i + 1$, for each $i \in [n]$; note that $f(i) \in [n']$. Independent of this random choice, sample $\vec{Z} = (Z_1, Z_2, \ldots, Z_{n'})$ uniformly at random from S', and define the final desired sample $\vec{X}$ to be $(Z_{f(1)}, Z_{f(2)}, \ldots, Z_{f(n)})$. S'' is indexible, since $S_{n',2,n,\rho}$ and S' are. Clearly, $\log|S''| = \log|S'| + O(\log\log n + \log k + \log(1/\epsilon))$.

We now show that for any $R = S_1 \times S_2 \times \cdots \times S_n \in \mathcal{R}^n_{m,k}$, $|\Pr(\vec{X} \in R) - \mathrm{vol}(R)| \leq \epsilon$ holds, where the volume $\mathrm{vol}(R)$ is defined to be $(\prod_i |S_i|)/m^n$. Suppose R is nontrivial in dimensions $a_1, a_2, \ldots, a_s$, where $s \leq k$. Now, by Lemma 12, $f(a_1), f(a_2), \ldots, f(a_s)$ are all distinct with a probability of at least $1 - \epsilon/2$; conditional on this, the definition of S' implies that $|\Pr(\vec{X} \in R) - \mathrm{vol}(R)| \leq \epsilon/2$. Also, if $f(a_1), f(a_2), \ldots, f(a_s)$ are not all distinct (which happens with a probability of at most $\epsilon/2$), $|\Pr(\vec{X} \in R) - \mathrm{vol}(R)|$ can be at most 1, with probability 1. Hence,

$$|\Pr(\vec{X} \in R) - \mathrm{vol}(R)| \leq (1 - \epsilon/2)\epsilon/2 + \epsilon/2 \leq \epsilon,$$

as required. □

Recall that an indexible ϵ-approximation S' for $\mathcal{R}^{n'}_m$, with $\log|S'| = O(\log m + \log n' + \log^2(1/\epsilon))$, is presented in [6]. Using this with Theorem 11 gives the following.

Corollary 13 There is an explicit indexible ϵ-approximation S'' for $\mathcal{R}^n_{m,k}$, with $\log|S''| = O(\log\log n + \log k + \log m + \log^2(1/\epsilon))$.

5.2 The main construction

Theorem 17, the main theorem here, constructs an indexible ϵ-approximation S for $\mathcal{R}^n_{m,k}$, with $\log|S| = O(\log\log n + (\log m)\log(1/\epsilon) + \log(1/\epsilon)\log(\lceil k/\log(1/\epsilon)\rceil))$. We start with Lemma 14.

Lemma 14 For any positive integer $t \leq n$, there is an explicitly constructible and indexible ϵ-approximation $T_{m,t,n,\epsilon}$ for $\mathcal{R}^n_{m,t}$, with $\log|T_{m,t,n,\epsilon}| = O(\log\log n + t\log m + \log(1/\epsilon))$.

Proof: Let $\epsilon' = \epsilon/m^t$. We shall show that taking $T_{m,t,n,\epsilon} = S_{m,t,n,\epsilon'}$ will suffice; the proof is then completed by invoking (12). We stress that this lemma is in itself quite simple, but will be useful later on when we show how to reduce our basic problem to the case of "small" t: $t = O(\log(1/\epsilon))$.

Let $\vec{X}$ be sampled uniformly at random from $S_{m,t,n,\epsilon'}$, and let $R = S_1 \times \cdots \times S_n$ be an arbitrary member of $\mathcal{R}^n_{m,t}$. We now show that $|\Pr(\vec{X} \in R) - (\prod_i |S_i|)/m^n| \leq \epsilon$, which will complete the proof. Assume w.l.o.g. that R is trivial in dimensions $t+1, t+2, \ldots, n$; hence, $S_{t+1} = S_{t+2} = \cdots = S_n = \{0, 1, \ldots, m-1\}$. For any $\vec{p} = (p_1, p_2, \ldots, p_t)$ where $p_1 \in S_1, p_2 \in S_2, \ldots, p_t \in S_t$, let $R'(\vec{p})$ denote $\{p_1\} \times \cdots \times \{p_t\} \times S_{t+1} \times \cdots \times S_n$. $R'(\vec{p})$ is trivially a member of $\mathrm{BOXES}_{t,n,m}$. Thus,

$$|\Pr(\vec{X} \in R'(\vec{p})) - 1/m^t| \leq \epsilon'. \tag{13}$$

Now,

$$\begin{aligned}
&|\Pr(\vec{X} \in R) - (\textstyle\prod_{i\in[n]} |S_i|)/m^n| \\
&= |\textstyle\sum_{p_1\in S_1,\ldots,p_t\in S_t} (\Pr(\vec{X} \in R'(\vec{p})) - 1/m^t)| \\
&\leq \textstyle\sum_{p_1\in S_1,\ldots,p_t\in S_t} |\Pr(\vec{X} \in R'(\vec{p})) - 1/m^t| \\
&\leq (\textstyle\prod_{i\in[t]} |S_i|)\epsilon' \text{ (by (13))} \\
&\leq m^t\epsilon' = \epsilon
\end{aligned}$$

as required. □

Our approach now is to reduce the problem of constructing an ϵ-approximation for $\mathcal{R}^n_{m,k}$ to that of constructing an ϵ'-approximation for $\mathcal{R}^n_{m,t}$, where t is "small" $(O(\log(1/\epsilon)))$ and with ϵ' chosen appropriately. We may then invoke Lemma 14. Next, a useful lemma from the full version of [10]:

Lemma 15 Let $Y_1, \ldots, Y_t$ be arbitrary binary r.v.s, and let $Z_1, \ldots, Z_t$ be independent binary r.v.s. Then, for any positive integer $s \leq t$, $|\Pr(\bigwedge_{i\in[t]}(Y_i = 0)) - \Pr(\bigwedge_{i\in[t]}(Z_i = 0))|$ is at most $2^{-s} + e^{1-s/(2e)} + \sum_{\ell=1}^{s} \sum_{A\subseteq[t]:|A|=\ell} |\Pr(\bigwedge_{i\in A}(Y_i = 1)) - \Pr(\bigwedge_{i\in A}(Z_i = 1))|$.

We also need a simple proposition from [10] (see also [8]):

Proposition 16 For any positive integers r, t, $r \leq t$, we have $\sum_{i=0}^{r} \binom{t}{i} \leq (te/r)^r$.

Define

$$\begin{aligned}
&k' = \min\{p \in \mathbf{N} : \ 2^{-p} + e \cdot e^{-p/(2e)} \leq \epsilon/2\}, \text{ and} \\
&\epsilon' = (\epsilon/2) \cdot (k'/(ke))^{k'}.
\end{aligned} \tag{14}$$

Note that $k' = \Theta(\log(1/\epsilon))$, and that $1/\epsilon' \leq \mathrm{poly}(1/\epsilon, (\lceil k/\log(1/\epsilon)\rceil)^{\log(1/\epsilon)})$. We now present our main result on approximating combinatorial rectangles:

Theorem 17 There is an explicit and indexible ϵ-approximation S for $\mathcal{R}^n_{m,k}$, with $\log|S|$ being

$$O(\log\log n + (\log m)\log(1/\epsilon) + \log(1/\epsilon)\log(\lceil k/\log(1/\epsilon)\rceil)).$$

Proof: Let k' and ϵ' be as in (14). We now show that taking $S = T_{m,k',n,\epsilon'}$ (as introduced in the statement of Lemma 14) will suffice; the upper bound on $\log|T_{m,k',n,\epsilon'}|$ from the statement of Lemma 14 will then complete the proof.

To show this, let $R = S_1 \times \cdots \times S_n$ be an arbitrary member of $\mathcal{R}^n_{m,k}$. We now prove that for $\vec{X}$ sampled uniformly at random from $T_{m,k',n,\epsilon'}$, $|\Pr(\vec{X} \in R) - (\prod_i |S_i|)/m^n| \leq \epsilon$, as required. We assume w.l.o.g. that R is trivial in dimensions $k+1, k+2, \ldots, n$. For each $i \in [k]$, let: (i) $Y_i \in \{0,1\}$ be a random variable that is 1 iff $X_i \notin S_i$, and (ii) $Z_i \in \{0,1\}$ be a random variable that is 1 iff a point drawn uniformly at random from $[0, m-1]^n$ does not lie in S_i. Thus, our goal is to show that

$$|\Pr(\bigwedge_{i\in[k]} (Y_i = 0)) - \Pr(\bigwedge_{i\in[k]} (Z_i = 0))| \leq \epsilon. \tag{15}$$

The proof technique now borrows largely from [10]. For each $i \in [k]$, let $T_i = \{0, 1, \ldots, m-1\} - S_i$. Note that the condition "$s \leq t$" in the statement of Lemma 15 is not crucial: if $s > t$, we can always reset $s := t$ for Lemma 15 to hold. Similarly, in our current context, if $k' > k$, we can set $k := k'$. By Lemma 15 and from the fact that $2^{-k'} + e \cdot e^{-k'/(2e)} \leq \epsilon/2$ by definition of k', we see that

$$\begin{aligned}
&|\Pr(\bigwedge_{i\in[k]} (Z_i = 0)) - \Pr(\bigwedge_{i\in[k]} (X_i = 0))| \\
&\leq \epsilon/2 + \textstyle\sum_{\ell=1}^{k'} \sum_{A\subseteq[k]:|A|=\ell} |\Pr(\bigwedge_{i\in A}(Y_i = 1)) \\
&\qquad\qquad - \Pr(\textstyle\bigwedge_{i\in A}(Z_i = 1))| \\
&\leq \epsilon/2 + \textstyle\sum_{\ell=1}^{k'} \sum_{A\subseteq[k]:|A|=\ell} |\Pr(\bigwedge_{i\in A}(X_i \in T_i) \\
&\qquad\qquad -(\textstyle\prod_{i\in A} |T_i|)/m^\ell|.
\end{aligned} \tag{16}$$

We now bound $|\Pr(\bigwedge_{i\in A}(X_i \in T_i) - (\prod_{i\in A} |T_i|)/m^\ell|$, for any generic set $A \subseteq [k]$ with $|A| = \ell \leq k'$. Since (i) $T_{m,k',n,\epsilon'}$ is an ϵ'-approximation for $\mathcal{R}^n_{m,k'}$ by Lemma 14, and (ii) $A \subseteq [k]$ with $|A| \leq k'$, we see that $|\Pr(\bigwedge_{i\in A}(X_i \in T_i)) - (\prod_{i\in A} |T_i|)/m^\ell| \leq \epsilon'$. Thus, by the bound (16), $|\Pr(\bigwedge_{i\in[k]}(Z_i = 0)) - \Pr(\bigwedge_{i\in[k]}(X_i = 0))|$ is at most $\epsilon/2 + \epsilon' \sum_{\ell=1}^{k'} \binom{k}{\ell}$, which is at most $\epsilon/2 + (ke/k')^{k'}\epsilon'$ by Proposition 16; this in turn is at most ϵ by definition of ϵ'. Thus, as (15) has been established, and the proof is complete. □

Note that the above proof will work with any ϵ'-approximation S for $\mathcal{R}^n_{m,k'}$, in place of the specific choice $S = T_{m,k',n,\epsilon'}$. Thus we get the following bootstrapping result:

Theorem 18 Let k' and ϵ' be as in (14). Then, any indexible ϵ'-approximation for $\mathcal{R}^n_{m,k'}$ is also an indexible ϵ-approximation for $\mathcal{R}^n_{m,k}$. In particular, suppose, for all (n, m, k, ϵ), there is an indexible ϵ-approximation S' for $\mathcal{R}^n_{m,k}$ with $\log|S'| = O(\log\log n + k + \log m + \log(1/\epsilon))$. Then , for all (n, m, k, ϵ), there is an indexible ϵ-approximation S'' for $\mathcal{R}^n_{m,k}$ with $\log|S''| = O(\log\log n + \log k + \log m + \log(1/\epsilon) + \log(1/\epsilon)\log(\lceil k/\log(1/\epsilon)\rceil))$.

6 Constructing certain Ramsey-type graphs

Recall the notion of an (s, t, n)-graph from the introduction; as mentioned there, we shall assume throughout w.l.o.g.

that $s \leq t$ (if we wish to construct an (s,t,n)-graph where $s > t$, we can always take the complement of a (t,s,n)-graph). The basic probabilistic approach to showing that an (s,t,n)-graph exists, is to construct a random graph G on n vertices, in which each edge is put in with a certain probability $p = p(s,t,n)$, independent of the other edges. The probability that any given subset A of the vertices with $|A| = s+1$ induces a clique is $p^{\binom{s+1}{2}}$; the probability that any given subset B of the vertices with $|B| = t+1$ induces an independent set is $(1-p)^{\binom{t+1}{2}}$. Thus, if s, t, n and p satisfy

$$\binom{n}{s+1} p^{\binom{s+1}{2}} + \binom{n}{t+1} (1-p)^{\binom{t+1}{2}} < 1, \qquad (17)$$

we get an (s,t,n)-graph with positive probability, i.e., we have shown that an (s,t,n)-graph exists. There are more involved probabilistic approaches than this to show the existence of (s,t,n)-graphs, e.g., using the "deletion method" [5] or, even stronger, the Lovász Local Lemma [33] or certain large-deviation inequalities [22]. If the basic probabilistic method (the usage of (17)) shows that an (s,t,n)-graph exists, these more refined methods usually help show that an (s,t^{α},n)-graph exists, where $\alpha < 1$ is some constant. Since there are still enormous gaps between the nonconstructive and known constructive bounds for Ramsey-type graphs, we just follow the basic approach of using (17) for our simple constructive method.

It can be checked that the method of conditional probabilities can be used to "constructivize" the simple approach above, leading to a deterministic sequential algorithm of time complexity poly$(\binom{n}{t})$. (Interestingly, this is also the best-known bound to check if a given graph is an (s,t,n)-graph.) The two drawbacks here are that (i) the running time is rather high, and (ii) this approach seems inherently sequential, i.e., not parallelizable. We tackle the first problem by a "graph product" result of [1], and the second via ϵ-approximations for axis-parallel rectangles.

Graph products. Given undirected graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, the following product graph $G_1 \times G_2 = (V_1 \times V_2, H)$ is considered in [1]: $\{(u_1,v_1),(u_2,v_2)\} \in H$ iff either (i) $u_1 \neq u_2$ and $(u_1,u_2) \in E_1$, or (ii) $u_1 = u_2$ and $(v_1,v_2) \in E_2$. The following fact is shown in [1]: if G_1 is an (s_1,t_1,n_1)-graph and G_2 is an (s_2,t_2,n_2)-graph, then $G_1 \times G_2$ is an $(s_1 s_2, t_1 t_2, n_1 n_2)$-graph. Applying this some r times on a given (s,t,ℓ)-graph G thus shows that G^r is an (s^r, t^r, ℓ^r)-graph. Thus, if for some "small" ℓ, we can efficiently construct an (s,t,ℓ)-graph G, we can then efficiently construct a much larger graph G^r that has ℓ^r vertices, with s and t getting replaced by s^r and t^r respectively.

Approximations of distributions. Let $X_1, X_2, \ldots, X_n$ be independent random variables taking on values in some finite set, say $\{0,1,\ldots,m-1\}$. We call a sample space (multi-set) D (whose elements are members of $\{0,1,\ldots,m-1\}^n$) a (k,ϵ)-approximation for $(X_1,X_2,\ldots,X_n)$ iff $\vec{Y} = (Y_1,Y_2,\ldots,Y_n)$ chosen uniformly at random from D satisfies:

$$\forall I \subseteq [n] \text{ with } |I| \leq k,\ \forall a_1,\ldots,a_{|I|} \in \{0,1,\ldots,m-1\},$$
$$|Pr(\wedge_{i\in I}(Y_i = a_i)) - \textstyle\prod_{i\in I} Pr(X_i = a_i)| \leq \epsilon.$$

Suppose S_1 is an explicit indexible $(\epsilon/2)$-approximation for BOXES$_{k,n,\ell}$, where $\ell = \lceil 4k/\epsilon \rceil$. Then, it is shown in [13] that there exists a (k,ϵ)-approximation S_2 for $(X_1,X_2,\ldots,X_n)$, such that: (a) $|S_2| = |S_1|$, and (b) given a uniformly random sample from S_1, a uniformly random sample from S_2 can be generated deterministically in time polynomial in n and m, and in NC. In fact, the sample space $S_{m,k,n,\epsilon}$ (see (12)) of [10] can be constructed in polylog$(n + |S_{m,k,n,\epsilon}|)$ time using poly$(n, |S_{m,k,n,\epsilon}|)$ processors in an EREW PRAM. Thus we have, in particular,

Lemma 19 Let $X_1, X_2, \ldots, X_a$ be i.i.d. binary random variables. Then, for any $k \leq a$, there is a (k,ϵ)-approximation S_0 for $(X_1,\ldots,X_a)$, such that: $|S_0| \leq$ poly$(\log a, 1/\epsilon, (\lceil k/\log(1/\epsilon)\rceil)^{\log(1/\epsilon)})$. Also, S_0 can be constructed by poly$(a, |S_0|)$ processors in polylog$(a + |S_0|)$ time on an EREW PRAM.

Theorem 20 There are absolute constants $c_0, c_1 > 0$ such that the following holds. Let $\ell \geq s \geq 3$ be any positive integers, r be any positive integer, and $t = \lceil c_0 \ell^{c_1/s} \log \ell \rceil \geq s$. Then, an (s^r, t^r, ℓ^r)-graph G can be constructed by poly$(\ell^r, \binom{\ell}{t}, \lceil t/\log \ell \rceil^{\log(\binom{\ell}{t})})$ processors on an EREW PRAM, in polylog$(\binom{\ell}{t} + r)$ time.

Proof: We just show how to construct an (s,t,ℓ)-graph H; we can then take the above-seen graph product r times to "boost" the construction to get the desired graph G.

If we construct a random graph on ℓ vertices $1, 2, \ldots, \ell$ with each edge probability being $p = \ell^{-c_2/s}$ for an appropriate constant $c_2 > 0$, it can be verified that s, t, p and ℓ satisfy

$$\binom{\ell}{s+1} p^{\binom{s+1}{2}} + \binom{\ell}{t+1} (1-p)^{\binom{t+1}{2}} < 1/2. \qquad (18)$$

(The constant $1/2$ in the r.h.s. is arbitrary; it can be replaced by $1 - \ell^{-\Theta(1)}$, to get slightly better values for s and t. We do not attempt this optimization here.) We can imagine the above random graph being constructed by generating $\binom{\ell}{2}$ i.i.d. binary random variables $\{X_{i,j} : 1 \leq i < j \leq \ell\}$, where $\Pr(X_{i,j} = 1) = p$ and where i and j are connected by an edge iff $X_{i,j} = 1$.

Construct, in parallel, a $(\binom{t+1}{2}, \epsilon)$-approximation S_0 for $\{X_{i,j} : 1 \leq i < j \leq \ell\}$, as guaranteed by Lemma 19; the value of ϵ will be determined below. Take a random sample (i.e., graph) from S_0. Since S_0 is a $(\binom{t+1}{2}, \epsilon)$-approximation, it is easily checked that the expected value of the sum of the number of induced cliques with $s+1$ vertices and the number of induced independent sets with $t+1$ vertices, is at most

$$\binom{\ell}{s+1} (p^{\binom{s+1}{2}} + \epsilon) + \binom{\ell}{t+1} ((1-p)^{\binom{t+1}{2}} + \epsilon).$$

For large enough ℓ, $t \leq \ell/2 - 1$; hence, since $s \leq t$, $\binom{\ell}{s+1} \leq \binom{\ell}{t+1}$. Thus, if we take $\epsilon = (4\binom{\ell}{t+1})^{-1}$, (18) shows that the above expected value is strictly smaller than 1. That is, at least one of the samples (graphs) in S_0 must be an (s,t,ℓ)-graph, and hence can be found by parallel exhaustive search in S_0. (Note that we can easily check if a given ℓ-vertex graph is an (s,t,ℓ)-graph, using poly$(\binom{\ell}{t+1})$ processors and polylog$(\binom{\ell}{t+1})$ time, on an EREW PRAM.)

Using the cardinality of and construction time for S_0 specified by Lemma 19, we complete the proof of the theorem. □

The two extremes for Ramsey-type n-vertex graphs are (i) where $s \sim t$, and (ii) where $s \ll t$, i.e., $s = c$ (some constant), and $t = n^{f(c)}$. While Theorem 20 can be used to construct a range of Ramsey-type graphs, the following corollary just lists these two extremes:

Corollary 21 For any n, the following Ramsey-type graphs can be constructed in parallel on an EREW PRAM. (i) $(2^{O(\sqrt{\log n})}, 2^{O(\sqrt{\log n})}, n)$-graphs using $n^{O((\log\log n)^2)}$ processors and polylog(n) time, and (ii) for any desired constants $\epsilon, \delta > 0$, $\epsilon < 1$, (c, n^{ϵ}, n)-graphs using $\exp(n^{\delta})$ processors and $O(n^{\delta})$ time, where $c = c(\epsilon, \delta)$ is a constant.

Proof: We apply Theorem 20 using the following values for the parameters. For (i), we take both s and t to be $\Theta(\sqrt{\log n}\log\log n)$, $\ell = 2^{\sqrt{\log n}\log\log n}$, and $r = \Theta(\sqrt{\log n}/\log\log n)$. For (ii), we take $s = c'(\epsilon, \delta)$, i.e., a suitably large constant, $t = n^{\epsilon'}$, $\ell = n^{\delta'}$ and $r = d$, where ϵ' and δ' are sufficiently small positive constants, and d is a sufficiently large constant. □

Acknowledgments. Theorem 7 was proved with David Haussler and Nick Littlestone; we thank them for their contribution. We also thank the STOC 1997 program committee member(s) and referee(s) for their helpful comments. Finally, we thank Carlos Domingo for pointing out an error in the hardness proof of Section 3.3 in an earlier version of the paper.

References

[1] H. L. Abbott. A note on Ramsey's theorem. Canad. Math. Bull., 15:9–10, 1972.

[2] N. Alon. Explicit Ramsey graphs and orthonormal labelings. The Electronic Journal of Combinatorics, 1, 1994, R12, 8pp.

[3] N. Alon, L. Babai, and A. Itai. A fast and simple randomized parallel algorithm for the maximal independent set problem. Journal of Algorithms, 7:567–583, 1986.

[4] N. Alon, O. Goldreich, J. Håstad, and R. Peralta. Simple constructions of almost k-wise independent random variables. Random Structures and Algorithms, 3(3):289–303, 1992.

[5] N. Alon, J. H. Spencer, and P. Erdős. The Probabilistic Method. John Wiley & Sons, 1992.

[6] R. Armoni, M. Saks, A. Wigderson, and S. Zhou. Discrepancy sets and pseudorandom generators for combinatorial rectangles. In Proc. IEEE Symposium on Foundations of Computer Science, pages 412–421, 1996.

[7] P. Auer. On learning from multi-instance examples: empirical evaluation of a theoretical approach. Submitted, 1997.

[8] A. Blumer, A. Ehrenfeucht, D. Haussler, and M.K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. JACM, 36(4):929–965, 1989.

[9] J. L. Carter and M. N. Wegman. Universal classes of hash functions. Journal of Computer and Systems Sciences, 18:143–154, 1979.

[10] S. Chari, P. Rohatgi, and A. Srinivasan. Improved Algorithms via Approximations of Probability Distributions. In Proc. ACM Symposium on the Theory of Computing, 584–592, 1994. Full version available as Technical Report TR 10/96, Dept. of Information Systems and Computer Science, National University of Singapore, 1996.

[11] T.G. Dietterich, R.H. Lathrop, and T. Lozano-Perez. Solving the multiple-instance problem with axis-parallel rectangles. Artificial Intelligence, 89(1-2):31–71,1997.

[12] A. Ehrenfeucht, D. Haussler, M. Kearns, and L.G. Valiant. A general lower bound on the number of examples needed for learning. Information and Computation, 82(3):247–251, 1989.

[13] G. Even, O. Goldreich, M. Luby, N. Nisan, and B. Veličković. Approximations of general independent distributions. In Proc. ACM Symposium on the Theory of Computing, pages 10–16, 1992.

[14] P. Erdős. Some remarks on the theory of graphs. Bulletin of the American Mathematics Society, 53:292–294, 1947.

[15] W. Feller. An Introduction to Probability and its Applications, volume 1. John Wiley and Sons, third edition, 1968.

[16] P. Frankl. A constructive lower bound for Ramsey numbers. Ars Combinatoria, 3:297–302, 1977.

[17] P. Frankl and R. M. Wilson. Intersection theorems with geometric consequences. Combinatorica, 1:357–368, 1981.

[18] D. Haussler, N. Littlestone, and M. K. Warmuth. Predicting $\{0,1\}$-functions on randomly drawn points. Information and Computation, 115(2):129–161, 1994.

[19] R. Impagliazzo, N. Nisan, and A. Wigderson. Pseudorandomness for network algorithms. In Proc. ACM Symposium on the Theory of Computing, pages 356–364, 1994.

[20] M.J. Kearns. Efficient noise-tolerant learning from statistical queries. In Proc. ACM Symposium on the Theory of Computing, pages 392–401, 1993.

[21] M. Kearns, M. Li, L. Pitt, and L.G. Valiant. On the learnability of Boolean formulae. Proceedings of the 19th Annual Symposium on the Theory of Computation, pages 285–295, 1987.

[22] M. Krivelevich. Bounding Ramsey numbers through large deviation inequalities. Random Structures and Algorithms, 7:145–155, 1995.

[23] N. Littlestone. Learning quickly when irrelevant attributes abound: a new linear-threshold algorithm. Machine Learning, 2:285–318, 1988.

[24] N. Linial, M. Luby, M. Saks, and D. Zuckerman. Efficient Construction of a Small Hitting Set for Combinatorial Rectangles in High Dimension. In Proc. ACM Symposium on the Theory of Computing, pages 258–267, 1993.

[25] P.M. Long and L. Tan. PAC learning axis-aligned rectangles with respect to product distributions from multiple-instance examples. The 1996 Conference on Computational Learning Theory, pages 228–234, 1996.

[26] A. Lubotzky, R. Phillips and P. Sarnak. Ramanujan graphs. Combinatorica, 8:261–277, 1988.

[27] M. Luby. A simple parallel algorithm for the maximal independent set problem. SIAM J. Comput., 15(4):1036–1053, 1986.

[28] J. Naor and M. Naor. Small-bias probability spaces: efficient constructions and applications. SIAM J. Comput., 22(4):838–856, 1993.

[29] N. Nisan. Extracting Randomness: How and Why. In Proc. IEEE Conference on Computational Complexity (formerly "Structure in Complexity Theory"), pages 44–58, 1996.

[30] L. Pitt and L.G. Valiant. Computational limitations on learning from examples. Journal of the ACM, 35(4):965–984, 1988

[31] L. Pitt and M.K. Warmuth. Prediction preserving reducibility. Journal of Computer and System Sciences, 41(3), 1990.

[32] D. Pollard. Convergence of Stochastic Processes. Springer Verlag, 1984.

[33] J. H. Spencer. Asymptotic lower bounds for Ramsey functions. Discrete Math., 20:69–76, 1977.

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

[35] V.N. Vapnik and A.Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. Theory of Probability and its Applications, 16(2):264–280, 1971.

[36] S.M. Weiss and C.A. Kulikowski. Computer systems that learn. Morgan Kauffman, 1991.

A Proof of Lemma 4

If X is a set, and G is a set of $\{0,1\}$ valued functions defined on X, define

$$\mathrm{VCdim}(G) = \max\{d : \exists x_1, ..., x_d \in X, \\ \{(g(x_1), ..., g(x_d)) : g \in G\} = \{0,1\}^d\}.$$

We will make use of the following lemma.

Lemma 22 ([35]) There is a constant c such that, for all X, for all permissible[1] sets G of $\{0,1\}$ valued functions defined on X, for all probability distributions D on X and $m \in \mathbf{N}$, and for all $0 < \epsilon, \delta < 1/2$, if $\mathrm{VCdim}(G) = 1$, and $m \geq \frac{c}{\epsilon^2} \log \frac{1}{\delta}$, then

$$D^m \left\{ \vec{x} : \exists g \in G \left| \left(\frac{1}{m} \sum_{i=1}^{m} g(x_i) \right) - \mathbf{E}_{u \in D}(g(u)) \right| > \epsilon \right\} \leq \delta.$$

Next, we record the VC-dimension of the set of random variables whose probabilities are the $\alpha_k(b)$'s as b varies. The proof is the same as the known proof for $\{(-\infty, b] : b \in \mathbf{R}\}$.

Lemma 23 Choose $n, k \in \mathbf{N}$, $k \leq n$. For each b, define $g_b : \mathbf{R}^n \to \{0,1\}$, by $g_b(\vec{u}) = 1 \Leftrightarrow u_k > b$. Then $\mathrm{VCdim}(\{g_b : b \in \mathbf{R}\}) \leq 1$.

Proof: Choose $\vec{u}, \vec{v} \in \mathbf{R}^n$. Assume without loss of generality that $u_k \leq v_k$. Then there is not a b such that $g_b(\vec{u}) = 1$ and $g_b(\vec{v}) = 0$. □

Now we analyze the VC-dimension relating to the $\varphi_k(b)$'s.

Lemma 24 Choose $n, k, r \in \mathbf{N}$, $k \leq n$, $\vec{a} \in \mathbf{R}^n$. For each b, define $g_b : (\mathbf{R}^n)^r \to \{0,1\}$, by

$$g_b(\vec{u}_1, ..., \vec{u}_r) = \begin{cases} 1 & \text{if } u_{1,k} > b \text{ and } \psi_{r_{\vec{a}}}(\vec{u}_1, ..., \vec{u}_n) = 0 \\ 0 & \text{otherwise.} \end{cases}$$

Then $\mathrm{VCdim}(\{g_b : b \in \mathbf{R}\}) \leq 1$.

Proof: Assume for contradiction that $\mathrm{VCdim}(\{g_b : b \in \mathbf{R}\}) > 1$. Then there exist $\vec{u}_1, ..., \vec{u}_r$ and $\vec{v}_1, ..., \vec{v}_r$ such that

$$\{(g_b(\vec{u}_1, ..., \vec{u}_r), g_b(\vec{v}_1, ..., \vec{v}_r)) : b \in \mathbf{R}\} = \{0,1\}^2.$$

Since there exists b such that $(g_b(\vec{u}_1, ..., \vec{u}_r), g_b(\vec{v}_1, ..., \vec{v}_r)) = (1,1)$, we have $\psi_{r_{\vec{a}}}(\vec{u}_1, ..., \vec{u}_r) = \psi_{r_{\vec{a}}}(\vec{v}_1, ..., \vec{v}_r) = 0$. Thus, for any $b \in \mathbf{R}$

$$\begin{aligned} g_b(\vec{u}_1, ..., \vec{u}_r) = 1 \Leftrightarrow u_{1,k} > b \\ g_b(\vec{v}_1, ..., \vec{v}_r) = 1 \Leftrightarrow v_{1,k} > b \end{aligned} \qquad (19)$$

The fact that there is a b such that $(g_b(\vec{u}_1, ..., \vec{u}_r), g_b(\vec{v}_1, ..., \vec{v}_r)) = (1,0)$ implies in conjunction with (19) that $u_{1,k} > v_{1,k}$, but the fact that there is a b such that $(g_b(\vec{u}_1, ..., \vec{u}_r), g_b(\vec{v}_1, ..., \vec{v}_r)) = (0,1)$ implies in conjunction with (19) that $u_{1,k} < v_{1,k}$, a contradiction. □

Proof of Lemma 4: Applying the usual Hoeffding bound (see [32, Appendix B]) proves (9). Combining[2] Lemma 22, Lemma 23 and Lemma 24, proves (10). □

[1] A technical measurability constraint (see Pollard's [32] Appendix C).

[2] The permissibility of the relevant sets of r.v.'s is easily verified.