Super-linear time-space tradeoff lower bounds for randomized computation

Paul Beame*
Computer Science and Engineering
University of Washington
Seattle, WA 98195-2350
beame@cs.washington.edu

Michael Saks†
Dept. of Mathematics
Rutgers University
New Brunswick, NJ
saks@math.rutgers.edu

Xiaodong Sun†
Dept. of Mathematics
Rutgers University
New Brunswick, NJ
sunxd@math.rutgers.edu

Erik Vee
Computer Science and Engineering
University of Washington
Seattle, WA 98195-2350
env@cs.washington.edu

Abstract

We prove the first time-space lower bound tradeoffs for randomized computation of decision problems. The bounds hold even in the case that the computation is allowed to have arbitrary probability of error on a small fraction of inputs. Our techniques are an extension of those used by Ajtai [4, 5] in his time-space tradeoffs for deterministic RAM algorithms computing element distinctness and for deterministic Boolean branching programs computing an explicit function based on quadratic forms over $GF(2)$.

Our results also give a quantitative improvement over those given by Ajtai. Ajtai shows, for certain specific functions, that any branching program using space $S = o(n)$ *requires time* T *that is superlinear. The functional form of the superlinear bound is not given in his paper, but optimizing the parameters in his arguments gives* $T = \Omega(n \log\log n / \log\log\log n)$ *for* $S = O(n^{1-\epsilon})$. *For the same functions considered by Ajtai, we prove a time-space tradeoff of the form* $T = \Omega(n\sqrt{\log(n/S)/\log\log(n/S)})$. *In particular, for space* $O(n^{1-\epsilon})$, *this improves the lower bound on time to* $\Omega(n\sqrt{\log n/\log\log n})$.

1 Introduction

The study of time-space tradeoffs for computational problems is fundamental to complexity theory. These tradeoffs were considered early in the history of complexity, and have continued to be an important area of research [9]. An important motivation for these investigations was the observation that for some natural problems such as sorting, algorithms were known that were extremely space efficient, and other algorithms were known that were very time efficient, but no known algorithm could simultaneously achieve the optimal time and space efficiency. Another motivation came from the general study of lower bounds where time-space tradeoff lower bounds can be viewed as milestones towards proving nontrivial (superlogarithmic) space lower bounds for problems in P or NP.

As with most lower bound problems in complexity theory, research divides into "uniform" and "nonuniform" models. In the uniform setting, a series of recent papers have established limitations on Turing machines computing SAT. The first work along these lines was by Fortnow [14], which was followed by [16] and [15]. The latter gives the best current result: any algorithm for SAT that runs in space $n^{o(1)}$ requires time at least $\Omega(n^{\phi-\epsilon})$ where $\phi = (\sqrt{5}-1)/2$ and ϵ is any positive constant. Although some of these lower bounds apply even to co-nondeterministic computation, none of them give any results for randomized algorithms.

In the nonuniform setting, the standard model is the *branching program*. In this model, a program for computing a function $f(x_1, \ldots, x_n)$ (where the variables take values in some finite domain D) is represented as a DAG with a unique start node. Each non-sink node is labeled by a variable and the arcs out of a node correspond to the possible values of the variable. Each sink node is labeled by an output value. Executing the program on a given input corresponds to following a path from the start node using the values of the input variables to determine the arcs to follow. The output of the program

* Research supported by NSF grant CCR-9800124.
†Research supported by NSF grant CCR-9700239 and by DIMACS

0-7695-0850-2/00 $10.00 © 2000 IEEE

is value labeling the sink node reached. The maximum length of a path corresponds to time and the logarithm of the number of nodes corresponds to space. This model is often called the D-way branching program model; in the case that the domain D is $\{0, 1\}$ is referred to as the *Boolean branching program* model.

In this model (or more precisely an extension which permits outputs along arcs during the course of computation), there was considerable success in proving time-space tradeoff lower bounds for *multi-output functions* such as sorting, pattern matching, matrix-vector product and hashing [10, 7, 1, 2, 17]. However, these techniques fundamentally break down when considering decision problems. In the *comparison branching program model* (where the inputs are numbers and the tests at nodes are pairwise comparisons of inputs) there were strong results obtained for the decision problem element distinctness [11, 18]. In the Boolean model, there is an extensive literature on various restricted *read-k* models ([13]) which have strict limitations on the number of times that any one variable may appear on any path in the branching program.

Recently, the first results have been obtained for decision problems on unrestricted branching programs using time more than n. In the D-way model, [8] exhibited a problem in P, where the domain D grows with the number of variables n, for which any subexponential size nondeterministic branching program has length $\Omega(n \log \log n)$. In the Boolean case, they obtained the first (barely) nontrivial bound by exhibiting a problem in P and a constant $\epsilon > 0$ for which any subexponential size branching program requires length at least $(1+\epsilon)n$. Extending techniques of [13] for bilinear forms, the lower bounds in [8] were shown for functions based on quadratic forms over finite fields.

In a remarkable breakthrough, Ajtai [5] exhibited a P-time computable Boolean function (also based on quadratic forms) for which any subexponential size deterministic branching program requires superlinear length. Much of the technical argument for this result was contained in a previous paper of Ajtai [4, 3] which developed a key tool for analyzing the branching programs. The earlier paper gave similar lower bounds for two non-Boolean problems whose input is a list of n binary strings, each of length $b = O(\log n)$ bits long:(1) determine whether the list contains a pair of strings within hamming distance δb for some fixed $\delta > 0$, and (2) determine whether the strings are all distinct.

The basic approach of all of these papers was to show that any branching program of "small" length and size must accept a subset of inputs that form a "large" *embedded rectangle*, and then to demonstrate that some particular functions don't have large embedded rectangles. for syntactic read-k branching programs in [13]. The first lower bounds on embedded rectangle size for general branching programs of small size and length was done by [8]. These bounds yielded the results from that paper mentioned above, and are also strong enough to give the hamming distance result of [4], but were not strong enough to yield the element distinctness and Boolean function lower bounds. Ajtai obtained these bounds by proving an amazing combinatorial lemma that gave much stronger lower bound on embedded rectangle size. This directly gave his tradeoff results for element distinctness and was the basis for the subsequent Boolean branching program lower bound.

1.1 Our results

In this paper, we extend Ajtai's approach for deterministic branching programs in order to obtain the first time-space tradeoff results for (two-sided error) randomized branching programs, and also for deterministic branching programs that are allowed to err on a small fraction of inputs. Previously, there were no known time-space tradeoffs even in the uniform setting for these modes of computation. Our results apply to randomized RAM algorithms as well.

We also obtain substantial quantitative improvement over the previous results. More specifically, we show that, for the functions considered by Ajtai, any branching program of subexponential size must have length at least $\Omega(n\sqrt{\frac{\log n}{\log\log n}})$. Ajtai does not explicitly give the functional form of his length bounds, but analyzing his argument gives at most an $\Omega(n\frac{\log\log n}{\log\log\log n})$ bound.

Finally, while our argument is heavily based on Ajtai's, our version is considerably simpler.

One of the key aspects of both our extension and our simplification is to apply the basic approach developed in [8] of breaking up branching programs into collections of decision trees called decision forests and then analyzing the resulting decision forests. This has the effect of applying the space restriction only once, early in the argument, rather than delaying the application of the space restriction until the end of the argument which complicates the analysis without fundamentally changing its ideas.

Our extension of Ajtai's lemma shows that for a small deterministic branching program not only is there a large embedded rectangle of accepted inputs, but there is a set of large embedded rectangles of accepted inputs that cover almost all such inputs without covering any one input too many times. From this we show that if the given branching program agrees with a given target function f on all but a small fraction of inputs then there is a large embedded rectangle almost all of whose inputs are ones of f. We obtain our lower bounds for random algorithms by strengthening Ajtai's arguments about element distinctness and the quadratic forms to show that, not only do the functions not accept any relatively large embedded rectangle, they reject a large fraction of inputs in any such rectangle.

2 Definitions

Throughout this paper D denotes a finite set and n a positive integer. We write $[n]$ for the set $\{1, \ldots, n\}$. For finite set N, an *N-ary input over domain D* is a point x in D^N, the set of maps from N to D. An element of N is called a *variable index* or, simply, an *index*. We normally take N to be $[n]$ for some integer n, and write D^n for $D^{[n]}$.

If $A \subseteq N$, a point $\sigma \in D^A$ is a *partial input on A*. For a partial input σ, $\texttt{fixed}(\sigma)$ denotes the index set A on which it is defined and $\texttt{unfixed}(\sigma)$ denotes the set $N - A$. If σ and π are partial inputs with $\texttt{fixed}(\sigma) \cap \texttt{fixed}(\pi) = \emptyset$ then $\sigma\pi$ denotes the partial input on $\texttt{fixed}(\sigma) \cup \texttt{fixed}(\pi)$ that agrees with σ on $\texttt{fixed}(\sigma)$ and with π on $\texttt{fixed}(\pi)$.

For $x \in D^N$ and $A \subseteq N$, the *projection x_A of x onto A* is the partial input on A that agrees with x. For $S \subseteq D^N$, $S_A = \{x_A : x \in S\}$. For a partial input σ, $D^N(\sigma)$, the set of *extensions* of σ in D^N, is $\{x \in D^N : x_{\texttt{fixed}(\sigma)} = \sigma\}$.

A product $U \times V$ of two finite sets is called a *(combinatorial) rectangle*. If $A \subseteq N$ is an index subset, and $Y \subseteq D^A$ and $Z \subseteq D^{N-A}$, then the product set $Y \times Z$ is naturally identified with the subset $R = \{\sigma\rho : \sigma \in Y, \rho \in Z\}$ of D^N, and a set of this form is called a *rectangle in D^N*. This notion of rectangle has been used, for example, in the study of communication complexity in the "best-partition" model and in the study of read-once branching programs.

We need a more general notion of rectangle. An *embedded rectangle* R in D^N is a triple (B, A_1, A_2) where A_1 and A_2 are disjoint subsets of N and $B \subseteq D^N$ satisfies: (i) The projection $B_{N-A_1-A_2}$ consists of a single partial input σ, (ii) If $\tau_1 \in B_{A_1}, \tau_2 \in B_{A_2}$ then the point $\tau_1\tau_2\sigma \in R$. B is called the *body* of R and A_1 and A_2 are the feet of R. The sets R_{A_1} and R_{A_2} are the *legs* of the rectangle and σ is the *spine*. Abusing terminology, we typically use the same letter for an embedded rectangle and its body, writing $R = (R, A_1, A_2)$. This could cause trouble if we needed to refer to two rectangles with the same body but different feet, but this will not come up in this paper.

We can specify an embedded rectangle by its feet, legs and spine. Let A_1 and A_2 be disjoint subsets of N, $Y_1 \subseteq D^{A_1}$ and $Y_2 \subseteq D^{A_2}$, and σ be a partial input on $N - A_1 - A_2$. Then the set $\{\tau_1\tau_2\sigma : \tau_1 \in Y_1, \tau_2 \in Y_2\}$ is the body of the unique embedded rectangle with feet (A_1, A_2), legs (Y_1, Y_2) and spine σ.

For an embedded rectangle $R = (R, A_1, A_2)$, and $j \in \{1, 2\}$ we define $m_j(R)$ to be $|A_j|$. If $m_1(R) = m_2(R)$ we say that R is a *balanced* rectangle, and define $m(R)$ to be the common value. We also define $\alpha_j(R)$ is defined to be $|R_{A_j}|/|D^{A_j}|$, and the *leg-density* of R, $\alpha(R)$ to be $\min\{\alpha_1, (R), \alpha_2(R)\}$.

For later reference, we note an easy technical fact:

Proposition 1. *Let (R, A_1, A_2) be an embedded rectangle, let $B_1 \subseteq A_1$, and let $\delta > 0$. Then there is a collection $\mathcal{R}$ of disjoint embedded rectangles contained in R that together cover at least a $(1 - \delta)$ fraction of points of R, and such that each $Q \in \mathcal{R}$ has feet (B_1, A_2), and satisfies $\alpha_2(Q) = \alpha_2(R)$ and $\alpha_1(Q) \geq \delta\alpha_1(R)$.*

We use the standard definitions of deterministic branching programs as described in the introduction. We say that a branching program is *inquisitive* if on every input x, the path followed by x reads all of the variables of x. We view randomized branching programs as distributions over deterministic branching programs; this model is at least as powerful as the model in which the programs contain explicit random choice nodes, and thus our results apply to both models since we prove lower bounds.

A *decision tree* is a branching program B whose underlying graph is a tree. Every function on n variables is computable by a deterministic decision tree of length n. Following common practice, the length of a decision tree is referred to as its *height*.

A *decision forest* is a set of decision trees. More precisely for domain D and integers n and r and $\epsilon > 0$, an n-variate (r, ϵ)-decision forest F over D is a collection of at most r decision trees such that each tree is an n-variate tree over domain D and has height at most ϵn. F is viewed as a function on D^n by the rule $F(x) = \wedge_{T \in F} T(x)$. A decision forest F is *inquisitive* if on every input x, for each $i \in [n]$, at least one of the trees $T \in F$ reads x_i.

3 Main Decomposition Theorems

3.1 Overview

The main approach taken in [8, 4, 5] for proving time-space tradeoff lower bounds is to show that if f can be computed by a branching program running in time T and space S, where T and S are suitably small, then f must evaluate to 1 on some embedded rectangle R whose feet and leg-density are both large. Roughly speaking, large feet means that $m(R) = \beta n$ where β is a function of

T/n, and large leg-density means $\alpha(R) > |D|^{-w(m(R))}$ where $w(m)$ is small enough compared to m.

The first step in showing the existence of these embedded rectangles is to view a branching program of length kn as divided into $r \gg k$ layers, each layer consisting of roughly $\frac{k}{r}n$ consecutive levels, and to focus on the impact of the space limitation only at the boundary between these layers. This echoes the approach used for multi-output problems, starting with [12, 10], but in the multi-output case the argument relies heavily on the fact that for each input, there must be a layer that produces at least a $1/r$ fraction of the output for that input, which allows one to derive a contradiction by restricting attention to a single layer. In the case of decision problems, we must maintain the connection between the different layers. We do this by using the decomposition from [8] of any branching program into a disjunction of decision forests (Lemma 2).

There are two main differences between our results and previous results for decision problems. First of all, we obtain substantially larger values for the feet and leg-density of the obtained rectangles. Secondly, we show that not only is there one large embedded rectangle in $f^{-1}(1)$ but there is a collection of such embedded rectangles that together cover *almost all* the inputs on which f outputs 1, and such that no input is covered more that $2k$ times. This allows us to prove lower bounds for randomized and distributional as well as deterministic branching program complexity.

We summarize the relationships between the different results in Figure 1. In each case, we assume that we begin with a D-way branching program that accepts a δ fraction of all inputs in D^n. Each result shows that a branching program for some function f running in time $T = kn$ and space S admits a large embedded rectangle, whose feet are each size at least βn where $\beta = \beta(k)$ and whose leg-density is at least $\alpha = 2^{-\lambda(\beta)n-Sr}$ where λ is a nonnegative function of β and r is a function of k. In the first three lines of the table, the guaranteed rectangle consists entirely of points in $f^{-1}(1)$. In the last case, where the branching program is randomized with 2-sided error ϵ, the guaranteed rectangle may contain a small fraction of 0's. The last column gives the size of D and range of k for which the rectangle bounds can be used to get linear space bounds for explicit functions. The restriction on D in the first line comes from the fact that to get a nontrivial tradeoffs we need that $2^{\lambda(\beta)n}$ is "sufficiently small" compared to $D^{\beta n}$. The upper bound on k in each row is because the lower bound on space is roughly $\frac{\beta}{r}n$.

In section 3.2 we describe the decomposition into decision forests and in section 3.3 we show how to find the appropriate partitions into embedded rectangles within each decision forest. We derive the key theorems concerning branching programs in section 3.4.

3.2 Decomposition into Decision Forests

The following lemma is a minor variant of one proved in [8].

Lemma 2. *Let $k \in \mathbf{R}$ and $n, s \in \mathbf{N}$ and D be a finite set. Let B be an (inquisitive) n-variate branching program over domain D having length at most kn and size at most s. Then for any integer $r \in [k, n]$, the function f computed by B can be expressed as: $f = \bigvee_{i=1}^{u} F_i$, where $u \leq s^r$, each F_i is an (inquisitive) $(r, \frac{k+1}{r})$-decision forest, and the sets $F_i^{-1}(1)$ are pairwise disjoint sets of inputs.*

3.3 Finding large embedded rectangles in decision forests

Throughout this section, D is a fixed finite domain, $n \geq r \geq k \geq 1$ are integers and F is a fixed inquisitive D-way $(r, k/r)$-decision forest over index set $[n]$. For input $x \in D^n$ and decision tree T we define $\mathtt{read}(x, T)$ to be the set of indices read by T on input x. For $F_1 \subseteq F$, $\mathtt{read}(x, F_1) = \bigcup_{T \in F_1} \mathtt{read}(x, T)$.

Our goal is to cover $F^{-1}(1)$ by large rectangles, where "large" means that both the feet and leg-density are suitably large, as described in Section 3.1. Our first step is to show that any pair (F_1, F_2) of disjoint subforests of F is naturally associated with a partition $\mathcal{R}(F_1, F_2)$ of $F^{-1}(1)$ into embedded rectangles.

Let F_1 and F_2 be disjoint subsets of F, $x \in D^n$ and $S \subseteq D^n$. We define:

- $\mathtt{core}(x, F_1) = \mathtt{read}(x, F_1) - \mathtt{read}(x, F - F_1)$, the F_1*-core of* x, is the set of indices which on input x are read by at least one tree in F_1 and by no tree outside of F_1. By our assumption that F is inquisitive, this is the same as $[n] - \mathtt{read}(x, F - F_1)$.
- $\mathtt{stem}(x, F_1)$, the F_1-stem of x, is the partial input obtained by projecting x to $[n] - \mathtt{core}(x, F_1)$.
- $\mathtt{stem}(x, F_1, F_2)$ is the partial input on $[n] - \mathtt{core}(x, F_1) - \mathtt{core}(x, F_2)$ obtained from projecting x.

We now come to the key definition. We say that inputs $x, y \in F^{-1}(1)$ are (F_1, F_2)*-equivalent* if and only if $\mathtt{core}(x, F_1) = \mathtt{core}(y, F_1)$, $\mathtt{core}(x, F_2) = \mathtt{core}(y, F_2)$, and $\mathtt{stem}(x, F_1, F_2) = \mathtt{stem}(y, F_1, F_2)$. Let $\mathcal{R}(F_1, F_2)$ be the set of (F_1, F_2)-equivalence classes. For $R \in \mathcal{R}(F_1, F_2)$, we write $\mathtt{core}(R, F_1)$ for the common value of $\mathtt{core}(x, F_1)$ shared by all $x \in R$ and define $\mathtt{core}(R, F_2)$ and $\mathtt{stem}(R, F_1, F_2)$ analogously.

Paper	β	r	$\lambda(\beta)$	Program type	$\frac{\lvert R\cap f^{-1}(0)\rvert}{\lvert R\rvert}$	Applicability
[8]	2^{-k}	$\theta(k^2 2^k)$	$4(k+1)\beta$	non-determ.	0	$\log\log\lvert D\rvert = \Omega(k)$,$k = O(\frac{\log n}{\log\log n})$
[4]	$2^{-k^{\theta(k)}}$	$2^{k^{\theta(k)}}$	$\beta^{1+1/(50k)}$	determ.	0	all D, $k = O(\frac{\log\log n}{\log\log\log n})$
Here	$k^{-\theta(k^2)}$	$k^{\theta(k^2)}$	$\beta^{1+1/(8k^2)}$	determ.	0	all D, $k = O(\sqrt{\frac{\log n}{\log\log n}})$
Here	$k^{-\theta(k^2)}$	$k^{\theta(k^2)}$	$\beta^{1+1/(8k^2)}$	random 2-sided err. ϵ	$O(k\epsilon)$	all D, $k = O(\sqrt{\frac{\log n}{\log\log n}})$

Figure 1. Results concerning existence of an embedded rectangle with $m(R) \geq \beta n$ and $\alpha(R) \geq 2^{-\lambda(\beta)n-Sr}$ for a given branching program with time $T = kn$ and space S that accepts a δ fraction of D^n.

Lemma 3. *Let F be an inquisitive decision forest and let $F_1, F_2 \subset F$ be disjoint subforests of F. Let $R \in \mathcal{R}(F_1, F_2)$. Then R is an embedded rectangle with feet $(\texttt{core}(R, F_1), \texttt{core}(R, F_2))$ and spine $\texttt{stem}(R, F_1, F_2)$.*

Proof. Let $A_1 = \texttt{core}(R, F_1)$ and $A_2 = \texttt{core}(R, F_2)$ and $\sigma = \texttt{stem}(R, F_1, F_2)$. By definition, A_1 and A_2 are disjoint. Let Q be the embedded rectangle defined by A_1, A_2, R_{A_1}, R_{A_2}, and σ. Clearly, $R \subseteq Q$, and it suffices to show that $Q \subseteq R$. Let A_0 denote $[n] - A_1 - A_2$.

Let $z \in Q$. By definition of Q, there is a $y^1 \in R$ such that $z_{A_1} = y^1_{A_1}$ and a $y^2 \in R$ such that $z_{A_2} = y^2_{A_2}$. Furthermore $z_{A_0} = y^1_{A_0} = y^2_{A_0} = \rho$. On input y^1, each tree $T \in F - F_2$ reads only variables in $[n] - A_2$, so each $T \in F - F_2$ behaves the same on z as it does on y^1. Therefore $\texttt{read}(z, F - F_2) = \texttt{read}(y^1, F - F_2)$, and thus $\texttt{core}(z, F_2) = \texttt{core}(y^1, F_2) = A_2$ since F is inquisitive, and each tree in $F - F_2$ accepts z since $R \subseteq F^{-1}(1)$. By a symmetric argument, $\texttt{core}(z, F_1) = \texttt{core}(y^2, F_1) = A_1$, and each tree in $F - F_1$ accepts z. Thus $z \in F^{-1}(1)$, $\texttt{core}(z, F_1) = A_1$, $\texttt{core}(z, F_2) = A_2$, and $\texttt{stem}(z, F_1, F_2) = \rho$ and thus $z \in R$ as required. □

Thus, each pair of disjoint forests F_1, F_2 induces a partition of $F^{-1}(1)$ into disjoint embedded rectangles. However, we have no guarantee that for an arbitrary (F_1, F_2) the rectangles in $\mathcal{R}(F_1, F_2)$ are large. In fact, we will not be able to show that any one fixed pair (F_1, F_2) can guarantee large rectangles. Instead, we will show that almost every accepted input is contained in a large rectangle from $\mathcal{R}(F_1, F_2)$ for one of a small number of pairs (F_1, F_2).

To choose the desired pairs of forests, we will analyze properties of the rectangle family corresponding to a pair of forests chosen at random. In [8], (F_1, F_2) was chosen to be a random partition of F into two parts. Ajtai [4] used a more general parameterized family of distributions, and we use a variant of the ones he used. For $q \in (0, \frac{1}{2}]$, let $\mathcal{F}_q$ be the distribution which chooses (F_1, F_2) by independently assigning each decision tree $T \in F$ as follows:

$$T \in \begin{cases} F_1 & \text{with probability } q \\ F_2 & \text{with probability } q \\ F - F_1 - F_2 & \text{with probability } 1 - 2q. \end{cases}$$

The distribution used in [8] corresponds to the case $q = 1/2$.

For $x \in D^n$, let $\mu(x, q) = E[|\texttt{core}(x, F_1)|] = E[|\texttt{core}(x, F_2)|]$ for (F_1, F_2) selected according to $\mathcal{F}_q$. We now show that $\mu(x, q)$ is a fairly large fraction of n, and also that for each x, with high probability, both $\texttt{core}(x, F_1)$ and $\texttt{core}(x, F_2)$ are close to $\mu(x, q)$. This lemma generalizes one proved in [8] for the $q = 1/2$ case.

Lemma 4. *Let $n \geq r \geq k$ and let F be an n-variate inquisitive $(r, k/r)$-decision forest. Let x be any input. For any q, if (F_1, F_2) is chosen according to $\mathcal{F}_q$, then:*
(a) $\mu(x, q) \geq q^k n$.
(b) for each $j \in \{1, 2\}$,
$\Pr[||\texttt{core}(x, F_j)| - \mu(x, q)| \geq \frac{1}{2}\mu(x, q)] \leq \frac{4k^2}{rq^k}$

Thus when (F_1, F_2) is chosen according to $\mathcal{F}_q$, for a "typical" $x \in F^{-1}(1)$, the rectangle $R = R(x, F_1, F_2)$ has $m_1(R)$ and $m_2(R)$ both close to $\mu(x, q) \geq q^k n$. This gives us a collection of rectangles with large feet that covers most of $F^{-1}(1)$. If we only cared about the foot size, then we would clearly choose $q = 1/2$, and we would essentially be done. However, we also want the rectangles in our family to have large leg-density.

In the special case that all of the trees in F are *oblivious* (that is, the choice of variables queried in a given tree depends only on the level and not on the path followed by the input) the value $q = 1/2$ suffices for large leg-density as well: In this case, the values of the cores do not depend on the choice of input and so, for any given pair (F_1, F_2), all of the rectangles in $\mathcal{R}(F_1, F_2)$ have the same pair of feet (A_1, A_2). Thus, by definition of $\mathcal{R}(F_1, F_2)$, these rectangles are determined only by their spines σ on $[n]-A_1-A_2$. It easily follows that for $\eta > 0$ all but at most $2\eta|D^n|$ points of $F^{-1}(1)$ are covered by rectangles of leg-density at least η since any rectangle R with $\alpha_j(R) \leq \eta$ covers at most $\eta|D|^{|A_1|+|A_2|}$ inputs and there are only $|D|^{n-|A_1|-|A_2|}$ rectangles in $\mathcal{R}(F_1, F_2)$.

In the general case, large leg-density is much harder to achieve and we need a more detailed understanding of leg-density. In a single class $R = R(x, F_1, F_2)$, all inputs agree on the spine $\sigma = \mathtt{stem}(x, F_1, F_2)$. In order for R to have large leg-density, both the $R_{\mathtt{core}(x,F_j)}$ must be large. In particular, it must be the case that for $j = 1, 2$, for many assignments τ_j to $\mathtt{core}(x, F_j)$, there are many inputs in F^{-1} that extend $\sigma\tau_j$ and have the same F_1- and F_2-cores as x. Observe that for any such assignment, τ_j, $\sigma\tau_j$ is the F_{3-j}-stem of many inputs y in R. Thus we consider how varying the inputs, subject to fixing their F_1- or F_2-stems, affects their F_1- and F_2-cores.

We simplify this by observing that fixing an input's F_j-stem also fixes its F_j-core:

Proposition 5. *For any subforest F' of F, if $x, y \in D^n$ have the same value on $\mathtt{stem}(x, F')$ then $\mathtt{core}(y, F') = \mathtt{core}(x, F')$ and $\mathtt{stem}(y, F') = \mathtt{stem}(x, F')$.*

Proof. Since x and y agree on all elements of $[n] - \mathtt{core}(x, F')$, the computations of all trees of F outside of F' are the same on x and y. Thus, in particular, $\mathtt{read}(y, F - F') = \mathtt{read}(x, F - F')$ and so $\mathtt{core}(y, F') = [n] - \mathtt{read}(y, F - F') = [n] - \mathtt{read}(x, F - F') = \mathtt{core}(x, F')$. It then follows that $\mathtt{stem}(y, F') = \mathtt{stem}(x, F')$ since they agree on $[n] - \mathtt{core}(y, F') = [n] - \mathtt{core}(x, F')$. □

Furthermore, we can analyze the inputs with different F_j-stems separately since the F_j-stems partition the set of inputs:

Proposition 6. *For any subforest F' of F, the set $\mathcal{E}(F') = \{D^n(\rho) : \exists x \in D^n, \rho = \mathtt{stem}(x, F')\}$ partitions D^n.*

Proof. Clearly every input $x \in D^n$ is an extension of its own F'-stem so every element of D^n is covered by an element of $\mathcal{E}(F')$. Furthermore, by Proposition 5, any input $y \in D^n(\rho)$ for $\rho = \mathtt{stem}(x, F')$ has $\mathtt{stem}(y, F') = \rho$ and thus the various sets $D^n(\rho)$ are disjoint. □

Thus, to show that the leg-density is large for rectangles associated with most inputs in some set $J \subseteq D^n$, it suffices to show that, for each $\rho = \mathtt{stem}(x, F_j)$ with $x \in J$, many inputs $y \in D^n(\rho)$ that are also in J have the same value of $\mathtt{core}(y, F_{3-j})$. There are clearly, *a priori*, at most $|D^n(\rho)| = |D|^{|\mathtt{unfixed}(\rho)|} = |D|^{|\mathtt{core}(x,F_j)|}$ such F_{3-j}-cores. The following lemma shows that one can obtain density bounds indirectly by showing that the number of distinct F_j-cores of such y is much less than $D^{|\mathtt{unfixed}(\rho)|}$. For ρ an F_j-stem of some input in J, let $\mathtt{numcores}_j(\rho, J)$ be the number of different F_{3-j}-cores of inputs in J that are extensions of ρ.

Lemma 7. *Let F be an n-variable inquisitive decision forest on domain D, let F_1, F_2 be subforests of F and $J \subseteq D^n$. Let $\eta \in [0, 1]$. Suppose that there is an integer function g such that for each $j \in \{1, 2\}$ and each ρ that is an F_j-stem of some input in J, $\mathtt{numcores}_j(\rho, J) \leq g(|\mathtt{unfixed}(\rho)|)$. Then the subcollection of rectangles $R \in \mathcal{R}(F_1, F_2)$ that satisfy $\alpha_j(R) \geq \frac{\eta}{g(m_j(R))}$ for $j \in \{1, 2\}$ covers all but at most $2\eta|D^n|$ points of J.*

Proof. Fix $j \in \{1, 2\}$. Call a rectangle $R \in \mathcal{R}(F_1, F_2)$ *bad* if $\alpha_j(R) < \frac{\eta}{g(m_j(R))}$. It suffices to show that the number of inputs in J that are in bad rectangles is at most $\eta|D^n|$.

For any $x \in F^{-1}$, let $Q(x)$ be the set of inputs $y \in F^{-1}$ such that $\mathtt{stem}(y, F_j) = \mathtt{stem}(x, F_j)$ and $\mathtt{core}(y, F_{3-j}) = \mathtt{core}(x, F_{3-j})$ and call the values of these quantities ρ_x and C_x, respectively. Observe that the sets $Q(x)$ partition F^{-1} and that the distinct sets may be specified unambiguously using the notation $Q(\rho_x, C_x)$ instead of $Q(x)$. By Proposition 5 and our discussion above, we have $Q(x) \subset R(x, F_1, F_2)$, $m_j(R(x, F_1, F_2)) = \mathtt{unfixed}(\rho_x)$, and $\alpha_j(R(x, F_1, F_2)) = |Q(x)|/|D^n(\rho_x)|$. Call the set $Q(x)$ *bad* if $|Q(x)| < \frac{\eta}{g(|\mathtt{unfixed}(\rho_x)|)}|D^n(\rho_x)|$. Thus if $x \in J$ is in a bad rectangle $R(x, F_1, F_2)$ then $Q(x)$ is bad; we therefore count the number of inputs in bad rectangles by counting the number in bad $Q(x)$.

Fix a ρ that is the F_j-stem of some input in J. We first count the number of inputs of J in bad sets of the form $Q(\rho, C)$. Each such bad set has at most $\frac{\eta}{g(|\mathtt{unfixed}(\rho)|)}|D^n(\rho)|$ elements by definition. By our assumption, $\mathtt{numcores}_j(\rho, J) \leq g(|\mathtt{unfixed}(\rho)|)$ which implies that there are at most $g(|\mathtt{unfixed}(\rho)|)$ different sets C such that $Q(\rho, C)$ contains any input in J. Thus the total number of inputs of J in $D^n(\rho)$ that are in bad sets is at most $\eta|D^n(\rho)|$.

By Proposition 6, the sets $|D^n(\rho)|$ partition D^n so the total number of inputs of J that are in bad sets is at most $\eta|D^n|$, as required. □

To apply this lemma for given J, F_1 and F_2, we need a bounding function g. For this, we want for $j \in \{1, 2\}$ and $\rho \in \mathtt{stem}(J, F_j)$ that the number of distinct sets $\mathtt{core}(x, F_j)$ with $x \in J \cap D^n(\rho)$ is small. As Ajtai did, we use the following simple observation:

Proposition 8. *If $\mathcal{C}$ is a collection of subsets of $[n]$ such that for any two sets $A, B \in \mathcal{C}$, the symmetric difference $A \Delta B$ has size at most d, then $|\mathcal{C}| \leq S(n, d)$, where $S(n, d) = \sum_{j \leq d} \binom{n}{d}$.*

We want to arrange that for any two inputs $x, y \in D^n(\rho)$ of interest, $\mathtt{core}(x, F_j) \Delta \mathtt{core}(y, F_j)$ has small size. Unfortunately, we cannot achieve this simply by choosing a single pair of disjoint subforests based on one value of q as in the oblivious case. Instead, we identify a partition of $F^{-1}(1)$ into a small number of sets J based on their "access" patterns and find a suitable pair of disjoint subforests using a different value of q for each such class. To this end, for fixed (F_1, F_2), it will be useful to define, for $j \in \{1,2\}$ and positive integer $\ell \leq r$:

$$\mathtt{acc}(x,\ell) = \{i \in [n] : \text{ on input } x, x_i \text{ is read in exactly } \ell \text{ trees of } F\}$$

$$B_j(x,\ell) = \mathtt{core}(x, F_j) - \mathtt{acc}(x,\ell)$$

$$B_j'(x,\ell) = \{i \in [n] : \text{ on input } x, x_i \text{ is read in exactly } \ell \text{ trees of } F_j, \text{ at least one tree of } F_{3-j}, \text{ and no trees of } F - F_1 - F_2.\}.$$

Lemma 9. *Let (F_1, F_2) be a pair of disjoint subforests of the forest F and let ℓ be a positive integer. For $j \in \{1,2\}$ and inputs $x, y \in D^n$ such that $\mathtt{stem}(x, F_{3-j}) = \mathtt{stem}(y, F_{3-j})$ we have*

$$\mathtt{core}(x,F_j)\Delta\mathtt{core}(y,F_j) \subseteq B_j(x,\ell) \cup B_j(y,\ell) \cup B_j'(x,\ell) \cup B_j'(y,\ell).$$

Proof. By symmetry in j, it suffices to consider $i \in \mathtt{core}(x, F_1) - \mathtt{core}(y, F_1)$ and show $i \in B_1(x,\ell) \cup B_1'(y,\ell)$.

If $i \notin \mathtt{acc}(x,\ell)$, then $i \in B_1(x,\ell)$. Suppose $i \in \mathtt{acc}(x,\ell)$. On input x, i is read by exactly ℓ trees in F_1, and by no trees of $F - F_1 - F_2$, and the same is true for y since x and y agree outside of $\mathtt{core}(x, F_2) = \mathtt{core}(y, F_2)$. Since $i \notin \mathtt{core}(y, F_1)$, at least one tree of F_2 reads i on input y, so $i \in B_j'(\ell, y)$. Therefore $i \in B_1(x,\ell) \cup B_1'(y,\ell)$. □

The following lemma shows that for each input x, we can choose $\ell = \ell(x)$ and $q = q(x)$ from a small set of values such that for (F_1, F_2) chosen according to $\mathcal{F}_q$, the expected sizes of $B_j(x,\ell)$ and $B_j'(x,\ell)$ $(j = 1,2)$ are substantially smaller than the expected core size, $\mu(x, q)$. Our bounds substantially improve those implicit in [4, 5] because we give a more precise description of these two quantities and give a sharper calculation of their expected sizes. Roughly speaking, in each case, the analysis in [4] only uses the randomness of one of the forests in the pair (F_1, F_2) whereas we use the randomness of both forests.

Lemma 10. *Let F be an n-variable inquisitive $(r, k/r)$-decision forest with $n \geq r \geq k \geq 3$. Let $q_1 \leq 1/4k$. For every input x, there is a pair $(\ell, b) = (\ell(x), b(x))$ of integers with $1 \leq \ell \leq k$ and $1 < b \leq 2k$, such that for (F_1, F_2) chosen according to $\mathcal{F}_{q_1^b}$ and $j \in \{1,2\}$,*
(a) $\mathrm{E}[|B_j(x,\ell)|] \leq 4q_1 \cdot \mu(x, q_1^b)$.
(b) $\mathrm{E}[|B_j'(x,\ell)|] \leq 2kq_1 \cdot \mu(x, q_1^b)$.

Proof. Let $\nu_h = |\mathtt{acc}(x,h)|$ for $h = 1, \dots, r$. It is easy to see that $\mu(x,q) = \sum_{h=1}^{r} \nu_h q^h$. We will choose ℓ and $q = q_1^b$ so that term $\nu_\ell q^\ell$ overwhelmingly dominates the sum.

For $a \geq 1$, let $q_a = q_1^a$. Let $h(a)$ be the least index such that $\nu_{h(a)} q_a^{h(a)} \geq \nu_h q_a^h$ for all $h \geq 1$. It is easy to show that $(h(a) : a \geq 1)$ is a decreasing sequence of positive integers with $h(a) \leq k$. By the pigeonhole principle, there exists a $b \in \{2, \dots, 2k\}$ such that $h(b-1) = h(b) = h(b+1)$. Set $\ell = \ell(x)$ to be $h(b)$ and let $b(x) = b$. By the choice of b, $\nu_h q_b^h \leq \nu_\ell q_b^\ell \cdot q_1^{|h-\ell|}$. Then, for (F_1, F_2) chosen according to $\mathcal{F}_{q_b}$,

$$\mathrm{E}[|B_j(x,\ell)|] = \sum_{h \neq \ell} \nu_h q_b^h \leq \sum_{h \neq \ell} \nu_\ell q_b^\ell \cdot q_1^{|\ell-h|} \leq 4q_1 \cdot \mu(x, q_b).$$

Note that $B_j'(x,\ell) \subseteq \cup_{h \geq \ell+1} \mathtt{acc}(x,h)$. For $h \geq \ell+1$ and $i \in \mathtt{acc}(x,h)$, $i \in B_j'(x,\ell)$ if and only if exactly ℓ out of the h trees that read i on x are in F_j and the rest are in F_{3-j}, which happens with probability $q_b^h \binom{h}{\ell} = q_b^h \frac{\ell+1}{1} \frac{\ell+2}{2} \cdots \frac{h}{h-\ell} \leq q_b^h (k+1)^{h-\ell}$, since $\ell \leq k$. Summing over $h > \ell$ and $i \in \mathtt{acc}(x,h)$,

$$\mathrm{E}[|B_j'(x,\ell)|] = \sum_{h=\ell+1}^{r} \nu_h q_b^h \binom{h}{\ell} \leq \sum_{h=\ell+1}^{r} \nu_h q_b^h (k+1)^{h-\ell} \leq 2kq_1 \mu(x, q_b).$$

□

For $b \in \{2, \dots, 2k\}$ and $\ell \in \{1, \dots, k\}$, let $C^{\ell,b} = \{x \in F^{-1}(1) : \ell(x) = \ell, b(x) = b\}$, and let $C^b = \cup_\ell C^{\ell,b}$. We now show that if $I \subseteq C^b$ for some $b \in \{2, \dots, 2k\}$, we can choose a pair of disjoint subforests (F_1^b, F_2^b) so that for most points x of I, the rectangle $R(x, F_1^b, F_2^b)$ is large.

Lemma 11. *Let F be an n-variable inquisitive $(r, k/r)$ decision forest with $n \geq r \geq k \geq 8$. Let $q_1 \leq 1/(4k)$, let $b \in \{2, \dots, 2k\}$ and let $q_b = q_1^b$. Let $I \subseteq C^b$. Let $\gamma, \delta > 0$, and suppose $r \geq \frac{4k^2}{\gamma q_b^k}$. Then there is a pair of forests (F_1, F_2) and a subset I' of I with $|I'| \geq |I|(1 - 6\gamma) - 2\delta|D|^n$ such that for each $x \in I'$ and $j \in \{1,2\}$ the rectangle $R = R(x, F_1, F_2)$ satisfies: $m_j(R) \in [\frac{\mu(x,q_b)}{2}, \frac{3\mu(x,q_b)}{2}]$ and $\alpha_j(R) \geq \frac{\delta}{kS(n, \frac{10kq_1}{\gamma} m_j(R))}$.*

Proof. Select (F_1, F_2) according to $\mathcal{F}_{q_b}$.

Let $z \in I$ and let $\ell = \ell(z)$. We claim that with probability at least $1 - 6\gamma$, the following three events hold for both $j \in \{1,2\}$.
(i) $\frac{1}{2}\mu(z, q_b) \leq |\mathtt{core}(z, F_j)| \leq \frac{3}{2}\mu(z, q_b)$,

(ii) $|B_j(z,\ell)| \leq 8q_1|\mathtt{core}(z,F_j)|/\gamma$.
(iii) $|B'_j(z,\ell)| \leq 4kq_1|\mathtt{core}(z,F_j)|/\gamma$.

For each j, Lemma 4 says that (i) fails with probability at most $4k^2/(rq_b^k)$, which is at most γ by hypothesis. Since (i) implies $\mu(z,q_b) \leq 2|\mathtt{core}(z,F_j)|$, Lemma 10 with Markov's inequality implies that, conditioned on (i), the failure probabilities of (ii) and (iii) are each at most γ. This proves the claim.

It follows that there is a fixed pair (F_1,F_2) and a $I'' \subseteq I$ with $|I''| \geq (1-6\gamma)|I|$, such that for each $z \in I''$, (i), (ii) and (iii) hold for $j = 1$ and $j = 2$. Note that (i) implies the desired bounds on $m_j(R(z,F_1,F_2))$.

For each $\ell \in [k]$, let $I''_\ell = \{x \in I'' : \ell(x) = \ell\}$. We now determine a function g that, for each ℓ separately, will allow us to apply Lemma 7 with $J = I''_\ell$, to show that most points of I''_ℓ are covered by rectangles that are sufficiently dense. Consider the F_1-stem ρ of some input in I''_ℓ and fix y in $D^n(\rho) \cap I''_\ell$. Lemma 9 with (ii) and (iii) above imply that for $x \in D^n(\rho) \cap I''_\ell$, $|\mathtt{core}(y,F_2)\Delta\mathtt{core}(x,F_2)| \leq \frac{(8k+16)q_1}{\gamma}|\mathtt{unfixed}(\rho)|$ which is at most $\frac{10kq_1}{\gamma}|\mathtt{unfixed}(\rho)|$ since $k \geq 8$. By Proposition 8, $\mathtt{numcores}_j(\rho,I''_\ell) \leq S(n,\frac{10kq_1}{\gamma}|\mathtt{unfixed}(\rho)|)$.

Now apply Lemma 7 with $g(m) = S(n,\frac{10kq_1}{\gamma}m)$ and $\eta = \delta/k$. This gives $I'_\ell \subseteq I''_\ell$ of size at least $|I''_\ell| - 2\delta|D|^n/k$, such that for every $x \in I'_\ell$, $\alpha_j(R(x,F_1,F_2)) \geq \delta/(k|D|^n S(n,\frac{10kq_1}{\gamma}|\mathtt{core}(x,F_j)|))$. Let $I' = \cup_{\ell=1}^k I'_\ell$. Then $|I'| \geq |I''| - 2\delta|D|^n \geq |I|(1-6\gamma) - 2\delta|D|^n$. □

Lemma 12. *Let F be an n-variable $(r,k/r)$ decision forest where $n \geq r \geq k \geq 8$ are integers. Let $q_1 \leq 1/(4k)$. Let J be a subset of $F^{-1}(1)$. Let $\gamma',\delta' > 0$, and suppose $r \geq \frac{48k^2}{\gamma' q_1^{2k^2}}$. Then there is a family $\mathcal{R}$ of rectangles each contained in $F^{-1}(1)$ such that $\bigcup_{R\in\mathcal{R}} R$ covers a subset J' of J of size at least $|J|(1-\gamma') - |D^n|\delta'$, and such that $\mathcal{R}$ can be partitioned into subcollectons $\{\mathcal{R}^b : b \in \{2,\ldots,2k\}\}$, where for each b, the rectangles in $\mathcal{R}^b$ are disjoint and each $R \in \mathcal{R}^b$ satisfies $m(R) = m_1(R) = m_2(R) \geq \frac{1}{2}q_1^{bk}$ and $\alpha_1(R),\alpha_2(R) \geq \frac{\gamma'\delta'}{8k^2 S(n,\frac{360kq_1}{\gamma'}m(R))}$.*

Proof. For each $b \in \{2,\ldots,2k\}$, apply the previous lemma with $I = I^b = J \cap C^b$ and $\gamma = \gamma'/12$ and $\delta = \delta'/4k$. Let (F_1^b,F_2^b) be the set of subforests and J^b be the set I' from the conclusion of the lemma. Let $\mathcal{Q}^b = \{R(x,F_1^b,F_2^b) : x \in J^b\}$. Let $J'' = \bigcup_{b=2}^{2k} J^b$ and $\mathcal{Q} = \bigcup_{b=2}^{2k}\mathcal{Q}^b$. Then $|J''| = \sum_b |J^b| \geq \sum_b \left(|I^b|(1-\gamma'/2) - |D^n|\delta'/2k\right) \geq |J|(1-\gamma'/2) - |D^n|\delta'$.

The rectangles in $\mathcal{Q}$ need not be balanced, so we use Proposition 1 on each rectangle of $\mathcal{Q}$. Let $(Q,A_1,A_2) \in \mathcal{Q}$, and without loss of generality, suppose that $|A_1| \geq |A_2|$. From the conclusion of lemma 11, we have $\alpha_1(Q),\alpha_2(Q) \geq \delta'/(4k^2 S(n,\frac{360kq_1}{\gamma'}|A_2|))$ since $|A_1| \leq 3|A_2|$. Choose $B_1 \subseteq A_1$ of size $|A_2|$ and apply Proposition 1 with $\delta = \gamma'/2$ to obtain a collection $\mathcal{R}(Q)$ of disjoint subrectangles of Q each with feet (B_1,A_2) that together cover at least $(1-\gamma'/2)|Q|$ points into subrectangles and such that each $R \in \mathcal{R}(Q)$ satisfies $\alpha_2(R) = \alpha_2(Q)$ and $\alpha_1(R) \geq \gamma'\alpha_1(Q)/2$. Take $\mathcal{R}$ to be the union of $\mathcal{R}(Q)$ over $Q \in \mathcal{Q}$, and J' to be the union of all the rectangles in $\mathcal{R}$, so $|J'| \geq |J''|(1-\gamma'/2) \geq |J|(1-\gamma') - |D^n|\delta'$. The conclusion of Lemma 11, together with the lower bound on $\mu(x,q)$ given by Lemma 4 implies that the rectangles in $\mathcal{R}$ have the claimed properties. □

3.4 Embedded Rectangles in Branching Programs

Theorem 13. *Let $k \geq 8$ be an integer, $q_1 \leq 2^{-30}k^{-6}$, $n \geq r \geq 200k^2/q_1^{4k^2}$. Let B be a branching program of length at most $(k-2)n$ and size 2^S and let $I \subseteq B^{-1}(1)$. There is a collection $\mathcal{R}$ of embedded rectangles each contained in $B^{-1}(1)$ that satisfies:*
1. Each rectangle of $\mathcal{R}$ is contained in $B^{-1}(1)$.
2. $\bigcup_{R\in\mathcal{R}} R$ covers at least $|I|/2$ inputs of I.
3. No input belongs to more than $2k-1$ rectangles of $\mathcal{R}$.
4. Each rectangle $R \in \mathcal{R}$ satisfies $m_1(R) = m_2(R) \geq q_1^{2k^2}n/2$ and $\alpha(R) \geq 2^{-q_1^{1/2}m-Sr}|I|/|D|^n$.

Proof. Let B' be the length $(k-1)n$ inquisitive branching program obtained from B by adding n layers at the begining that obliviously query each variable. By Lemma 2, there is a family $\mathcal{S}$ consisting of 2^{Sr} $(r,k/r)$ decision forests, such that $B = \bigwedge_{F\in\mathcal{S}} F$. As remarked after that lemma, each of the decision forests is inquisitive. Note that the collection of sets $\{F^{-1}(1) : F \in \mathcal{S}\}$ partitions $B^{-1}(1)$.

For each forest $F \in \mathcal{S}$, apply Lemma 12 with $J = F^{-1}(1) \cap I$, $\gamma' = 1/4$ and $\delta' = |I|/(2^{Sr+2}|D^n|)$, and let $\mathcal{R}_F$ be the family of embedded rectangles obtained in the conclusion of the lemma. Define $\mathcal{R} = \bigcup_{F\in\mathcal{S}}\mathcal{R}_F$. We claim that $\mathcal{R}$ satisfies the conditions asserted in the lemma.

The rectangles of $\mathcal{R}_F$ are contained in $F^{-1}(1)$ so every $R \in \mathcal{R}$ is contained in $B^{-1}(1)$. Since no input is covered by more than k rectangles in $\mathcal{R}_F$ and the sets covered by $F^{-1}(1)$ are disjoint for distinct $F \in \mathcal{S}$ and each input is covered at most k rectangles of $\mathcal{R}$. For each F, $\mathcal{R}(F)$ covers at least $\frac{3}{4}|F^{-1}(1)\cap I| - |I|/2^{Sr+2}$ points of $F^{-1}(1)$, so summing over at most 2^{Sr} different F, we have that $\mathcal{R}(F)$ covers at least $|I|/2$ points of I.

Again by the conclusion of Lemma 12 each $R \in \mathcal{R}_F$ has $m(R) = m_1(R) = m_2(R) \geq q_1^{2k^2}n/2$ and

$\alpha(R) \geq |I|/(2^{Sr}|D^n|128k^2S(n, 1440kq_1m(R)))$. A routine calculation yields the desired lower bound on $\alpha(R)$ using our hypotheses on k, q_1, r, and n. □

We can strengthen the above theorem to show that there is a collection of rectangles that covers almost all of I but the strengthening complicates the analysis and is unnecessary for obtaining our lower bounds.

The first part of the following corollary is a quantitative strengthening of Ajtai's main technical result; the second part extends this to branching programs that are allowed to make a small fraction of errors.

Corollary 14. *Let $k \geq 8$ be an integer, $q_1 \leq 2^{-30}k^{-6}$, $n \geq r \geq q_1^{-5k^2}$. Let B be an n-variate branching program over domain D of length $\leq (k-2)n$ and size 2^S.*
1. There is an embedded rectangle R contained in $B^{-1}(1)$ satisfying $m_1(R) = m_2(R) \geq q_1^{2k^2}n/2$ and $\alpha(R) \geq 2^{-q_1^{1/2}m-Sr}|B^{-1}(1)|/|D|^n$.
2. Let f is an n-variate decision function over D and suppose B agrees with f on at least $(1-\epsilon)|D^n|$ inputs. Let $\delta \leq |f^{-1}(1)|/|D^n|$. Then there is an embedded rectangle R contained in $B^{-1}(1)$ satisfying $m_1(R) = m_2(R) \geq q_1^{2k^2}n/2$ and $\alpha(R) \geq 2^{-q_1^{1/2}m-Sr}(\delta-\epsilon)$, such that f is 0 on at most a $4\epsilon k/(\delta-\epsilon)$ fraction of points in R.

Proof. Apply Theorem 13 with $I = B^{-1}(1)$ (noting that the lower bound on r in the hypothesis of the present theorem implies the hypothesis on r for that theorem) and let $\mathcal{R}$ be the resulting collection of rectangles. The first part of the corollary follows by choosing any R in $\mathcal{R}$. For the second part, note that the hypotheses imply that $|B^{-1}(1)|/|D^n| \geq \delta - \epsilon$, so all rectangles in $\mathcal{R}$ satisfy $\alpha(R) \geq 2^{-q_1^{1/2}m-Sr}(\delta-\epsilon)$, and together the rectangles cover at least $(\delta-\epsilon)|D^n|/2$ inputs of $B^{-1}(1)$. Call an input x *bad* if $B(x) \neq f(x)$ and for $R \in \mathcal{R}$, let $Bad(R)$ be the set of bad inputs of R. Now $\sum_{R\in\mathcal{R}}|Bad(R)| \leq 2k\epsilon|D^n|$ since each input appears in at most $2k$ rectangles. Also $\sum_{R\in\mathcal{R}}|Bad(R)| \geq \min_R \frac{|Bad(R)|}{|R|}\sum_{R\in\mathcal{R}}|R| \geq \min_R \frac{|Bad(R)|}{|R|}\frac{\delta-\epsilon}{2}|D^n|$. So the rectangle minimizing $|Bad(R)|/|R|$ satisfies $\frac{|Bad(R)|}{|R|} \leq 4\epsilon k/(\delta-\epsilon)$. □

4 Lower Bounds

4.1 Element Distinctness

In this section we consider the element distinctness function $ED : D^n \to \{0,1\}$ which is 1 if and only if there is no pair $i \neq j \in X$ such that $x(i) = x(j)$. By simple calculation one can show that if $|D| \geq n^2 - n$, at least a $1/e$ fraction of all inputs $x \in D^n$ have $ED(x) = 1$. We first use Ajtai's argument to obtain a lower bound for deterministic branching programs computing ED.

Lemma 15. *Let $ED : D^n \to \{0,1\}$. Any embedded rectangle $R \subseteq D^n$ such that $ED(x) = 1$ for all $x \in R$ has $\alpha(R) \leq 2^{-m(R)}$.*

Proof. Let A_1, A_2 be the feet of R, and for $j \in \{1,2\}$, let $S_j = \cup_{i\in A_j}R_i$ (where R_i is the set of elements of D that appear in coordinate i of some point of R). $ED(x) = 1$ for all $x \in R$ implies $S_1 \cap S_2 = \emptyset$, so for some index h $|S_h| \leq |D|/2$. Thus $\alpha_h(R) \leq (|S_h|/|D|)^{m_h(R)} \leq \frac{1}{2}^{m(R)}$. □

Theorem 16. *There is a constant $c > 0$ such that any $[1,n^2]$-way deterministic branching program computing $ED : [1,n^2]^n \to \{0,1\}$ in time T and size 2^S requires $T = \Omega(n\sqrt{\log(n/S)/\log\log(n/S)})$.*

Proof. Suppose we have a branching program of length $(k-2)n$ and size $s = 2^S$ for ED. Apply Corollary 14(i) with $q_1 = 2^{-30}k^{-6}$ and $r = \lceil q_1^{-5k^2}\rceil$. We obtain an embedded rectangle on which B outputs 1 such that $m \geq q_1^{2k^2}n/2$ and $\alpha \geq 2^{-q_1^{1/2}m-Sr}/e > 2^{-q_1^{1/2}m-Sr-2}$. Using Lemma 15, this means $2^{-q_1^{1/2}m-Sr-2} \leq 2^{-m}$ and thus $Sr \geq m(1-q_1^{1/2}) - 2 \geq q_1^{2k^2}n/4$ or $S \geq q_1^{2k^2}n/(4r)$. Thus for some constant $c > 0$ any algorithm solving ED in time kn requires space at least $k^{-ck^2}n$. Substituting $T = (k-2)n$, and re-arranging we obtain the claimed tradeoff. □

We now consider randomized branching programs for ED. We will use the second part of Corollary 14, but first we need to show that any rectangle on which ED is mostly 1 can not be very dense.

Lemma 17. *If (R, A_1, A_2) is an n-variate embedded rectangle over $[n^2]$ with $|A_1| = |A_2| = m$ such that at most a $1/12$ fraction of $x \in R$ have $ED(x) = 0$ then $\alpha(R) \leq (8/9)^{m/2}2^{2+n/m}$.*

Proof sketch. Observe that if $x \in R$ has $ED(x) = 1$ then x_{A_1} and x_{A_2} must be disjoint sets. We derive our bound by adapting the proof of a lower bound for ϵ-error communication complexity of the set-disjointness problem due to Babai, Frankl, and Simon [6]. □

Theorem 18. *There is a constant $c > 0$ such that any randomized $[1,n^2]$-way branching program computing $ED : [1,n^2]^n \to \{0,1\}$ in time T and size 2^S with probability of error at most $n/200T$ requires $T = \Omega(n\sqrt{\log(n/S)/\log\log(n/S)})$.*

Proof. It suffices to prove the lower bound for deterministic branching programs that approximate ED within error ϵ.

Choose n to be a sufficiently large integer. We will apply the second part of Corollary 14 and to this end, we assume without loss of generality that $k \geq 8$ and define for $q_1 = 2^{-30}k^{-6}$, and $r = \lceil q_1^{-5k^2} \rceil$. Let B be a deterministic branching program of length at most $(k-2)n$ and size 2^S that approximates ED_n within $1/200(k-2)$. We will show that for some $c > 0$, $k \geq c\sqrt{\log(n/S)/\log\log(n/S)}$. If $n < r^2$ this is immediate, so assume $n > r^2$.

Applying Corollary 14(ii), we obtain a balanced rectangle R with $m(R) \geq q_1^{2k^2}n/2$ and $\alpha(R) \geq 2^{-Sr-q_1^{1/2}m-2}$ such that ED is 0 for at most a fraction $\frac{4k}{200(k-2)}/(1/e - 1/100(k-2)) \leq 1/12$ since $k \geq 8$. Applying the previous lemma, we have that $\alpha(R) \leq 2^{2+n/m}(8/9)^m$, so combining the upper and lower bounds on $\alpha(R)$ and simplifying we get $2^{Sr} \geq 2^{-q_1^{1/2}m-4-n/m}(9/8)^m$, which, for n sufficiently large and k satisfying the restrictions above, is at least $(10/9)^m$. From this we deduce $S \geq \log(10/9)\frac{m}{r}$ which is at least $c_0 n/k^{c_1 k^2}$ for some c_0, c_1 independent of n and k. It follows that for some constant $c > 0$ and sufficiently large n, $k \geq \sqrt{\log(n/S)/\log\log(n/S)}$. □

Corollary 19. *There is a constant $c > 0$ so that for any $\delta \geq 0$ there is a constant c_δ such that any randomized RAM algorithm for element distinctness on inputs in the range $[1, n^2]$ taking at most $c_\delta n\sqrt{\log n/\log\log n}$ time and having at most $c_\delta\sqrt{\frac{\log\log n}{\log n}}$ error requires at least $n^{1-\delta}$ space.*

4.2 Boolean Branching Programs Computing Quadratic Forms

If D is a finite field and M is an $n \times n$ matrix with entries in D, let F_M denote the function on D^n given by $F_M(x) = x^T Mx$. By considering such functions in the case $D = GF(2)$, Ajtai constructed an explicit family of boolean functions which can not be computed by a deterministic branching program of subexponential size and linear length. In this subsection we extend this result to randomized branching programs.

Using ideas of Borodin, Razborov, Smolensky [13], Beame, Saks, and Thathachar had shown (in the case where $D = GF(q)$ for odd q) that if M is a symmetric matrix that is *rigid* in the sense that all sub-matrices of M have rank that is suitably large relative to their size, then $x^T Mx$ can not be constant on any large embedded rectangle. This result does not hold when $q = 2$, and Ajtai developed a variant that holds for this case. Lemma 20 encapsulates and strengthens Ajtai's argument.

Lemma 20. *Let M be a $n \times n$ matrix with entries $GF(2)$ and suppose that (R, A_1, A_2) is an embedded rectangle in $GF(2)^n$ with $|A_1| = |A_2|$. Let P be the submatrix of $M + M^T$ induced on $A_1 \times A_2$. Suppose that $\alpha(R) \geq 2^{2-rank(P)/2}$. Then for each $b \in GF(2)$, the fraction of inputs of $x \in R$ for which $x^T Mx = b$ is at least $1/16$.*

Combining this lemma with Corollary 14 gives the following result which says that if M is a matrix whose quadratic form function is well approximated by a small branching program then M must have a large submatrix of small rank, that contains no entry on the diagonal.

Theorem 21. *Let n, r, k be positive integers and $q_1 > 0$ with $k \geq 8$, $q_1 \leq 2^{-30}k^{-6}$, $n \geq r \geq q_1^{-5k^2}$. Let M be an $n \times n$ matrix with entries in $GF(2)$, with associated quadratic form function f. Suppose that B is an n-variate branching program over $GF(2)$ of length at most $(k-2)n$ and size s that disagrees with f on at most a fraction $1/160k$ inputs. Then there are two disjoint subsets $A_1, A_2 \subseteq [n]$ with $|A_1| = |A_2| = m$ where $m \geq q_1^{2k^2}n/2$ such that the submatrix of $P = M + M^T$ induced by $A_1 \times A_2$ has rank at most $2Sr + 2q_1^{1/2}m + 8$.*

Proof. Let $b \in GF(2)$ be such that $|f^{-1}(b)| \geq 2^{n-1}$. Define the function f' by $f'(x) = f(x) + b - 1$ and define the branching program B' analogously from B by replacing output 0 by $b - 1$ and output 1 by b.

Applying the second part of Corollary 14 to f' and B' with $\delta > 1/2$ and $\epsilon = 1/160k$, we get a balanced rectangle $R = (R, A_1, A_2)$ contained in $B^{-1}(1)$ satisfying $m(R) = |A_1| = |A_2| \geq q_1^{2k^2}n/2$ and $\alpha(R) \geq 2^{-q^{1/2}m(R)-Sr}(\frac{1}{2} - \frac{1}{160k}) \geq 2^{-q^{1/2}m(R)-Sr-2}$ such that f' is 0 on at most $4(1/160k)k/(1/2 - 1/160k) < 1/16$ fraction of points of R. By Lemma 20, $\alpha(R)$ must be less than $2^{2-rank(P)/2}$. Combining the upper and lower bounds on $\alpha(R)$ we deduce $rank(P) \leq 2Sr + 2q_1^{1/2}m(R) + 8$. □

This theorem can be applied to give time-space tradeoffs for the quadratic form function for any matrix M for which $M + M^T$ has the property that every large submatrix that avoids the diagonal has large enough rank. The Sylvester matrices considered in [13, 8] have the property that for $\beta \in [0, 1]$ every $\lceil \beta n \rceil \times \lceil \beta n \rceil$ submatrix has rank at least $\beta^2 n$. This is not strong enough to get good tradeoffs from Theorem 21.

Instead Ajtai looks at Hankel matrices, matrices whose every anti-diagonal is constant. Given a vector $y \in GF(2)^{2n-1}$, define the Hankel matrix $H[y]$ whose i, j entry is $H[y]_{i,j} = y_{i+j-1}$. Ajtai proved the following lemma concerning the rigidity properties of random Hankel matrices over $GF(2)$. (Here a random Hankel matrix means a matrix $H[y]$ where y is chosen uniformly at random from $GF(2)^{2n-1}$.)

Lemma 22. *Assume that n, s, R, t are positive integers, $t^2 < s < n$, $R < Q = \lfloor s/t^2 \rfloor$. If H is a random $n \times n$ Hankel matrix over $GF(q)$, the probability that there is some $s \times s$ sub-matrix of H of rank less than R is at most $\binom{n}{Qt}^2 \binom{Q}{Q-R+1} q^{-\frac{1}{4}(Q-R+1)t^2}$.*

We restate this in a more convenient form, making no attempt to optimize constants.

Corollary 23. *Let n be an integer and H be a random a random $n \times n$ Hankel matrix over $GF(2)$. With probability at least 1/2, for all integers s satisfying $(1024 + 64 \log n)^2 < s < n$ every $s \times s$ submatrix of H has rank at least $s/(2048 + 128 \log(n/s))^2 - 2$.*

The rigidity property of random Hankel matrices above is strong enough to be useful. However, there are two minor problems in making use of this. In Theorem 21, to prove a tradeoff for the function F_M, we need that $M + M^T$ be rigid. Hankel matrices are symmetric which means that $M + M^T = 0$, since we are over $GF(2)$. (In [13] and [8] this problem was avoided by working over fields of odd characteristic.) Following Ajtai, define $L(M)$ to be the lower triangular matrix obtained by changing all entries of M that are on or above the diagonal to 0. Then $L(M) + L(M)^T$ agrees with M except on the diagonal, and if M is sufficiently rigid, we can use Theorem 21 to get a time-space tradeoff for the quadratic form associated with $L(M)$.

The second problem is that we want lower bounds for explicit functions, and a random Hankel matrix does not give an explicit function. But, since a Hankel matrix is specified by only $2n - 1$ values, we can prove lower bounds on the explicit function $G_n(x, y)$ where $x \in GF(2)^n$ and $y \in GF(2)^{2n-1}$, which is defined to be $x^T M x$ where $M = L(H[y])$.

Theorem 24. *There is a constant $c' > 0$ such that any randomized Boolean branching program computing $G_n(x, y)$ in time T and size 2^S with probability of error at most $c'n/T$ requires $T \geq c'n\sqrt{\log(n/S)/\log\log(n/S)}$.*

References

[1] K. R. Abrahamson. A time-space tradeoff for Boolean matrix multiplication. In *Proceedings 31st Annual Symposium on Foundations of Computer Science*, pages 412–419, St. Louis, MO, Oct. 1990. IEEE.

[2] K. R. Abrahamson. Time–space tradeoffs for algebraic problems on general sequential models. *Journal of Computer and System Sciences*, 43(2):269–289, Oct. 1991.

[3] M. Ajtai. Determinism versus non-determinism for linear time RAMs with memory restrictions. Technical Report TR98-077, Electronic Colloquium in Computation Complexity, `http://www.eccc.uni-trier.de/eccc/`, 1998. Revision 1.

[4] M. Ajtai. Determinism versus non-determinism for linear time RAMs with memory restrictions. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing*, 1999.

[5] M. Ajtai. A non-linear time lower bound for boolean branching programs. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*. IEEE, 1999.

[6] L. Babai, P. Frankl, and J. Simon. Complexity classes in communication complexity theory. In *27th Annual Symposium on Foundations of Computer Science*, pages 337–347, Toronto, Ontario, Oct. 1986. IEEE.

[7] P. W. Beame. A general time-space tradeoff for finding unique elements. *SIAM Journal on Computing*, 20(2):270–277, 1991.

[8] P. W. Beame, M. Saks, and J. S. Thathachar. Time-space tradeoffs for branching programs. In *Proceedings 39th Annual Symposium on Foundations of Computer Science*, pages 254–263, Palo Alto, CA, Nov. 1998. IEEE.

[9] A. Borodin. Time space tradeoffs (getting closer to the barrier?). In *4th International Symposium on Algorithms and Computation*, pages 209–229, Hong Kong, Dec. 1993.

[10] A. Borodin and S. A. Cook. A time-space tradeoff for sorting on a general sequential model of computation. *SIAM Journal on Computing*, 11(2):287–297, May 1982.

[11] A. Borodin, F. E. Fich, F. Meyer auf der Heide, E. Upfal, and A. Wigderson. A time-space tradeoff for element distinctness. *SIAM Journal on Computing*, 16(1):97–99, Feb. 1987.

[12] A. Borodin, M. J. Fischer, D. G. Kirkpatrick, N. A. Lynch, and M. Tompa. A time-space tradeoff for sorting on non-oblivious machines. *Journal of Computer and System Sciences*, 22(3):351–364, June 1981.

[13] A. Borodin, A. A. Razborov, and R. Smolensky. On lower bounds for read-k times branching programs. *Computational Complexity*, 3:1–18, Oct. 1993.

[14] L. Fortnow. Nondeterministic polynomial time versus nondeterministic logarithmic space: Time-space tradeoffs for satisfiability. In *Proceedings, Twelfth Annual IEEE Conference on Computational Complexity*, pages 52–60, Ulm, Germany, 24–27 June 1997. IEEE Computer Society Press.

[15] L. Fortnow and D. van Melkebeek. Time-space tradeoffs for nondeterministic computation. In *Proceedings, Fifteenth Annual IEEE Conference on Computational Complexity*. IEEE Computer Society Press, July 2000. To appear.

[16] R. Lipton and A. Viglas. Time-space tradeoffs for sat. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*. IEEE, 1999.

[17] Y. Mansour, N. Nisan, and P. Tiwari. The computational complexity of universal hashing. *Theoretical Computer Science*, 107:121–133, 1993.

[18] A. C. Yao. Near-optimal time-space tradeoff for element distinctness. In *29th Annual Symposium on Foundations of Computer Science*, pages 91–97, White Plains, NY, Oct. 1988. IEEE.