Time-space tradeoff lower bounds for randomized computation of decision problems*

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

October 24, 2002

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 extension of those used by Ajtai [Ajt99a, Ajt99b] and by Beame, Jayram[1], and Saks [BST98, BJS01] that applied to deterministic branching programs. Our results also give a quantitative improvement over the previous results.

Previous time-space tradeoff results for decision problems can be divided naturally into results for functions with *boolean domain*, that is, each input variable is $\{0,1\}$-valued, and the case of *large domain*, where each input variable takes on values from a set whose size grows with the number of variables.

In the case of boolean domain, Ajtai exhibited an explicit class of functions, and proved that any deterministic Boolean branching program or RAM 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 (for randomized branching programs with error) 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 $\Omega(n\sqrt{\log n/\log\log n})$.

In the large domain case, we prove lower bounds of the form $T = \Omega(n\log(n/S))$ for randomized computation of decision problems for the element distinctness function, and for certain functions associated to quadratic forms over large fields. These bounds improve on previous results of Beame, Jayram, and Saks [BST98, BJS01], Ajtai [Ajt99a], and Pagter [Pag01].

*A preliminary version of this paper appeared in the Proceedings of the 41st IEEE Symposium on Foundations of Computer Science

† Research supported by NSF grants CCR-9800124 and CCR-0098066.

‡Research supported by NSF grants CCR-9700239, CCR-9988526, and by DIMACS

[1]T.S Jayram, formerly Jayram S. Thathachar

1 Introduction

The efficiency of an algorithm is typically measured according to its use of some relevant computational resource. The most widely studied resource in this context is *computation time*, but another important resource is memory or *computation space*. Typically, algorithmic design problems focus on the goal of minimizing one of these resources. It is very natural to study the relationship between these two goals.

It is well known that these goals are somewhat compatible; if we have an upper bound of S on the amount of space used by a terminating algorithm, then that algorithm has at most 2^S distinct memory configurations and therefore runs in time at most 2^S. This observation shows that a very space-efficient algorithm is at least somewhat time efficient.

Typically, this 2^S upper bound on time is very weak, and there are algorithms having much better time bounds. Indeed, for many fundamental computational problems such as sorting, matrix multiplication, and directed graph connectivity, the goals of minimizing time and space seem to be in conflict; the most time-efficient algorithms known require heavy memory resources, and as one decreases the amount of memory used, the amount of time needed to solve the problem apparently increases significantly. This apparent trade-off between time and space has motivated a large body of research within complexity theory [Bor93]. Such research has a dual motivation. First, we seek to provide a sound basis for the belief that such tradeoffs are inherent, and to understand the underlying characteristics of problems that exhibit such tradeoffs. Second, such research fits into the broader goal of proving computational lower bounds. Since we have had only very limited success in proving lower bounds on the time needed to solve a particular computational problem, or on the space needed to solve a particular computational problem, one might hope to make progress by considering the simultaneous restriction of time and space.

As with most lower bound problems in complexity theory, research divides into *uniform* and *nonuniform* models. In the uniform computational setting, an algorithm is modeled by a single program or, more formally, by a Turing machine, that operates on inputs of all lengths. In the nonuniform setting, an algorithm is modeled by a sequence of simple combinatorial structures (typically, directed graphs), one for each input size. A further dichotomy is drawn between *decision* problems (whose output is a single bit, indicating "Yes" or "No") and *multi-output* problems.

In the uniform setting, a series of recent papers have established time-space limitations on Turing machines that are able to solve the CNF-satisfiability (SAT) decision problem. The first work along these lines was by Fortnow [For97], which was followed by [LV99] and [FvM00]. 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 is the value labelling 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 that permits outputs along arcs during the course of computation), there was considerable success in proving time-space tradeoff lower bounds for *multi-output func-*

tions such as sorting, pattern matching, matrix-vector product and hashing [BC82, Bea91, Abr90, Abr91, MNT93]. The basic technique is to consider a space-limited computation, and show that in any short span of time, it is impossible to accurately produce more than a very small amount of the output. This technique is inherently incapable of providing results in the case of decision problems, where the entire output is a single bit.

Until recently, the only time-space tradeoff results for decision problems were for models where the access to the input was limited in some significant way. In the *comparison branching program model* (where the inputs are numbers, and the only access to the input allowed is pairwise comparison to determine order) strong time-space tradeoffs were obtained for the element distinctness decision problem [BFMadH$^+$87, Yao88]. There is also an extensive literature on various restricted *read-k* models ([BRS93, Oko93]) 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, [BST98, BJS01] 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 $\Omega(n \log\log n)$. (As we discuss later, the technique is powerful enough to show length lower bounds of $\Omega(n \log n)$ for sub-exponential size branching programs.) 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 $(1+\epsilon)n$. The lower bounds in [BST98, BJS01] were shown for functions based on quadratic forms over finite fields extending techniques of Borodin, Razborov, and Smolensky [BRS93] that showed size lower bounds for read-k branching programs computing bilinear forms.

In a remarkable breakthrough, Ajtai [Ajt99b] 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 [Ajt99a, Ajt98] 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: (1) Hamming closeness – determine whether the list contains a pair of strings within Hamming distance δb for some fixed $\delta > 0$, and (2) Element distinctness – determine whether the strings are all distinct. Ajtai's proof of the lower bound for Hamming closeness used ideas similar to those used by Okol'nishnikova [Oko93] to prove lower bounds in the read-k case; however, his argument for element distinctness contains deeper ideas that are the key to his lower bounds for Boolean branching programs.

The basic approach of all of these time-space tradeoffs for decision problems on branching programs 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 exhibit concrete functions that accept no large embedded rectangles. (We will define embedded rectangle in section 2.1; for now it suffices for the reader to know that it is a highly structured subset of D^n.) This was done for syntactic read-k branching programs in [BRS93, Oko93]. The first lower bounds on embedded rectangle size for general branching programs of small size and length were shown in [BST98, BJS01]. These bounds gave the results from that paper mentioned above, and are also strong enough to give the Hamming closeness result of [Ajt99a], but were not strong enough to give the element distinctness and Boolean function lower bounds. Ajtai obtained these bounds by proving a striking sequence of combinatorial lemmas that gave a 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. We also extend the lower bound technique of Beame, Jayram, and Saks to randomized branching programs. Since the branching program model is stronger than the RAM model our results apply to (two-sided error) randomized RAM algorithms as well.

We obtain substantial quantitative improvement over the previous results. More specifically, we show that, for element distinctness and the boolean quadratic form considered by Ajtai, any two-sided error 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.

For functions whose variables take on values from a large domain, stronger lower bounds were already known, and we improve on these slightly. For certain quadratic forms over larger fields, an $\Omega(n\log\log n)$ lower bound on length for deterministic branching programs of subexponential size was proved in [BST98, BJS01]. The same techniques can be applied to the natural generalizations of the quadratic forms considered by Ajtai to large domains, to immediately yield $\Omega(n\log n)$ length lower bounds for deterministic branching programs of subexponential size. We obtain the same bound for two-sided error randomized branching programs. For the Hamming closeness problem, Pagter [Pag01] had obtained an $\Omega(n\frac{\log n}{\log\log n})$ lower bound for one-sided error randomized branching programs of subexponential size by careful analysis of Ajtai's argument [Ajt98, Ajt99a]. We improve this to an $\Omega(n\log n)$ lower bound that again holds for two-sided error branching programs.

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

One superficial difference in our presentation that makes some of the exposition simpler is that we apply the basic approach developed in [BST98, BJS01] 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 carrying the space restriction throughout the argument. Our approach simplifies 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, Hamming closeness, and the quadratic forms to show that, not only do the functions not accept any relatively large embedded rectangle, they reject a significant fraction of inputs in any such rectangle.

2 Preliminaries

2.1 Sets and functions

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

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 $\{x \in D^N : x_{\mathtt{fixed}(\sigma)} = \sigma\}$.

A function whose range is $\{0,1\}$ is a *decision function*. A decision function whose domain is $\{0,1\}^N$ for some index set N is a *Boolean function*.

2.2 Embedded Rectangles

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 $\tau_1\tau_2\sigma \in B$. B is called the *body* of R and A_1 and A_2 are the *feet* of R. The sets B_{A_1} and B_{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 sometimes omit the word "embedded" and simply say that R is a rectangle.

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:

- $m_j(R) = |A_j|$,
- $m(R) = \min\{m_1(R), m_2(R)\}$,
- $\alpha_j(R) = |R_{A_j}|/|D|^{|A_j|}$,
- $\alpha(R) = \min\{\alpha_1(R), \alpha_2(R)\}$.

$\alpha(R)$ is called the *leg-density* of R and $\alpha_j(R)$ is called the *j-density* of R for $j = 1, 2$. Let $m \in [n]$, $\epsilon \in [0, 1]$ and $\lambda : [n] \longrightarrow [0, 1]$. We say that R is:

- *c-balanced* if $m_1(R) \le cm_2(R)$ and $m_2(R) \le cm_1(R)$.
- *balanced* if it is 1-balanced, i.e., $m_1(R) = m_2(R)$.
- *λ-dense* if $\alpha(R) \ge \lambda(m(R))$ and *λ-sparse* otherwise.
- *(m, λ)-large* if $m(R) \ge m$, and R is λ-dense.

Let (R, A_1, A_2) be a rectangle with legs $Y_1 = R_{A_1}$ and $Y_2 = R_{A_2}$ and spine σ. Let $\Pi_1 = R_{A_1 - B_1}$ and $\Pi_2 = R_{A_2 - B_2}$. For each $\pi_1 \in \Pi_1$ and $\pi_2 \in \Pi_2$, the set $R(\pi_1\pi_2) = R \cap D^n(\pi_1\pi_2)$ is a rectangle with feet (B_1, B_2), spine $\sigma\pi_1\pi_2$, and legs $Y_j(\pi_j) = Y_j \cap D^n(\pi_j)$, for $j \in \{1, 2\}$. The collection of rectangles $\{(R(\pi_1\pi_2), B_1, B_2) : \pi_1 \in \Pi_1, \pi_2 \in \Pi_2\}$ partitions R and is called the *(B_1, B_2)-refinement of* R, and is denoted $\mathrm{Refine}(R, B_1, B_2)$.

2.3 Branching programs

Since we are only interested in the computation of decision (single output) functions here, we present our definitions of branching programs only for this case. A *(deterministic) branching program* $\mathcal{B}$ on domain D and index set N is an acyclic directed graph with the following properties:

- There is a unique source node, denoted $start_{\mathcal{B}}$.
- Each sink node v has a label $output(v)$, which is 0 or 1.
- Each non-sink node v is labeled by an index $i(v) \in N$
- There are exactly $|D|$ arcs out of each non-sink node, each with a different element $value(a)$ of D.

Intuitively, a branching program is executed on input x by starting at $start_{\mathcal{B}}$, reading the variable $x_{i(start_{\mathcal{B}})}$ and following the unique arc labeled by $x_{i(start_{\mathcal{B}})}$. This process is continued until a sink is reached and the output of the computation is the output value of the sink.

We say that $\mathcal{B}$ *accepts* the input x if the sink reached on input x is labeled 1. We view $\mathcal{B}$ as a decision function from D^n by defining $\mathcal{B}(x) = 1$ if and only if $\mathcal{B}$ accepts x. For a function $f : D^N \longrightarrow \{0, 1\}$, we say that $\mathcal{B}$ computes f if $\mathcal{B}(x) = f(x)$ for all x and that $\mathcal{B}$ *approximates f with error at most* ϵ if the fraction of inputs x such that $f(x) \ne \mathcal{B}(x)$ is at most ϵ.

Two measures associated with $\mathcal{B}$ are *size* which equals the number of nodes, and *length* which is the length of the longest path.

A branching program of length d is *leveled* if the nodes can be partitioned into d sets $V_0, V_1, \ldots, V_d$ where $V_0 = \{start_{\mathcal{B}}\}$ is the source, V_d is the set of sink nodes and every arc out of V_i goes to V_{i+1}, for $0 \le i < d$. By a well-known observation (see, e.g. [BFK+81]), every branching program $\mathcal{B}$ of size s and length d, can be converted into a leveled branching program $\mathcal{B}'$ of length d that has at most s nodes in each of its levels and computes the same function as $\mathcal{B}$ (and is deterministic if $\mathcal{B}$ is).

For our purposes, a *randomized* branching program $\widetilde{\mathcal{B}}$ with domain D and index set N is a probability distribution over deterministic branching programs with domain D and index set N. Executing $\widetilde{\mathcal{B}}$ on input $x \in D^N$ corresponds to selecting the deterministic branching program $\mathcal{B}$ according to the distribution $\widetilde{\mathcal{B}}$

and evaluating $\mathcal{B}(x)$. We say that $\widetilde{\mathcal{B}}$ computes the function f with error at most ϵ if for every input x, $\Pr[\widetilde{\mathcal{B}}(x) = f(x)] \geq 1 - \epsilon$. The length (resp. size) of $\widetilde{\mathcal{B}}$ is the maximum length of any branching program that gets nonzero probability under the distribution.

This notion of probabilistic branching program differs from the standard notion which is obtained by modifying the definition of deterministic branching program to allow "random" nodes which are not labeled by variables, but where the execution randomly selects an output arc. It is well known and easy to see that our notion is at least as powerful as the standard notion and thus is sufficient for the purpose of proving lower bounds.

We note the following well known fact.

Proposition 2.1. *Let $f : D^n \longrightarrow \{0,1\}$ and suppose $\widetilde{\mathcal{B}}$ is a randomized branching program of size at most S and length at most T that computes f with error probability at most ϵ. Then there is a deterministic branching program $\mathcal{B}$ of size at most S and length at most T that approximates f with error at most ϵ.*

Proof. For deterministic branching program $\mathcal{B}$ and input x, let $g(\mathcal{B}, x) = 1$ if $\mathcal{B}(x) \neq f(x)$ and 0 otherwise. Define $q(\mathcal{B}) = |D|^{-n} \sum_{x \in D^n} g(\mathcal{B}, x)$. For each x, the probability that $\widetilde{\mathcal{B}}(x) \neq f(x)$ is equal to the expectation $\mathrm{E}_{\widetilde{\mathcal{B}}}[g(\widetilde{\mathcal{B}}, x)]$ which is at most ϵ, by hypothesis. Averaging over x, we have $\mathrm{E}_{\widetilde{\mathcal{B}}}[q(\mathcal{B})] \leq \epsilon$ which means there is a $\mathcal{B}$ having nonzero probability under $\widetilde{\mathcal{B}}$ such that $q(\mathcal{B}) \leq \epsilon$. □

2.4 Decision Trees and Decision Forests

A *decision tree* is a branching program $\mathcal{B}$ whose underlying graph is a tree rooted at $start_{\mathcal{B}}$. In particular, a decision tree is leveled. 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.

2.5 Converting branching programs to a disjunction of decision forests

The following result is a minor variant of a lemma proved in [BST98, BJS01], which says roughly that the function computed by a branching program that is not too large and not too deep can be expressed as the OR of a not too large collection of decision forests, each of which consists of a small set of shallow trees.

Lemma 2.2. *Let $k, S \in \mathbf{R}$ and $n \in \mathbf{N}$ and D be a finite set. Let $\mathcal{B}$ be an n-variate branching program over domain D having length at most kn and size at most 2^S. Then for any integer $r \in [kn]$, the function f computed by $\mathcal{B}$ can be expressed as:*

$$f = \bigvee_{i=1}^{u} F_i,$$

where $u \leq 2^{Sr}$, each F_i is an inquisitive $(r, \frac{k+2}{r})$-decision forest, and the sets $F_i^{-1}(1)$ are pairwise disjoint sets of inputs.

Proof. As noted in Section 2.3, there is a leveled branching program $\mathcal{B}'$ of length kn with at most 2^S nodes per level that computes the same function as $\mathcal{B}$. Furthermore, let $\mathcal{B}''$ be the length $(k+1)n$ branching program obtained from $\mathcal{B}'$ by adding n layers at the beginning that obliviously query each variable. For distinct nodes v and w of $\mathcal{B}''$, let $f_{v,w}$ denote the function on D^n which is 1 on input σ if, starting from v, the path consistent with σ leads to w. It is easy to see that if v is at level i and w is at level $j > i$, then $f_{v,w}$ can be computed by a decision tree of height $j-i$. For each positive integer i less than r define $l_i = \lceil \frac{ikn}{r} \rceil$. Note that $l_1 < \cdots < l_{r-1} < kn$ divides the interval $[0, kn]$ into r intervals each of size at most $\frac{kn}{r} + 1 \leq (\frac{k+1}{r})n$. An input is accepted by $\mathcal{B}''$ if and only there is a sequence of nodes $v_0, v_1, v_2, \ldots, v_{r-1}, v_r$, where v_0 is the start node, v_r is the accepting node and for $i \in [r-1]$, v_i is at level l_i, such that $f_{v_{i-1},v_i}(\sigma) = 1$ for each $i \in [r]$. Therefore

$$f = \bigvee_{v_1,\ldots v_{r-1}} \bigwedge_{i=0}^{r-1} f_{v_i,v_{i+1}}.$$

There are at most $2^{S(r-1)}$ terms in the $\bigvee$, and each term is a $(r, \frac{k+2}{r})$ decision forest.

Finally, each input follows a unique path, and so is accepted by at most one of the decision forests. Note that since $\mathcal{B}''$ obliviously reads all variables at the beginning, each of the decision forests in the decomposition produced in the above argument is inquisitive. □

3 Overview and comparison to previous results

The main approach taken in [BST98, BJS01, Ajt99a, Ajt99b] for proving time-space tradeoff lower bounds is to show that for any branching program running in time T and space S, where T and S are suitably small, if the fraction of inputs for which the branching program outputs 1 is not too small then there must be some embedded rectangle R having large feet and leg-density consisting entirely of inputs on which the program outputs 1.

There are two main differences between our results and previous results for decision problems. First of all, we obtain substantially larger values for the foot size and leg-density of the obtained rectangles. Secondly, we show that not only is there one large embedded rectangle on which the branching program outputs 1 but there is a collection of such embedded rectangles that together cover most of the inputs on which the branching program outputs 1, and such that no input is covered too many 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 Table 1. Each result has the following form: Given a branching program of depth (time) $T = kn$ and 2^S nodes (space S) of the indicated program type that computes function f that is 1 on at least a $\delta(f)$ fraction of its inputs, then there is a (balanced) embedded rectangle R that is (m, λ)-large (as defined in section 2.2), for suitably large m and λ, that contains very few inputs of $f^{-1}(0)$. The lower bound on foot size m has the form $n/\beta_0(k)$, and the lower bound on leg density has the form $\lambda(m) = \delta(f)/2^{\beta_1(k)m+\beta_2(k)S}$, where $\beta_0, \beta_1, \beta_2$ are nonnegative valued functions. The quantity $\beta_1(k)m + \beta_2(k)S$, which appears in the exponent of 2 in the expression for $\lambda(m)$ provides an upper bound on $\log_2(\delta(f)/\alpha(R))$, which we call the *leg-deficiency* of R. Smaller values of $\beta_0(k), \beta_1(k), \beta_2(k)$ give larger embedded rectangles and better the time-space tradeoff lower bounds.

The Error column indicates the fraction of inputs of the rectangle that belong to $f^{-1}(0)$. This error is 0 except in the case that the branching program has 2-sided error ϵ, in which case it is proportional to ϵ.

Any nonempty rectangle has leg-deficiency at most $m \log |D|$, and to obtain non-trivial time-space trade-

Paper	Foot Size $m(R)$	Leg Deficiency $\log_2(\delta(f)/\alpha(R))$	Program Type	Error $\frac{\lvert R\cap f^{-1}(0)\rvert}{\lvert R\rvert}$	Applicability
[BST98, BJS01]	$2^{-O(k)}n$	$O(k)m + 2^{O(k)}S$	non-determ.	0	$k = O(\log n)$, $\lvert D\rvert = 2^{\Omega(k)}$
[Ajt99a, Pag01]	$k^{-O(k)}n$	$O(k\log k)m + k^{O(k)}S$	determ./ 1-sided err. ϵ	0 0	$k = O(\frac{\log n}{\log\log n})$, $\lvert D\rvert = k^{\Omega(k)}$
Here	$2^{-O(k)}n$	$O(k)m + 2^{O(k)}S$	2-sided err. ϵ	$O(\epsilon)$	$k = O(\log n)$, $\lvert D\rvert = 2^{\Omega(k)}$
[Ajt99a]	$2^{-k^{O(k)}}n$	$2^{-k^{\Omega(k)}}m + 2^{k^{O(k)}}S$	determ.	0	$k = O(\frac{\log\log n}{\log\log\log n})$
Here	$k^{-O(k^2)}n$	$k^{-\Omega(1)}m + k^{O(k^2)}S$	determ.	0	$k = O(\sqrt{\frac{\log n}{\log\log n}})$
Here	$k^{-O(k^2)}n$	$k^{-\Omega(1)}m + k^{O(k^2)}S$	2-sided err. ϵ	$O(k\epsilon)$	$k = O(\sqrt{\frac{\log n}{\log\log n}})$

Table 1: Properties of embedded rectangle R found given a D-way branching program with time $T = kn$ and space S computing a function $f : D^n \to \{0,1\}$ with $\delta(f) = |f^{-1}(1)|/|D^n|$.

offs results, we will need leg-deficiency considerably smaller. Thus, in the expression $\beta_1(k)m + \beta_2(k)S$, we need $\beta_1(k)$ to be sufficiently smaller than $\log|D|$. In particular, the first group of bounds in the table is useful only if $|D|$ is sufficiently large. The second group of bounds has $\beta_1(k) = o(1)$ which enables us to obtain results for the most interesting case, $D = \{0,1\}$.

In general, the best lower bound achievable from each result will be of the form $T = \Omega(n\beta^{-1}(\frac{n\log|D|}{S}))$ where $\beta(k) = \beta_0(k)\cdot\beta_2(k)$. The upper bound on $k = T/n$ listed in the last column is the limit on the best lower bound achievable given a polynomial size branching program.

Section 5 contains the precise statements and proofs of the new stronger results outlined above that if f is a decision function computed by a small and shallow branching program then there is a collection of large rectangles that covers a substantial portion of $f^{-1}(1)$. As in [BST98, BJS01], the main step (which appears in section 4) is to prove corresponding results for the case that f is computed by a small and shallow decision forest. Straightforward application of Lemma 2.2 then gives the desired results about small branching programs. Applications of this result to lower bounds on specific functions are given in section 6.

4 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 $[n]$. (Such an F arises from a branching program of depth $(k-2)n$ using the construction of Lemma 2.2.) Our goal here is to show that one can find a collection of embedded rectangles, such that:

(G1) Each rectangle is contained in $F^{-1}(1)$.

(G2) No single input belongs to "many" rectangles.

(G3) The union of the rectangles covers all but a small number of inputs in $F^{-1}(1)$.

(G4) Each rectangle in the collection has foot size at least n/β_0 where β_0 depends only on k and is as small as possible.

(G5) Each rectangle in the collection is λ-dense where $\lambda : [n] \longrightarrow [0,1]$ is a function that is as large as possible and, in particular, satisfies $\lambda(m) \geq |D|^{-\epsilon m}$ for some constant $\epsilon < 1$.

(G6) Each rectangle is balanced.

All but the first and last of these conditions depend on parameters that will be selected as we proceed. The first three conditions, (G1), (G2), (G3), concern the coverage of the set of rectangles with respect to $F^{-1}(1)$, whereas the last three, (G4), (G5), (G6) refer to parameters of the individual rectangles within the cover. We will first concentrate on obtaining sets of rectangles with the coverage properties that satisfy the parameter conditions (G4), (G5), which together imply that each rectangle is large; we will only derive the balance condition (G6) at the end of the argument. However, in proving conditions (G4) and (G5) we will find it useful to first ensure that the rectangles are all approximately balanced, more precisely 3-balanced; the final balance condition will follow easily afterward.

4.1 Constructing a rectangle partition from two disjoint forests

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. We start by looking at the combinatorial structure induced by a single subforest on the set of inputs. Let $T \in F$, $F_1 \subseteq F$, and $x \in D^n$ We define:

- $\mathtt{read}(x, T)$ is the set of indices read by T on input x.
- $\mathtt{read}(x, F_1) = \bigcup_{T \in F_1} \mathtt{read}(x, T)$.
- $\mathtt{core}(x, F_1) = \mathtt{read}(x, F_1) - \mathtt{read}(x, F - F_1)$, 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 $[n] - \mathtt{core}(x, F_1)$. Since F is inquisitive, this means that $\mathtt{stem}(x, F_1)$ is the projection of x onto $\mathtt{read}(x, F - F_1)$.
- $\mathtt{stems}(F_1)$, the set of F_1 stems, is the set of partial inputs ρ for which there exists $x \in D^n$ with $\mathtt{stem}(x, F_1) = \rho$.

For $\rho \in \mathtt{stems}(F_1)$, it is clear from the definition that any $x \in D^n$ satisfying $\mathtt{stem}(x, F_1) = \rho$ belongs to $D^n(\rho)$. The converse of this also true, though less obvious:

Lemma 4.1. *Let F_j be a subforest of an inquisitive decision forest F and let $\rho \in \mathtt{stems}(F_j)$. For all $x \in D^n(\rho)$, $\mathtt{stem}(x, F_j) = \rho$ and $\mathtt{core}(x, F_j) = \mathtt{unfixed}(\rho)$.*

Proof. Let $x \in D^n(\rho)$. Since $\rho \in \mathtt{stems}(F_j)$ there is an input y with $\rho = \mathtt{stem}(y, F_j)$. Since F is inquisitive, ρ is the projection of y onto $\mathtt{read}(y, F - F_j)$, which means that on input y, the trees of $F - F_j$ read precisely the indices of $\mathtt{fixed}(\rho)$. Since $x \in D^n(\rho)$, each $T \in F - F_j$ behaves the same on x as it does on y. So $\mathtt{read}(x, F - F_j) = \mathtt{fixed}(\rho)$. Thus $\mathtt{core}(x, F_j) = \mathtt{unfixed}(\rho)$, and the restriction of x to $\mathtt{read}(x, F - F_j)$ is also ρ, i.e., $\mathtt{stem}(x, F_j) = \rho$. $\square$

Now we consider the combinatorial structure induced by a pair of subforests F_1 and F_2 which are disjoint subsets of F. Define:

- $\mathtt{stem}(x, F_1, F_2)$ is the partial input on $[n] - \mathtt{core}(x, F_1) - \mathtt{core}(x, F_2)$ obtained from projecting x.

We 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. For $x \in F^{-1}(1)$, let $R(x, F_1, F_2)$ denote the equivalence class containing x.

Lemma 4.2. *Let $F_1, F_2 \subset F$ be disjoint subforests of the inquisitive decision forest F. Let $R \in \mathcal{R}(F_1, F_2)$. Then R is an embedded rectangle with feet $(\mathtt{core}(R, F_1), \mathtt{core}(R, F_2))$ and spine $\mathtt{stem}(R, F_1, F_2)$.*

Proof. Let $A_1 = \mathtt{core}(R, F_1)$ and $A_2 = \mathtt{core}(R, F_2)$ and $\sigma = \mathtt{stem}(R, F_1, F_2)$. By definition, A_1 and A_2 are disjoint. Let $L_1 = \{\tau_1 \in D^{A_1} : \tau_1\sigma \in \mathtt{stems}(F_2)\}$ and $L_2 = \{\tau_2 \in D^{A_2} : \sigma\tau_2 \in \mathtt{stems}(F_1)\}$. Let Q be the embedded rectangle with feet A_1 and A_2, legs L_1 and L_2, and spine σ. It suffices to show that $R = Q$.

First we show $R \subseteq Q$. Let $x \in R$. By definition of R, $\mathtt{core}(x, F_1) = A_1$ and $\mathtt{core}(x, F_2) = A_2$. Write $x \in R$ as $\tau_1\sigma\tau_2$ where $\tau_1 \in D^{A_1}$, and $\tau_2 \in D^{A_2}$. Since $\tau_1\sigma = \mathtt{stem}(x, F_2)$ and $\sigma\tau_2 = \mathtt{stem}(x, F_2)$, we have $\tau_1 \in L_1$ and $\tau_2 \in L_2$ and therefore $x \in Q$.

Next we show $Q \subseteq R$. Let $x = \tau_1\sigma\tau_2 \in Q$ such that $\tau_1 \in L_1$ and $\tau_2 \in L_2$. Now since $\tau_1\sigma \in \mathtt{stems}(F_2)$ and $\sigma\tau_2 \in \mathtt{stems}(F_1)$, by Lemma 4.1 we have $\mathtt{core}(x, F_2) = \mathtt{unfixed}(\tau_1\sigma) = A_2$ and $\mathtt{core}(x, F_1) = \mathtt{unfixed}(\sigma\tau_2) = A_1$. Therefore, $x \in R$. $\square$

Thus, each pair of disjoint forests F_1, F_2 induces a partition $\mathcal{R}(F_1, F_2)$ of $F^{-1}(1)$ into embedded rectangles (which thus satisfies the covering conditions (G1), (G2) and (G3)). However, we also want the rectangles in our collection to be suitably large (and balanced). There is no guarantee, for an arbitrary pair of forests F_1, F_2, if we eliminate rectangles of its associated partition that are not suitably large, that the remainder will cover a sufficiently large fraction of $F^{-1}(1)$ (violating (G3)). To help with this we use the probabilistic method to choose a pair of forests F_1, F_2 for which this idea suffices in certain cases. Depending on the notions of "suitably large" that we require, even applying this idea with a single pair of forests may not suffice. For these stronger results we need to apply the probabilistic method to obtain several different choices of pairs of forests whose associated partitions have the property that the suitably large rectangles in the union of the partitions covers most of the inputs in $F^{-1}(1)$. If the number of different choices is not too large then we will be able to satisfy (G3) without violating (G2).

4.2 Analysis of core size for randomly chosen forests

We begin by defining a parameterized family of probability distributions over pairs (F_1, F_2) of forests and analyzing properties of $\mathcal{R}(F_1, F_2)$ when (F_1, F_2) is chosen according to a distribution in this family. In [BST98, BJS01], (F_1, F_2) was chosen to be a random partition of F into two parts. Ajtai [Ajt99a] 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 [BST98, BJS01] corresponds to the case $q = 1/2$.

For $x \in D^n$, let $\mu(x,q) = E[|\mathtt{core}(x, F_1)|] = E[|\mathtt{core}(x, F_2)|]$ 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 $\mathtt{core}(x, F_1)$ and $\mathtt{core}(x, F_2)$ are close to $\mu(x,q)$. This lemma generalizes a lemma proved in [BST98, BJS01] for the $q = 1/2$ case. Ajtai proved tighter concentration bounds for his distributions using a more detailed analysis, but since the tighter bounds are not significant in the final results, we content ourselves with a simple second moment argument.

Lemma 4.3. *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 $j \in \{1,2\}$, $\Pr\left[||\mathtt{core}(x, F_j)| - \mu(x,q)| \geq \frac{1}{2}\mu(x,q)\right] \leq \frac{4k^2}{rq^k}$

Proof. By symmetry, it is enough to consider the case $j = 1$.

For $i \in [n]$. $\Pr[i \in \mathtt{core}(x, F_1)] = q^{t(i)}$, where $t(i)$ is the number of trees that access variable i on input x. Thus $\mathrm{E}[|\mathtt{core}(x, F_1)|] = \sum_{i \in [n]} q^{t(i)}$. Since F makes at most kn reads on input x, $\frac{1}{n}\sum_{i\in[n]} t(i) \leq k$. By the arithmetic-geometric mean inequality, $\mathrm{E}[|\mathtt{core}(x, F_1)|] = \sum_i q^{t(i)} \geq n q^{\frac{1}{n}\sum_i t(i)} \geq q^k n$.

Next we upper bound $\mathrm{Var}[|\mathtt{core}(x, F_1)|]$. Let $M(i)$ be the event that $i \in \mathtt{core}(x, F_1)$. For $1 \leq i, i' \leq n$, we say $i \sim i'$ if there is $T \in F$ that accesses both x_i and $x_{i'}$ on input x. Now

$$\mathrm{Var}[|\mathtt{core}(x, F_1)|] = \sum_{i,i'} (\Pr[M(i) \wedge M(i')] - \Pr[M(i)] \cdot \Pr[M(i')]).$$

If $\neg(i \sim i')$ then the events $M(i)$ and $M(i')$ are independent and the corresponding term in the sum is 0. If $i \sim i'$ then we upper bound $\Pr[M(i) \wedge M(i')] - \Pr[M(i)] \cdot \Pr[M(i')]$ crudely by $\Pr[M(i)] = q^{t(i)}$. Since on input x, each tree reads at most $\frac{k}{r}n$ variables, for each i the number of i' such that $i \sim i'$ is at most $t(i)\frac{k}{r}n$. Thus,

$$\mathrm{Var}[|\mathtt{core}(x, F_1)|] \leq \frac{k}{r} n \sum_{i=1}^{n} t(i) q^{t(i)} \leq \frac{k}{r} \sum_{i=1}^{n} t(i) \sum_{j=1}^{n} q^{t(j)} \leq \frac{k^2 n}{r} \mu(x,q).$$

(The second inequality uses a form of Chebyshev's inequality (e.g. [HLP52, Theorem 43, page 43]) which says that when a_i and b_i are positive and anti-correlated, $\sum_{i=1}^n a_i b_i \leq \sum_{i=1}^n a_i \sum_{j=1}^n b_j / n$.)

We now use the more usual form of Chebyshev's inequality: for any random variable Z with finite expectation and variance, $\Pr[|Z - \mathrm{E}[Z]| \geq \zeta] \leq \mathrm{Var}[Z]/\zeta^2$.

$$\begin{aligned} \Pr\left[|\mathtt{core}(x, F_1)| - \mu(x,q)||| \geq \frac{1}{2}\mu(x,q)\right] &\leq 4\mathrm{Var}[|\mathtt{core}(x, F_1)|]/\mu(x,q)^2 \\ &\leq \frac{4k^2 n}{r\mu(x,q)} \leq \frac{4k^2}{rq^k}. \end{aligned}$$

□

4.3 Choosing rectangles with high leg-density: overview

The lemma in the previous subsection implies that for (F_1, F_2) chosen according to $\mathcal{F}_q$, the subset of $\mathcal{R}(F_1, F_2)$ consisting of those rectangles that both have foot size at least $q^k n/2$ and are 3-balanced covers all but a few inputs of $F^{-1}(1)$. Provided that q is not too small, this would produce a set of rectangles that satisfy some version of the covering conditions (G1),(G2),(G3) as well as the lower bound on foot-size (G4) and approximate balance. If we did not care about the leg-density bound (G5), then we would choose $q = 1/2$, and we would essentially be done. However, we want the chosen rectangles to have sufficiently high leg-density to satisfy (G5). To obtain the time-space tradeoffs for the various functions considered in [BST98, BJS01] and [Ajt99a, Ajt99b], we will want the leg-density bound $\lambda(m) = |D|^{-\epsilon m}$ for some $\epsilon < 1$. (Notice that for $\lambda(m) \leq |D|^{-m}$, any nonempty rectangle is trivially λ-dense.)

We would like that for F_1, F_2 chosen according to the $\mathcal{F}_q$, almost all inputs in $F^{-1}(1)$ are in rectangles that are λ-dense, for some appropriate $\lambda(m)$. 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), it is easy to show that this is true for *every* choice of (F_1, F_2) even if we take $\lambda(m)$ to be a constant function. In this case, for any given pair (F_1, F_2), $\mathtt{core}(x, F_1)$ and $\mathtt{core}(x, F_2)$ are the same for all inputs x, so all of the rectangles in $\mathcal{R}(F_1, F_2)$ have the same pair of feet (A_1, A_2). Thus these rectangles are determined only by their spines σ on $[n] - A_1 - A_2$. For any $\eta > 0$, and for $j \in \{1, 2\}$, 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 $\mathcal{R}(F_1, F_2)$. Therefore, for the constant function $\lambda(m) = \eta$, the number of inputs that are not in λ-dense rectangles is at most $2\eta |D^n|$.

The idea of this argument is that the definition of λ-sparse imposes an upper bound on the size of each λ-sparse rectangle and we multiply this by (an upper bound on) $|\mathcal{R}(F_1, F_2)|$. In the general (nonoblivious) case, the rectangles in $\mathcal{R}(F_1, F_2)$ do not all have the same feet, which creates two problems: (1) the size upper bound on a λ-sparse rectangle also depends on the size of the feet, and so is different for different rectangles, and, more significantly, (2) it is harder to get good upper bounds on $|\mathcal{R}(F_1, F_2)|$.

The rest of this section is devoted to proving two lemmas, Lemma 4.4 and Lemma 4.13. The first lemma uses a simple argument that achieves a leg-density lower bound $\lambda(m) = 2^{-O(km)}$, which is enough to prove time-space tradeoffs for some functions in the case that the domain D is large, in particular larger than 2^{ck} for some constant c. The second lemma is much harder and achieves a leg-density lower bound $\lambda(m) = 2^{-\epsilon m}$ for $\epsilon < 1$, which is needed for the time-space tradeoffs for boolean functions and for the element distinctness problem.

4.4 Weak lower bounds on leg-density

Lemma 4.4. *Let F be an n-variable inquisitive $(r, k/r)$ decision forest where $n \geq r \geq k \geq 2$ are integers. Let $1 > \gamma', \delta' > 0$ and suppose that $r \geq 2^{k+4}k^2/\gamma'$. Then there is a family $\mathcal{R}$ of disjoint rectangles such that each rectangle $R \in \mathcal{R}$ is a subset of $F^{-1}(1)$ and satisfies $m(R) = m_1(R) = m_2(R) \geq \lceil n/2^{k+1} \rceil$ and $\alpha(R) \geq 2^{-12(k+1)m(R)}\delta'$, and such that the set $|\cup_{R \in \mathcal{R}} R|$ has size at least $(1 - \gamma')|F^{-1}(1)| - \delta'|D^n|$.*

Proof. Let $r \geq 2^{k+4}k^2/\gamma'$ and choose (F_1, F_2) according to $\mathcal{F}_{1/2}$. By Lemma 4.3, for each $x \in F^{-1}(1)$, there is a $w_x = \frac{1}{2}\mu(x, 1/2) \geq n/2^{k+1}$ such that $\Pr[|\mathtt{core}(x, F_j)| \notin [w_x, 3w_x]] \leq \gamma'/2$ for $j = 1, 2$. Therefore there is a pair (F_1, F_2) such that $|\mathtt{core}(x, F_1)|, |\mathtt{core}(x, F_2)| \in [w_x, 3w_x]$ for all inputs x in a subset J of $F^{-1}(1)$ of size at least $(1 - \gamma')|F^{-1}(1)|$. Let $\mathcal{Q}$ be the set of all embedded rectangles $R \in \mathcal{R}(F_1, F_2)$ that contain at least one element of J. By construction every embedded rectangle R in $\mathcal{Q}$ has $n/2^{k+1} \leq m(R) = \min(m_1(R), m_2(R))$ and $\max(m_1(R), m_2(R)) \leq 3m(R)$.

We first partition each of the embedded rectangles in $\mathcal{Q}$ to produce a set $\mathcal{Q}'$ of balanced rectangles as follows: For each embedded rectangle (R, A_1, A_2) in $\mathcal{Q}$, if $j \in \{1,2\}$ is an index such that $m(R) = m_j(R) = |A_j|$, we define $B_j = A_j$, define $B_{3-j} \subseteq [n]$ to be the set consisting of the smallest $m(R)$ elements of A_{3-j} and replace (R, A_1, A_2) by its partition into embedded rectangles with feet B_1 and B_2, $\mathrm{Refine}(R, B_1, B_2)$ (as defined in section 2.2). Clearly each embedded rectangle $R' \in \mathrm{Refine}(R, B_1, B_2)$ has $m(R') = m_1(R') = m_2(R') = m(R) \geq n/2^{k+1}$.

We now define the subset $\mathcal{R}$ of $\mathcal{Q}'$ to be those embedded rectangles R' such that $|R'| \geq 2^{-12(k+1)m(R')}\delta'|D|^{2m(R')}$. We claim that the union of all rectangles in $\mathcal{Q}' - \mathcal{R}$ contains at most $\delta'|D|^n$ inputs.

Each rectangle in $\mathcal{Q}$ is defined by its feet corresponding to the common core sets $A_1, A_2 \subset [n]$ and its spine, the partial assignment $\sigma \in D^{[n]-A_1-A_2}$ corresponding to the common stem. Furthermore, each refined rectangle R' in $\mathcal{Q}'$ is defined by specifying the rectangle R in $\mathcal{Q}$ from which it was derived, together with the partial assignment to the $\max(m_1(R), m_2(R)) - m(R)$ variables of largest index in the larger of A_1 or A_2.

We count the rectangles in $\mathcal{Q}'$ separately based on the possible values of $m_1(R)$ and $m_2(R)$ of the rectangle R from which they are derived. For each fixed pair (m_1, m_2) of integers there are at most $\binom{n}{m_1}\binom{n}{m_2}|D|^{n-m_1-m_2}$ rectangles $R \in \mathcal{Q}$ with $m_1(R) = m_1$ and $m_2(R) = m_2$ and thus at most

$$\binom{n}{m_1}\binom{n}{m_2}|D|^{n-m_1-m_2}|D|^{\max(m_1,m_2)-\min(m_1,m_2)} = \binom{n}{m_1}\binom{n}{m_2}|D|^{n-2\min(m_1,m_2)}$$

rectangles $R' \in \mathcal{Q}'$ derived from such rectangles R. By construction we only need to consider integer pairs (m_1, m_2) with $n \geq m_1, m_2 \geq n/2^{k+1}$ such that $\max(m_1, m_2) \leq 3\min(m_1, m_2)$. Now, using the fact (easily checkable given the standard bound $\binom{n}{m} \leq 2^{H_2(m/n)n}$ where $H_2(p) = p\log_2(1/p) + (1-p)\log_2(1/(1-p))$) that for $\ell \geq 1$ if $m \geq n/2^\ell$ then $\binom{n}{m} \leq 2^{2\ell m}$, for these values of m_1 and m_2,

$$\begin{aligned}\binom{n}{m_1}\binom{n}{m_2}|D|^{n-2\min(m_1,m_2)} &\leq 2^{2(k+1)(m_1+m_2)}|D|^{n-2\min(m_1,m_2)}\\ &\leq 2^{8(k+1)\min(m_1,m_2)}|D|^{n-2\min(m_1,m_2)}.\end{aligned}$$

Therefore the total number of inputs in rectangles R' in $\mathcal{Q}'$ with $|R'| < 2^{-12(k+1)m(R')}\delta'|D|^{2m(R')}$ such that $m_1(R) = m_1$ and $m_2(R) = m_2$ is at most $2^{-4(k+1)\min(m_1,m_2)}\delta'|D|^n$. Summing over all pairs (m_1, m_2) we need to consider shows that the number of inputs in J not covered by $\mathcal{R}$ is at most $n^2 2^{-4(k+1)n/2^{k+1}}\delta'|D|^n \leq \delta'|D|^n$ since $m_1, m_2 \geq n/2^{k+1}$ and $n/\log_2 n \geq 2^k/k$ for $n \geq r \geq 2^{k+4}k^2$.

Since any rectangle R' with both feet of size $m(R')$ has precisely $\alpha_1(R')\alpha_2(R')|D|^{2m(R')}$ elements and since $\alpha_j(R') \leq 1$ for $j = 1, 2$, for every rectangle R' in $\mathcal{R}$, $\alpha(R') = \min(\alpha_1(R'), \alpha_2(R')) \geq 2^{-12(k+1)m(R')}\delta'$ as required. $\square$

The proof of Lemma 4.4 is very similar to that of the result of [BST98, BJS01] cited in Table 1. The main difference here is that in [BST98, BJS01] the argument only produces a single rectangle that is suitably large and dense, while the above lemma gives a collection of disjoint rectangles that covers all but a small number of points in $F^{-1}(1)$; this extension will permit lower bounds for randomized branching programs with 2-sided error. We get a small savings of a 2^{-r} factor in the bound and the 12 in the exponent is slightly worse because of our extension to the randomized case, but these will not significantly change the lower bound when we extend it to the entire branching program.

This lemma is the only part of this section needed to prove the time-space tradeoffs for branching programs for the Hamming closeness function and for quadratic forms over large fields. The reader who wishes to get an idea how the "large rectangle" results are applied can go to section 5.1 and then the relevant parts of section 6.

4.5 A sufficient condition for high leg-density

We turn to the harder task of improving the density lower bounds on the rectangles in our cover to be much larger than 2^{-m}. Conceptually, our approach closely follows that used to prove the main lemma of [Ajt99a]. The overall strategy involves classifying inputs based on the pattern of accesses to their input variables made by the various trees in the decision forest.

We will begin by developing a general condition on a pair of forests F_1, F_2 and an arbitrary subset $J \subseteq D^n$ of inputs that will allow us to obtain good leg-density lower bounds on the rectangles in $\mathcal{R}(F_1, F_2)$ that cover most of J. We will then show that this condition holds if the restrictions of the access patterns of the inputs in J to the trees of F_1 and F_2 satisfy a certain property. Finally, we will show that there is a small set Γ of probabilities q satisfying the following. If the inputs are partitioned into classes based on their overall access patterns, for any such class of inputs J there is some $q \in \Gamma$ such that, for F_1, F_2 chosen from $\mathcal{F}_q$, the restrictions of the access patterns of the inputs in J to F_1 and F_2 satisfy the desired property.

We now work out the condition that implies large leg-density. Fix a pair of forests F_1, F_2. We begin with an alternate characterization of leg-density in terms of F_j-stems.

Lemma 4.5. *Let $\rho \in \mathtt{stems}(F_j)$ and let $R \in \mathcal{R}(F_1, F_2)$ satisfy $R \cap D^n(\rho) \neq \emptyset$. Then $\alpha_j(R) = |R \cap D^n(\rho)|/|D^n(\rho)|$.*

Proof. Let ρ and R be as hypothesized. Let A_1, A_2 be the feet of R, σ be the spine and Y_1, Y_2 be the legs. Suppose $x \in R \cap D^n(\rho)$. Let τ be the restriction of x to A_{3-j}. Then $\tau \in Y_{3-j}$ and $\sigma\tau = \mathtt{stem}(x, F_j)$. By Lemma 4.1 $\rho = \sigma\tau$. Since $R = \{\pi_1\pi_2\sigma : \pi_1 \in Y_1, \pi_2 \in Y_2\}$, we have that $R \cap D^n(\rho) = \{\pi_j\tau\sigma : \pi_j \in Y_j\}$ and thus $|R \cap D^n(\rho)| = |Y_j| = \alpha_j(R)|D^{A_j}| = \alpha_j(R)|D^n(\rho)|$. ☐

Now fix a subset $J \subseteq D^n$ of inputs. Very roughly, if one could show that for any $x \in J$ there are very few rectangles in $\mathcal{R}(F_1, F_2)$ containing inputs in J that extend $\mathtt{stem}(x, F_j)$ then by some kind of averaging one would expect that most points in J will lie in rectangles that have relatively large j-density. In order to make this rough argument precise we need the following property of $\mathtt{stems}(F_j)$ which follows immediately from Lemma 4.1.

Lemma 4.6. $\{D^n(\rho) : \rho \in \mathtt{stems}(F_j)\}$ *is a partition of* D^n.

Let $\lambda : [n] \to [0, 1]$ be an arbitrary function and let $\mathcal{Q}^j$ be the set of rectangles R in $\mathcal{R}(F_1, F_2)$ with $\alpha_j(R) < \lambda(m_j(R))$. The number of inputs of J that belong to elements of $\mathcal{Q}^j$ is $\sum_{R \in \mathcal{Q}^j} |R \cap J|$. To upper bound this sum, we classify points according to their F_j-stem and separately upper bound the number of points in each class that are contained in such sparse rectangles.

$$\begin{aligned}
\sum_{R\in\mathcal{Q}^j}|R\cap J| &= \sum_{\rho\in\mathtt{stems}(F_j)}\sum_{R\in\mathcal{Q}^j}|R\cap J\cap D^n(\rho)| \\
&\le \sum_{\rho\in\mathtt{stems}(F_j)}\sum_{\substack{R\in\mathcal{Q}^j\\ R\cap J\cap D^n(\rho)\neq\emptyset}}|R\cap D^n(\rho)| \\
&= \sum_{\rho\in\mathtt{stems}(F_j)}\sum_{\substack{R\in\mathcal{Q}^j\\ R\cap J\cap D^n(\rho)\neq\emptyset}}\alpha_j(R)\cdot|D^n(\rho)| \\
&< \sum_{\rho\in\mathtt{stems}(F_j)}|\{R\in\mathcal{R}(F_1,F_2):R\cap J\cap D^n(\rho)\neq\emptyset\}|\cdot\lambda(|\mathtt{unfixed}(\rho)|)\cdot|D^n(\rho)|.
\end{aligned}$$

Define $\mathtt{numrects}(\rho,J)=|\{R\in\mathcal{R}(F_1,F_2):R\cap J\cap D^n(\rho)\neq\emptyset\}|$. We rewrite the last line and continue:

$$\begin{aligned}
&\sum_{\rho\in\mathtt{stems}(F_j)}|D^n(\rho)|\cdot\lambda(|\mathtt{unfixed}(\rho)|)\cdot\mathtt{numrects}(\rho,J) \\
&\quad\le \max_{\rho\in\mathtt{stems}(F_j)}\lambda(|\mathtt{unfixed}(\rho)|)\cdot\mathtt{numrects}(\rho,J)\sum_{\rho\in\mathtt{stems}(F_j)}|D^n(\rho)| \\
&\quad= |D^n|\cdot\max_{\rho\in\mathtt{stems}(F_j)}\lambda(|\mathtt{unfixed}(\rho)|)\cdot\mathtt{numrects}(\rho,J),
\end{aligned}$$

where the last equality follows from Lemma 4.6. Let $P_{m,j}=\{\rho\in\mathtt{stems}(F_j):|\mathtt{unfixed}(\rho)|=m\}$. Since

$$\max_{\rho\in\mathtt{stems}(F_j)}\lambda(|\mathtt{unfixed}(\rho)|)\cdot\mathtt{numrects}(\rho,J)=\max_{m,\ P_{m,j}\neq\emptyset}(\lambda(m)\max_{\rho\in P_{m,j}}\mathtt{numrects}(\rho,J)),$$

we thus arrive at the following:

Lemma 4.7. *Let F be an n-variable inquisitive decision forest on domain D, let F_1,F_2 be subforests of F and $J\subseteq F^{-1}(1)$. Let $j\in\{1,2\}$, $\eta\in[0,1]$, and for each $m\in[n]$ let $P_{m,j}=\{\rho\in\mathtt{stems}(F_j):|\mathtt{unfixed}(\rho)|=m\}$. If $\lambda:[n]\longrightarrow[0,1]$ satisfies*

$$\lambda(m)\le\frac{\eta}{\max_{\rho\in P_{m,j}}\mathtt{numrects}(\rho,J)}$$

for each m such that $P_{m,j}\neq\emptyset$, then the rectangles R in $\mathcal{R}(F_1,F_2)$ with $\alpha_j(R)<\lambda(m_j(R))$ together cover at most $\eta|D^n|$ points of J.

4.6 Upper bounding $\mathtt{numrects}(\rho,J)$

To use this lemma, we need a good upper bound on $\max_{\rho\in P_{m,j}}\mathtt{numrects}(\rho,J)$. Of course, this quantity depends on F_1,F_2 and J. To this end, we prove an alternative characterization of $\mathtt{numrects}(\rho,J)$:

Proposition 4.8. *Fix the forest pair F_1,F_2. Let J be a subset of $F^{-1}(1)$. For $j\in\{1,2\}$, and $\rho\in\mathtt{stems}(F_j)$, $\mathtt{numrects}(\rho,J)$ is equal to the number of subsets C of $[n]$ for which there is an $x\in J$ with $\mathtt{stem}(x,F_j)=\rho$ and $\mathtt{core}(x,F_{3-j})=C$.*

Proof. For $x \in D^n(\rho)$, we have $\mathtt{core}(x, F_j) = \mathtt{unfixed}(\rho)$ and $\mathtt{stem}(x, F_1, F_2)$ is simply the projection of ρ onto $\mathtt{fixed}(\rho) - \mathtt{core}(x, F_{3-j})$. From this we conclude that for $x, y \in D^n(\rho) \cap J$, $R(x, F_1, F_2) = R(y, F_1, F_2)$ if and only if $\mathtt{core}(x, F_{3-j}) = \mathtt{core}(y, F_{3-j})$. The conclusion of the proposition is immediate. $\square$

Thus $\mathtt{numrects}(\rho, J)$ is the size of a particular collection of subsets of $[n]$, which we will upper bound using:

Proposition 4.9. *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 $S(n, d) = \sum_{j \leq d} \binom{n}{j}$.*

Thus an upper bound on $\mathtt{numrects}(\rho, J)$ will follow from an upper bound for $j = 1, 2$ on $|\mathtt{core}(x, F_{3-j}) \Delta \mathtt{core}(y, F_{3-j})|$ for all $x, y \in J$ having the same F_j-stem. We will carefully partition almost all of $F^{-1}(1)$ into sets J and choose subforests F_1, F_2 depending on certain properties of J so that for $j = 1, 2$ all $x, y \in J$ with the same F_j-stem will be such that $\mathtt{core}(x, F_{3-j}) \Delta \mathtt{core}(y, F_{3-j})$ is much smaller than $\mathtt{core}(x, F_j) = \mathtt{core}(y, F_j)$.

In order to do this, for $j = 1, 2$ we will associate each input $x \in J$ with a subset of variables (depending on j) so that for any two inputs x, y with the same F_j-stem, $\mathtt{core}(x, F_{3-j}) \Delta \mathtt{core}(y, F_{3-j})$ is contained in the union of the subset associated with x and the subset associated with y. Our goal will be achieved by showing that for $j = 1, 2$ and every $x \in J$ the subset of variables associated with x is much smaller than $\mathtt{core}(x, F_j)$.

The subset associated with x will be determined by classifying variables according to which trees read them on input x. In particular, it will depend on F_1 and F_2 and also on an auxiliary parameter ℓ which we will be free to choose later.

With (F_1, F_2) fixed, we define for $j \in \{1, 2\}$ and positive integer $\ell \leq r$:

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

We now show that associating each $x \in D^n$ to the subset $B_{3-j}(x, \ell) \cup B'_{3-j}(x, \ell)$, we get the desired property.

Lemma 4.10. *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 $\mathtt{stem}(x, F_j) = \mathtt{stem}(y, F_j)$ we have*

$$\mathtt{core}(x, F_{3-j}) \Delta \mathtt{core}(y, F_{3-j}) \subseteq B_{3-j}(x, \ell) \cup B'_{3-j}(x, \ell) \cup B_{3-j}(y, \ell) \cup B'_{3-j}(y, \ell).$$

Proof. By symmetry in j, x, y, it suffices to consider the case $j = 2$ and $i \in \mathtt{core}(x, F_1) - \mathtt{core}(y, F_1)$ and show $i \in B_1(x, \ell) \cup B'_1(y, \ell)$.

If $i \notin \mathtt{vset}(x, \ell)$, then $i \in B_1(x, \ell)$. Suppose $i \in \mathtt{vset}(x, \ell)$. 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 $i \in B'_j(y, \ell)$. Therefore $i \in B_1(x, \ell) \cup B'_1(y, \ell)$. $\square$

The free parameter ℓ in the above lemma gives us some freedom in choosing the sets to associate to each input. We want to choose (F_1, F_2) and ℓ so that for almost all inputs x, $B_{3-j}(x,\ell) \cup B'_{3-j}(x,\ell)$ is substantially smaller than $\mathtt{core}(x, F_j)$.

The key observation is that no variable whose index is in $B_{3-j}(x,\ell) \cup B'_{3-j}(x,\ell)$ is read in exactly ℓ trees of F. We will group inputs in $F^{-1}(1)$ into classes $J_{q,\ell}$ for a certain small set of values of $q \in (0, 1/2]$ and $\ell \in [r]$ such that for (F_1, F_2) chosen according to $\mathcal{F}_q$ for almost all $x \in J_{q,\ell}$, the overwhelming majority of the variables in $\mathtt{core}(x, F_1)$ and $\mathtt{core}(x, F_2)$ are read in exactly ℓ trees of F. Therefore for almost all $x \in J_{q,\ell}$, the sizes of $B_1(x,\ell)$ and $B_2(x,\ell)$ will be substantially smaller than the sizes of the cores, $\mathtt{core}(x, F_1)$ and $\mathtt{core}(x, F_2)$; a similar argument will allow us to obtain comparable upper bounds on the sizes of $B'_1(x,\ell)$ and $B'_2(x,\ell)$.

We now show how to group the inputs into the sets $J_{q,\ell}$. Our bounds substantially improve those implicit in [Ajt99a, Ajt99b] 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 [Ajt99a] only uses the randomness of one of the forests in the pair (F_1, F_2) while holding the other fixed. We restructure the analysis so that we can use the randomness of both forests.

Lemma 4.11. *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 $\nu_h = |\mathtt{vset}(x,h)|$ for $h = 1, \ldots, r$. 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 $h \geq 1$. Clearly $h(a)$ is a positive integer and we claim:

(1) $h(1) \leq k$.

(2) $h(a)$ is non-increasing with respect to a.

For the first claim, by Lemma 4.3, $\sum_h \nu_h q_1^h = \mu(x, q_1) \geq q_1^k n$. Since $\sum_{h>k} \nu_h q_1^h < n q_1^{k+1} \leq \frac{n}{k+1} q_1^k$, we have $\sum_{h \leq k} \nu_h q_1^h > \frac{kn}{k+1} q_1^k$, and so for some $\ell \in \{1, \ldots, k\}$, $\nu_\ell q_1^\ell > \frac{n}{k+1} q_1^k \geq \sum_{h>k} \nu_h q_1^h$, which proves the first claim. For the second claim we have for all $h > h(a)$, $\nu_{h(a)} q_a^{h(a)} \geq \nu_h q_a^h$ which implies $\nu_{h(a)} q_{a+1}^{h(a)} > \nu_h q_{a+1}^h$, so $h(a+1) \leq h(a)$.

By the pigeonhole principle, there exists a $b \in \{2, \ldots, 2k\}$ such that $h(b-1) = h(b) = h(b+1)$. Set $\ell = \ell(x)$ to be $h(b)$ and let $b(x) = b$. For $h \geq \ell$, $\nu_h q_1^{(b-1)h} \leq \nu_\ell q_1^{(b-1)\ell}$ implies $\nu_h q_b^h \leq \nu_\ell q_b^\ell q_1^{h-\ell}$. Similarly, for $h \leq \ell$, $\nu_h q_1^{(b+1)h} \leq \nu_\ell q_1^{(b+1)\ell}$ implies $\nu_h q_b^h \leq \nu_\ell q_b^\ell q_1^{\ell-h}$. Thus for $h \neq \ell$, $\nu_h q_b^h \leq \nu_\ell q_b^\ell \cdot q_1^{|h-\ell|}$. Then for

(F_1, F_2) chosen according to the distribution $\mathcal{F}_{q_b}$ we have:

$$\begin{aligned} \mathrm{E}[|B_j(x,\ell)|] &= \sum_{h\neq\ell} \nu_h q_b^h \leq \sum_{h\neq\ell} \nu_\ell q_b^\ell \cdot q_1^{|\ell-h|} \\ &\leq 2\nu_\ell q_b^\ell \sum_{p=1}^{\infty} q_1^p \leq 4\nu_\ell q_b^\ell \cdot q_1 \\ &\leq 4q_1 \cdot \mu(x, q_b), \end{aligned}$$

proving the first part of the lemma.

Note $B'_j(x,\ell) \subseteq \cup_{h\geq\ell+1}\texttt{vset}(x,h)$. For $h \geq \ell+1$ and $i \in \texttt{vset}(x,h)$, $i \in B'_j(x,\ell)$ 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 $i \in \texttt{vset}(x,h)$, we have:

$$\begin{aligned} \mathrm{E}[|B'_j(x,\ell)|] &= \sum_{h=\ell+1}^{r} \nu_h q_b^h \binom{h}{\ell} \\ &\leq \sum_{h=\ell+1}^{r} \nu_h q_b^h (k+1)^{h-\ell} \leq \sum_{h=\ell+1}^{r} \nu_\ell q_b^\ell (k+1)^{h-\ell} q_1^{h-\ell} \\ &\leq \nu_\ell q_b^\ell (k+1) q_1 \sum_{p=0}^{\infty}((k+1)q_1)^p \leq 2kq_1\mu(x,q_b), \end{aligned}$$

where the last inequality uses $k \geq 3$ (so that $(k+1) \leq \frac{4k}{3}$), $q_1 \leq \frac{1}{4k}$ and $\nu_\ell q_b^\ell \leq \mu(x,q_b)$. ☐

4.7 Putting things together

For $b \in \{2,\ldots,2k\}$ and $\ell \in \{1,\ldots,k\}$, 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 apply the probabilistic method for each b separately to show that if $I \subseteq C^b$ for some $b \in \{2,\ldots,2k\}$, 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 4.12. *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 $b \in \{2,\ldots,2k\}$ 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:*

$$m_j(R) \in [\frac{\mu(x,q_b)}{2}, \frac{3\mu(x,q_b)}{2}]$$
$$\alpha_j(R) \geq \frac{\delta}{kS(n, \frac{10kq_1}{\gamma}m_j(R))}.$$

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) $\frac{1}{2}\mu(z,q_b) \leq |\texttt{core}(z,F_j)| \leq \frac{3}{2}\mu(z,q_b)$,

(ii) $|B_{3-j}(z,\ell)| \le 8q_1|\texttt{core}(z,F_j)|/\gamma$,

(iii) $|B'_{3-j}(z,\ell)| \le 4kq_1|\texttt{core}(z,F_j)|/\gamma$.

Conditions (ii) and (iii) follow from (i) and the conditions (ii') $|B_{3-j}(z,\ell)| \le 4q_1\mu(z,q_b)/\gamma$ and (iii') $|B'_{3-j}(z,\ell)| \le 2kq_1\mu(z,q_b)/\gamma$. For each j, Lemma 4.3 says that (i) fails with probability at most $4k^2/(rq_b^k)$, which is at most γ by hypothesis, and Lemma 4.11 with Markov's inequality implies that (ii') and (iii') each fail with probability 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 $|I''| \ge (1-6\gamma)|I|$, 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 will apply Lemma 4.7 with $J = I''_\ell$ separately for each ℓ and $j=1,2$. Consider the F_j-stem ρ of some input in I''_ℓ. Lemma 4.10 with (ii) and (iii) above imply that for $x,y \in D^n(\rho)\cap I''_\ell$, $|\texttt{core}(y,F_{3-j})\Delta\texttt{core}(x,F_{3-j})| \le \frac{(8k+16)q_1}{\gamma}|\texttt{unfixed}(\rho)|$ since $\texttt{core}(x,F_j) = \texttt{core}(y,F_j) = \texttt{unfixed}(\rho)$. Since $k \ge 8$ this is at most $\frac{10kq_1}{\gamma}|\texttt{unfixed}(\rho)|$. By Propositions 4.8 and 4.9, $\texttt{numrects}(\rho, I''_\ell) \le S(n, \frac{10kq_1}{\gamma}|\texttt{unfixed}(\rho)|)$.

Now apply Lemma 4.7 for $j=1,2$ with $\eta = \delta/k$ and $\lambda(m) = \eta/S(n,\frac{10kq_1}{\gamma}m)$. This gives $I'_\ell \subseteq I''_\ell$ of size at least $|I''_\ell| - 2\delta|D|^n/k$, such that for every $x \in I'_\ell$, for $R = R(x,F_1,F_2)$ and for $j=1,2$, $\alpha_j(R) \ge \delta/(kS(n,\frac{10kq_1}{\gamma}m_j(R)))$ which gives the claimed bound on $\alpha_j(R)$ as a function of $m_j(R)$. Let $I' = \cup_{\ell=1}^k I'_\ell$. Then $|I'| \ge |I''| - 2\delta|D|^n \ge |I|(1-6\gamma) - 2\delta|D|^n$. □

We now combine the results for each $b \in \{2,\dots,2k\}$ from Lemma 4.12 and convert (most of) the resulting rectangles into balanced rectangles to arrive at the main result of this section, which says that we can find a collection of large rectangles, each contained in $F^{-1}(1)$, that covers all but a small number of inputs in $F^{-1}(1)$.

Lemma 4.13. *Let F be an n-variable inquisitive $(r,k/r)$ decision forest where $n \ge r \ge k \ge 8$ are integers. Let $q_1 \le 1/(4k)$. Let $\gamma',\delta' > 0$, and suppose $r \ge \frac{48k^2}{\gamma' q_1^{2k^2}}$. 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 of $F^{-1}(1)$ of size at least $|F^{-1}(1)|(1-\gamma') - |D^n|\delta'$, and such that $\mathcal{R}$ can be partitioned into subcollections $\{\mathcal{R}^b : b \in \{2,\dots,2k\}\}$, where for each b, the rectangles in $\mathcal{R}^b$ are disjoint and each $R \in \mathcal{R}^b$ satisfies*

$$m(R) = m_1(R) = m_2(R) \ge \frac{1}{2}q_1^{bk}n$$
$$\alpha_1(R),\alpha_2(R) \ge \frac{\gamma'\delta'}{8k^2S(n,\frac{360kq_1}{\gamma'}m(R))}.$$

Proof. For each $b \in \{2,\dots,2k\}$, apply the previous lemma with $I = I^b = C^b$ 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 $|J| = \sum_b |J^b| \ge \sum_b \left(|I^b|(1-\gamma'/2) - |D^n|\delta'/2k\right) \ge |F^{-1}(1)|(1-\gamma'/2) - |D^n|\delta'$. The rectangles in $\mathcal{Q}$ are all 3-balanced. We would like to replace this by a collection of balanced rectangles. Consider $(Q,A_1,A_2) \in \mathcal{Q}$ and without loss of generality assume that $|A_2| \le |A_1|$. From the conclusion of Lemma 4.12, we have $\alpha_1(Q),\alpha_2(Q) \ge \delta'/(4k^2S(n,\frac{360kq_1}{\gamma'}|A_2|))$ since $|A_1| \le 3|A_2|$. Choose $B_1 \subseteq A_1$ such that $|B_1| = |A_2|$ and consider the (B_1,A_2)-refinement of Q, Refine(Q,B_1,A_2) as defined in section 2.2. It is easy to see that $\alpha_2(P) = \alpha_2(Q)$

for every $P \in \text{Refine}(Q, B_1, A_2)$. Also, each $P \in \text{Refine}(Q, B_1, A_2)$ satisfies $|P| = \alpha_1(P)|D|^{|B_1|}$. Since there are at most $|D|^{|A_1|-|B_1|}$ such rectangles, the number of points covered by rectangles P with $\alpha_1(P) \leq \frac{\gamma'}{2}\alpha_1(Q)$ is at most $\frac{\gamma'}{2}\alpha_1(Q)|D|^{|A_1|} = \frac{\gamma'}{2}|Q|$. Thus, if we replace Q by the set of rectangles $P \in \text{Refine}(Q, B_1, A_2)$ with $\alpha_1(P) \geq \frac{\gamma'}{2}\alpha_1(Q)$, we 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 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 $|J'| \geq |J|(1 - \gamma'/2) \geq |F^{-1}(1)|(1 - \gamma') - |D^n|\delta'$. The conclusion of Lemma 4.12, together with the lower bound on $\mu(x, q)$ given by Lemma 4.3 implies that the rectangles in $\mathcal{R}$ have the claimed properties. □

5 Embedded Rectangles in Branching Programs

Lemmas 4.4 and 4.13 showed that every suitably small decision forest admits a nice family of rectangles that covers most of the accepted inputs. In this section, we use the connection between branching programs and decision forests given by Lemma 2.2 together with these two lemmas to show that most inputs accepted by an efficient branching program can be covered by a nice family of rectangles.

5.1 Embedded rectangles with small leg-density

We use the simple bound given in Lemma 4.4 to show the existence of embedded rectangles in branching programs.

Theorem 5.1. *Let $k \geq 4$ be an integer and $n \geq r \geq 2^{k+6}k^2$. Let $\mathcal{B}$ be a branching program of length at most $(k-2)n$ and size 2^S. There is a collection $\mathcal{R}$ of disjoint embedded rectangles such that each rectangle $R \in \mathcal{R}$ is a subset of $\mathcal{B}^{-1}(1)$ and satisfies $m(R) = m_1(R) = m_2(R) \geq n/2^{k+1}$ and $\alpha_1(R), \alpha_2(R) \geq 2^{-12(k+1)m(R)-2-Sr}|\mathcal{B}^{-1}(1)|/|D|^n$, and such that $\bigcup_{R\in\mathcal{R}} R$ covers at least $|\mathcal{B}^{-1}(1)|/2$ inputs in $\mathcal{B}^{-1}(1)$.*

Proof. By Lemma 2.2, there is a family $\mathcal{S}$ consisting of 2^{Sr} inquisitive $(r, k/r)$-decision forests, such that $\mathcal{B} = \bigvee_{F\in\mathcal{S}} F$. Note that the collection of sets $\{F^{-1}(1) : F \in \mathcal{S}\}$ partitions $\mathcal{B}^{-1}(1)$.

For each forest $F \in \mathcal{S}$, apply Lemma 4.4 with $\gamma' = 1/4$, and $\delta' = |\mathcal{B}^{-1}(1)|/(2^{Sr+2}|D^n|)$, and let $\mathcal{R}_F$ be the family of embedded rectangles obtained in the conclusion of the lemma. Define $\mathcal{R} = \bigcup_{F\in\mathcal{S}} \mathcal{R}_F$. By construction, $\bigcup_{R\in\mathcal{R}} R$ covers a subset of $\mathcal{B}^{-1}(1)$ of size at least $(1-\gamma')|\mathcal{B}^{-1}(1)| - 2^{Sr}\delta'|D^n| = \frac{1}{2}|\mathcal{B}^{-1}(1)|$, and each $R \in \mathcal{R}$ has $m(R) = m_1(R) = m_2(R) \geq n/2^{k+1}$ and $\alpha_1(R), \alpha_2(R) \geq 2^{-12(k+1)m(R)}\delta' = 2^{-12(k+1)m(R)-2-Sr}|\mathcal{B}^{-1}(1)|/|D|^n$ as required. □

Corollary 5.2. *Let $k \geq 4$ be an integer and $n \geq r \geq 2^{k+6}k^2$. Let $\mathcal{B}$ be an n-variate branching program over domain D of length at most $(k-2)n$ and size 2^S.*

(i) [BST98, BJS01] There is an embedded rectangle R contained in $\mathcal{B}^{-1}(1)$ satisfying $m(R) = m_1(R) = m_2(R) \geq n/2^{k+1}$ and $\alpha(R) \geq 2^{-12(k+1)m(R)-Sr-2}|\mathcal{B}^{-1}(1)|/|D|^n$.

(ii) Let f be an n-variate decision function over D and suppose $\mathcal{B}$ 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 $\mathcal{B}^{-1}(1)$ satisfying $m(R) = m_1(R) = m_2(R) \geq n/2^{k+1}$ and $\alpha(R) \geq 2^{-12(k+1)m(R)-Sr-2}(\delta - \epsilon)$, such that f is 0 on at most a $2\epsilon/(\delta-\epsilon)$ fraction of points in R.

Proof. Apply Theorem 5.1 and let $\mathcal{R}$ be the resulting collection of rectangles contained in $\mathcal{B}^{-1}(1)$. The first part of the corollary follows by choosing any $R \in \mathcal{R}$. For the second part, the hypothesis on $\mathcal{B}$ implies that $|\mathcal{B}^{-1}(1)|/|D^n| \geq \delta - \epsilon$ so all rectangles $R \in \mathcal{R}$ satisfy $\alpha(R) \geq 2^{-12(k+1)m(R)-Sr-2}(\delta - \epsilon)$ and together they cover at least $(\delta - \epsilon)|D^n|/2$ points in $\mathcal{B}^{-1}(1)$. Since $\mathcal{B}$ and f differ on at most $\epsilon|D^n|$ inputs, f is 0 for at most $\epsilon|D^n|$ of the at least $(\delta-\epsilon)|D^n|/2$ points covered by $\mathcal{R}$. Since the rectangles in $\mathcal{R}$ are disjoint, there must be some rectangle $R \in \mathcal{R}$ in which f is 0 on at most $2\epsilon/(\delta - \epsilon)$ fraction of the points in R. $\square$

5.2 Embedded rectangles with large leg-density

Now, using a similar argument and Lemma 4.13 in place of Lemma 4.4, we derive our main theorem which is more widely applicable than Theorem 5.1.

Theorem 5.3. *Let $k \geq 8$ be an integer, $q_1 \leq 2^{-40}k^{-8}$, $n \geq r \geq 200k^2/q_1^{4k^2}$. Let $\mathcal{B}$ be a branching program of length at most $(k-2)n$ and size 2^S. There is a collection $\mathcal{R}$ of embedded rectangles that satisfies:*

1. *Each rectangle of $\mathcal{R}$ is contained in $\mathcal{B}^{-1}(1)$.*
2. $\left|\bigcup_{R\in\mathcal{R}} R\right| \geq |\mathcal{B}^{-1}(1)|/2$.
3. *No input belongs to more than $2k-1$ rectangles of $\mathcal{R}$.*
4. *Each rectangle $R \in \mathcal{R}$ satisfies $m(R) = m_1(R) = m_2(R) \geq q_1^{2k^2}n/2$ and $\alpha(R) \geq 2^{-q_1^{1/2}m(R)-Sr}|\mathcal{B}^{-1}(1)|/|D|^n$.*

Proof. By Lemma 2.2, there is a family $\mathcal{S}$ consisting of 2^{Sr} inquisitive $(r, k/r)$-decision forests, such that $\mathcal{B} = \bigvee_{F\in\mathcal{S}} F$. Note that the collection of sets $\{F^{-1}(1) : F \in \mathcal{S}\}$ partitions $\mathcal{B}^{-1}(1)$.

For each forest $F \in \mathcal{S}$, apply Lemma 4.13 with $\gamma' = 1/4$ and $\delta' = |\mathcal{B}^{-1}(1)|/(2^{Sr+2}|D^n|)$, 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 four conditions asserted in the conclusion of the theorem.

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

Again by the conclusion of Lemma 4.13 each $R \in \mathcal{R}_F$ has $m(R) = m_1(R) = m_2(R) \geq q_1^{2k^2}n/2$ and $\alpha(R) \geq |\mathcal{B}^{-1}(1)|/(2^{Sr}|D^n|128k^2 S(n, 1440kq_1m(R)))$. It remains only to show that, under the given hypotheses on k, q_1, r and n, this last quantity is at least the claimed lower bound on $\alpha(R)$, and for this it suffices to show that the following two inequalities hold:

$$\begin{aligned} 128k^2 &\leq 2^{q_1^{1/2}m(R)/2}, \\ S(n, 1440kq_1m(R)) &\leq 2^{q_1^{1/2}m(R)/2}. \end{aligned}$$

For the first inequality, we note that $q_1^{1/2}m(R)/2 \geq q_1^{2k^2+1/2}n/4 \geq 128k^2$ by the hypotheses on q_1 and n. For the second inequality, let $\beta = 1440kq_1m(R)/n$. Since $\beta \in [0,1]$, we have $S(n, \beta n) = \sum_{k\leq\beta n}\binom{n}{k} \leq \sum_{k\leq\beta n}\binom{n}{k}\beta^{k-\beta n} \leq \beta^{-\beta n}(1+\beta)^n \leq (e/\beta)^{\beta n}$. Therefore, it suffices to show $(e/\beta)^{\beta n} \leq 2^{q_1^{1/2}m(R)/2}$

which is equivalent to showing $(e/\beta)^{2880k(q_1)^{1/2}} \leq 2$. Since $m(R) \geq q_1^{2k^2} n/2$, $e/\beta \leq q_1^{-4k^2}$ and so:

$$(e/\beta)^{2880k(q_1)^{1/2}} \leq \left(\frac{1}{q_1}\right)^{11520k^3(q_1)^{1/2}} \leq \left(\frac{1}{q_1}\right)^{q_1^{1/8}} \leq 2$$

since $x^{1/x^{1/8}} \leq 2$ for all $x \geq 2^{48}$. $\square$

In the above theorem, we obtain a rectangle cover that misses at most half of the points in $\mathcal{B}^{-1}(1)$. By a straightforward but tedious change in the analysis, we can strengthen the conclusion so that the fraction of uncovered points of $\mathcal{B}^{-1}(1)$ is arbitrarily small. This stronger version is not needed for the branching program lower bounds.

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

Corollary 5.4. *Let $k \geq 8$ be an integer, $q_1 \leq 2^{-40}k^{-8}$, $n \geq r \geq q_1^{-5k^2}$. Let $\mathcal{B}$ be an n-variate branching program over domain D of length at most $(k-2)n$ and size 2^S.*

(i) There is an embedded rectangle R contained in $\mathcal{B}^{-1}(1)$ satisfying $m(R) = m_1(R) = m_2(R) \geq q_1^{2k^2} n/2$ and $\alpha(R) \geq 2^{-q_1^{1/2} m(R) - Sr} |\mathcal{B}^{-1}(1)|/|D|^n$.

(ii) Let f be an n-variate decision function over D and suppose $\mathcal{B}$ 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 $\mathcal{B}^{-1}(1)$ satisfying $m(R) = m_1(R) = m_2(R) \geq q_1^{2k^2} n/2$ and $\alpha(R) \geq 2^{-q_1^{1/2} m(R) - Sr}(\delta - \epsilon)$, such that f is 0 on at most a $4\epsilon k/(\delta - \epsilon)$ fraction of points in R.

Proof. Apply Theorem 5.3 (noting that the lower bound on r in the hypothesis of the present corollary 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 $|\mathcal{B}^{-1}(1)|/|D^n| \geq \delta - \epsilon$, so all rectangles in $\mathcal{R}$ satisfy $\alpha(R) \geq 2^{-q_1^{1/2} m(R) - Sr}(\delta - \epsilon)$, and together the rectangles cover at least $(\delta - \epsilon)|D^n|/2$ inputs of $\mathcal{B}^{-1}(1)$. Call an input x *bad* if $\mathcal{B}(x) \neq f(x)$ 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 $\frac{|Bad(R)|}{|R|} \leq 4\epsilon k/(\delta - \epsilon)$. $\square$

6 Lower Bounds

We now use our two branching program characterizations from Corollaries 5.2 and 5.4 to derive lower bounds for a number of natural decision problems on branching programs and random access machines. These include two general classes of problems: problems based on quadratic forms over finite fields and problems involving all pairwise comparisons between input variables over large domains such as element distinctness and a related problem that we call Hamming closeness.

Most of these bounds apply for domains in which each input variable is from a relatively large domain D. For these our largest lower bounds are of the form $T = \Omega(n \log(\frac{n \log n}{S}))$ and are the largest known for general non-uniform computation of problems in NP. More importantly, we also obtain lower bounds for the most interesting case of Boolean computation and improve bounds of Ajtai [Ajt99b] for the computation of quadratic forms over $GF(2)$ to $T = \Omega(n\sqrt{\log(n/S)/\log\log(n/S)})$ which is substantially better than the best previous bounds for Boolean problems. In all cases, our lower bounds apply to randomized computation with 2-sided error as well as to deterministic computation.

There are three axes on which to consider our lower bounds: the function analyzed, deterministic versus randomized computation, and whether the bounds of Corollary 5.2 can be used or the bounds of Corollary 5.4 are required. (In general, where it is applicable, Corollary 5.2 will give larger lower bounds but its applicability is much more limited than Corollary 5.4.)

We analyze the problems related to quadratic forms first. For these problems, we present the arguments for the randomized case directly since the proof that no large embedded rectangles with small error exist is not much more involved than the one that no large error-free embedded rectangles exist. The lower bound for Boolean branching programs, which requires Corollary 5.4, is given in Theorem 6.6; then the lower bounds over large finite fields using Corollary 5.2 are given in Theorems 6.9 and 6.10.

We then analyze the element distinctness and Hamming closeness problems. For these problems, the deterministic analysis is much simpler than the randomized analysis so we present the deterministic lower bounds separately first. The methods for element distinctness and Hamming closeness are very similar to each other although the element distinctness lower bounds require the use of Corollary 5.4, but Corollary 5.2 suffices for Hamming closeness. The reader who wishes to see the simplest application of the embedded rectangle methods is encouraged to read to the deterministic lower bounds for element distinctness first.

While we state all of our results for branching programs, they apply to general sequential computation [BC82], which includes random access machines (even non-uniform ones) and Turing machines. For the functions defined over large domains in particular, the implications of these results for random access machines are especially natural and we include a number of these corollaries.

6.1 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 quadratic form function on D^n given by $F_M(x) = x^T Mx$.

Inspired by Borodin, Razborov, and Smolensky's use of bilinear forms to prove lower bounds for read-k branching programs [BRS93], Beame, Jayram, and Saks, [BST98, BJS01] considered quadratic forms over finite fields. In particular, taking M to be a slightly modified version of any of the Sylvester matrix used in [BRS93], and employing a variant of the deterministic case of Corollary 5.2, they showed that determining whether or not $F_M(x) = 0$ requires $T = \Omega(\min\{n\log\log n, n\log(\frac{n\log n}{S})\})$.

To derive this lower bound they extended ideas of Borodin, Razborov, and Smolensky to show that, in the case $D = GF(p)$ for prime power p, if M is a symmetric matrix that is *rigid*, in the sense that all submatrices of M have rank that is suitably large relative to their size, then $x^T Mx$ cannot be constant on any large embedded rectangle. The lower bounds for the quadratic forms based on modified Sylvester matrices follow from a lower bound on the rigidity of Sylvester matrices.

By considering different quadratic form functions in the case $D = GF(2)$, Ajtai constructed an explicit family of boolean functions which cannot be computed by a deterministic branching program of subexponential size and linear length [Ajt99b]. To derive non-trivial bounds, in addition to a more generally applicable branching program analysis than Corollary 5.2(i), he required matrices with larger rigidity than that of Sylvester matrices. Using our stronger branching program analysis we obtain a $T = \Omega(n\sqrt{\log(n/S)/\log\log(n/S)})$ lower bound for this Boolean function and a $T = \Omega(n\log(\frac{n\log n}{S}))$

lower bound for related functions over larger finite fields, both of which hold for randomized branching programs.

We begin by strengthening the previous properties of quadratic forms derived from matrix rigidity for use with randomized branching programs. As a preliminary we analyze bilinear forms and show that for an $m \times m$ matrix P of suitably high rank and $c \in GF(p)$, any large rectangle $U \times V \in GF(p) \times GF(p)$ has many pairs (u, v) of vectors such that $u^T Pv = c$.

Lemma 6.1. *Let P be an $m \times m$ matrix over $GF(p)$ where p is a prime power. Let $U, V \subseteq GF(p)^m$, $\alpha = |U|/p^m$, and $\beta = |V|/p^m$. Let $c \in GF(p)$ and N_c be the number of pairs $(u, v) \in U \times V$ such that $u^T Pv = c$. Then $|N_c - |U||V|/p|$ is at most $\frac{p-1}{\sqrt{\alpha\beta p^{\mathrm{rank}(P)}}}|U||V|/p$*

Proof. Let $r = \mathrm{rank}(P)$ and for $x \in GF(p)$ define

$$\gamma_c(x) = \begin{cases} 1 - \frac{1}{p} & \text{if } x = c \\ -\frac{1}{p} & \text{if } x \neq c. \end{cases}$$

Let $\Delta = N_c - |U||V|/p$ and observe that $\Delta = \sum_{u \in U} \sum_{v \in V} \gamma_c(u^T Pv)$.

Each $x \in GF(p)$ can be realized as a k-dimensional vector over $GF(q)$ for some prime q and some integer k. For $x, y \in GF(p)$, define $\langle x, y\rangle$ to be the scalar product of vectors corresponding to x, y. For $h \in GF(p)$, let $\chi_h(x) = e^{2\pi i\langle h,x\rangle/q}$ be characters of $GF(p)$. γ_c can be expressed as a linear combination of χ_h (for $h \in GF(p)$) as follows

$$\gamma_c = \sum_{h \in GF(p)} \alpha_j \chi_j,$$

where $\alpha_0 = 0$ and $\alpha_h = \frac{1}{p} e^{-2\pi i\langle h,c\rangle/q}$ for $h \neq 0$.

We can write $P = K^T L$ where K and L are each $r \times m$ matrices of rank r over $GF(p)$. Define integer-valued vectors a, b indexed by $GF(p)^r$ where $a(y)$ for $y \in GF(p)^r$ is the number of vectors $u \in U$ such that $Ku = y$ and $b(z)$ for $z \in GF(p)^r$ is the number of vectors $v \in V$ such that $Lv = z$.

For $h \in GF(p)$ let M_h be the $p^r \times p^r$ complex matrix whose rows and columns are indexed by vectors in $GF(p)^r$ with $M_h(y, z) = \chi_h(y^T z)$ where $y^T z$ is the inner product mod p. Observe that for $h \in GF(p)$, $M_h^* M_h(y_1, y_2) = \sum_{z \in GF(p)^r} \chi_h^*(y_1^T z)\chi_h(y_2^T z) = \sum_{z \in GF(p)^r} \chi_h((y_2 - y_1)^T z)$ which is 0 if $y_1 \neq y_2$ and

p^r if $y_1 = y_2$. Then

$$\begin{aligned}
\Delta^2 &= (\sum_{u\in U}\sum_{v\in V}\gamma_c(u^T K^T L v))^2 \\
&= (\sum_{h\in GF(p),h\neq 0}\alpha_h \sum_{u\in U}\sum_{v\in V}\chi_h(u^T K^T L v))^2 \\
&\leq (p-1)\sum_{h\in GF(p),h\neq 0}|\alpha_h|^2\left|\sum_{u\in U}\sum_{v\in V}\chi_h(u^T K^T L v)\right|^2 \\
&= \frac{p-1}{p^2}\sum_{h\in GF(p),h\neq 0}\left|\sum_{u\in U}\sum_{v\in V}\chi_h(u^T K^T L v)\right|^2 \\
&= \frac{p-1}{p^2}\sum_{h\in GF(p),h\neq 0}\left|\sum_{y\in GF(p)^r}\sum_{z\in GF(p)^r}\chi_h(y^T z)a(y)b(z)\right|^2 \\
&= \frac{p-1}{p^2}\sum_{h\in GF(p),h\neq 0}\left|a^T M_h b\right|^2 \\
&\leq \frac{p-1}{p^2}\sum_{h\in GF(p),h\neq 0}\|a\|^2\|M_h b\|^2 \\
&= \frac{p-1}{p^2}\sum_{h\in GF(p),h\neq 0}\|a\|^2 b^T M_h^* M_h b \\
&= \frac{p-1}{p^2}\sum_{h\in GF(p),h\neq 0}\|a\|^2\|b\|^2 p^r \\
&= \frac{(p-1)^2}{p^2}\|a\|^2\|b\|^2 p^r
\end{aligned}$$

where $\|\cdot\|$ is the Euclidean norm. (The inequalities on the third and seventh line are applications of the Cauchy-Schwartz inequality; the fifth line follows using the definitions of a and b and the next-to-last line follows by the properties of M_h.)

Now $\|a\|^2 \leq \max_x a(x)\sum_x a(x) \leq p^{m-r}|U|$, and similarly $\|b\|^2 \leq p^{m-r}|V|$. Thus $\Delta^2 \leq (1-\frac{1}{p})^2 p^{2m-r}|U||V|$ and so

$$\Delta \leq \sqrt{\frac{(p-1)^2 p^{2m}}{|U||V|p^r}}|U||V|/p = \frac{p-1}{\sqrt{\alpha\beta p^r}}|U||V|/p$$

as required. □

Lemma 6.2. *Let M be a $n\times n$ matrix with entries in $GF(p)$ and suppose that (R, A_1, A_2) is an embedded rectangle in $GF(p)^n$ with $|A_1| = |A_2|$. Let P be the submatrix of $M + M^T$ induced on $A_1 \times A_2$. Suppose that $\alpha(R) \geq p^{3-\mathrm{rank}(P)/2}$. Then for each $c \in GF(p)$, the fraction of inputs of $x \in R$ for which $x^T M x \equiv c \pmod p$ is more than $1/(4p)$.*

Proof. Let $m = |A_1| = |A_2|$, $A_0 = [n] - A_1 - A_2$, and, for $(i,j)\in\{0,1,2\}^2$, let $M_{i,j}$ denote the submatrix of M indexed by $A_i \times A_j$. Let σ be the spine of R. For $x \in R$, writing x_1 for x_{A_1} and x_2 for x_{A_2} we have:

$$x^T M x = C + f_1(x_1) + f_2(x_2) + x_1^T P x_2$$

where $C = \sigma^T M_{0,0}\sigma$, $P = M_{1,2} + M_{2,1}^T$ and for $j \in \{1,2\}$, f_j is a function with domain R_{A_j} which is defined by $f_j(x_j) = x_j^T M_{j,j} x_j + x_j^T(M_{j,0} + M_{0,j}^T)\sigma$.

We partition R based on the values of the f_j on the R_{A_j} for $j = 1,2$. For each pair $(c_1, c_2) \in GF(p)^2$, let $U_{c_1} = f_1^{-1}(c_1) \cap R_{A_1}$ and $V_{c_2} = f_2^{-1}(c_2) \cap R_{A_2}$.

For at least 1/2 the elements $x \in R$ the unique (c_1, c_2) such that $(x_1, x_2) \in U_{c_1} \times V_{c_2}$ satisfy $|U_{c_1}||V_{c_2}| \geq |R|/(2p^2)$ which implies that $|U_{c_1}||V_{c_2}|/p^{2m} \geq p^{4-\mathrm{rank}(P)}/2$. For each such $(x_1, x_2) \in U_{c_1} \times V_{c_2}$, we have $(x_1 x_2 \sigma)^T M (x_1 x_2 \sigma) = C + c_1 + c_2 + x_1^T P x_2$. By Lemma 6.1, $x_1^T P x_2 = c - C - c_1 - c_2$ for more than $|U_{c_1}||V_{c_2}|/(2p)$ of the pairs $(x_1, x_2) \in U_{c_1} \times V_{c_2}$. For each such pair, $x = x_1 x_2 \sigma$ is a point in R such that $x^T M x = c$. Since these elements over all such (c_1, c_2) account for at least 1/2 of R, in total more than a $1/(4p)$ fraction of the points $x \in R$ have $x^T M x = c$. $\square$

Combining this lemma with Corollary 5.4 gives the following result which says that if M is a matrix over $GF(2)$ 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 6.3. *Let n, r, k be positive integers and $q_1 > 0$ with $k \geq 8$, $q_1 \leq 2^{-40}k^{-8}$, $n \geq r \geq q_1^{-5k^2}$. Let M be an $n \times n$ matrix with entries in $GF(2)$, with associated quadratic form function f. Suppose that $\mathcal{B}$ is an n-variate branching program over $GF(2)$ of length at most $(k-2)n$ and size 2^S that disagrees with f on at most a fraction $1/80k$ of 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 $2Sr + 2q_1^{1/2} m + 10$.*

Proof. Let $b \in \{0,1\}$ 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 $\mathcal{B}'$ analogously from $\mathcal{B}$ by replacing output 0 by $b-1$ and output 1 by b.

Applying the second part of Corollary 5.4 to f' and $\mathcal{B}'$ with $\delta \geq 1/2$ and $\epsilon = 1/80k$, we get a balanced rectangle $R = (R, A_1, A_2)$ contained in $(\mathcal{B}')^{-1}(1)$ satisfying $m(R) = |A_1| = |A_2| \geq q_1^{2k^2} n/2$ and $\alpha(R) \geq 2^{-q_1^{1/2} m(R) - Sr}(\frac{1}{2} - \frac{1}{80k}) \geq 2^{-q_1^{1/2} m(R) - Sr - 2}$ such that f' is 0 on at most a $4(1/80k)k/(1/2 - 1/80k) < 1/8$ fraction of points of R. By Lemma 6.2 with $p = 2$, $\alpha(R)$ must be less than $2^{3-\mathrm{rank}(P)/2}$. Combining the upper and lower bounds on $\alpha(R)$ we deduce $\mathrm{rank}(P) \leq 2Sr + 2q_1^{1/2} m(R) + 10$. $\square$

This theorem can be applied to give time-space tradeoffs for the quadratic form function for any matrix M over $GF(2)$ 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 [BRS93, BST98] have the property that every $s \times s$ submatrix has rank at least s^2/n. However, this property is not strong enough to get good time-space tradeoff lower bounds over $GF(2)$ using Theorem 6.3.

Ajtai looked instead at Hankel matrices, matrices whose every anti-diagonal is constant. Given a vector $y \in GF(p)^{2n-1}$, 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 $GF(p)$. (Here a random Hankel matrix means a matrix $H[y]$ where y is chosen uniformly at random from $GF(p)^{2n-1}$.)

Lemma 6.4. *[Ajt99b, Lemma 9] 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 $GF(p)$, the probability that there is some $s \times s$ sub-matrix of H of rank less than R is at most*

$$\binom{n}{Qt}^2 \binom{Q}{Q-R+1} p^{-\frac{1}{4}(Q-R+1)t^2}.$$

As a direct consequence we get:

Corollary 6.5. *Let n be an integer and H be a random $n \times n$ Hankel matrix over $GF(p)$. With probability at least 1/2, for all integers s satisfying $(1024 + 64 \log_p n)^2 < s < n$ every $s \times s$ submatrix of H has rank at least $\frac{1}{2}s/(1024 + 64 \log_p(n/s))^2 - 2$.*

Proof. Let s be an integer in the range given in the hypothesis, let $t = \lfloor 1024 + 64 \log_p(n/s) \rfloor$, let $Q = \lfloor s/t^2 \rfloor$, and $R = \lfloor Q/2 \rfloor$ (which is at least $\frac{1}{2}s/(1024 + 64 \log_p(n/s))^2 - 2$). By Lemma 6.4, the probability that H has an $s \times s$ submatrix of rank less than R is at most:

$$\begin{aligned}\binom{n}{Qt}^2 \binom{Q}{Q-R+1} p^{-\frac{1}{4}(Q-R+1)t^2} &\leq \left(\frac{en}{Qt}\right)^{2Qt} 2^Q p^{-s/8} \leq \left(\frac{4n}{Qt}\right)^{2Qt} p^{-s/8} \\ &\leq \left(\frac{4nt}{s}\right)^{2s/t} p^{-s/8} \leq p^{-s/16} \leq 2^{-s/16}.\end{aligned}$$

The first inequality uses $(Q - R + 1)t^2 \geq s/2$ and $\binom{n}{k} \leq (en/k)^k$. The second uses $2^{1/2t}e \leq \sqrt{2}e \leq 4$ and the third uses the fact that uses that $(4n/k)^k$ is increasing in k for $k < n$, and that $Qt \leq s/t < n$. The fourth inequality uses $(4nt/s) \leq p^{t/32}$ which follows from $t < 2^{t/64} \leq p^{t/64}$ and $4n/s < p^{t/64}$.

Finally, summing the above bound over all integers $s \geq s_0 = \lceil (1024 + 64 \log_p n)^2 \rceil$, we get that the failure probability is at most $2^{-s_0/16}/(1 - 2^{-1/16})$. Using $1 - 2^{-x} > x/2$ for $x \in (0, 1)$ this is at most $2^{-s_0/16+5}$ which is easily seen to be less than 1/2. ☐

The rigidity property of random Hankel matrices above is strong enough but to prove a tradeoff for the function F_M using Theorem 6.3 we need that $M + M^T$ be rigid rather than M itself. Hankel matrices are symmetric which means that $M + M^T = 0$ since we are over $GF(2)$, which seems like a serious problem. Ajtai showed how this problem is easily overcome. 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 we can apply Theorem 6.3 to $L(M)$ instead of M.

Another important issue is that we want lower bounds for explicit functions; we already know that hard functions exist by simple counting arguments. A random Hankel matrix does not give an explicit function. However, 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 6.6. *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 $T \geq c'n\sqrt{\log(n/S)/\log\log(n/S)}$.*

Proof. Choose n to be a sufficiently large integer. Let $\widetilde{\mathcal{B}}$ be a randomized branching program with input variables $x_1, \ldots, x_n, y_1, \ldots, y_{2n-1}$ of length at most $(k - 2)n$ and size 2^S and suppose that the probability $\widetilde{\mathcal{B}}(x, y) \neq G(x, y)$ is less than $1/160k$. We want to show that for some constant c', $k \geq c'\sqrt{\log(n/S)/\log\log(n/S)}$.

We will apply Theorem 6.3 and to this end, we assume without loss of generality that $k \geq 8$ and define for $q_1 = 2^{-40}k^{-8}$, and $r = \lceil q_1^{-5k^2} \rceil$. If $n < r^2$, then $k \geq c\sqrt{\log n/\log\log n}$ for some c, and the desired result is trivial. So we may assume that k is such that $n \geq r^2$.

By Proposition 2.1, we can fix a deterministic branching program $\mathcal{B}$ that agrees with $G(x, y)$ on at least a $1 - 1/160k$ fraction of all inputs. For each y, let $\mathcal{B}[y]$ be the branching program obtained from $\mathcal{B}$ by hardwiring the values of y, and let $\epsilon(y)$ be the fraction of inputs x such that $\mathcal{B}[y](x) \neq x^T H[y]x$. Let Y be the set of all $y \in Y$, such that $H[y]$ has the rigidity property specified in the conclusion of Corollary 6.5. By that corollary, half of all y belong to Y. Therefore, there must exist a $y \in Y$ such that $\epsilon(y) < 1/80k$. Fix such a y and apply Theorem 6.3 for the matrix $H[y]$ and the branching program $\mathcal{B}[y]$. We conclude that there are disjoint subsets A_1, A_2 of $[n]$, of equal size $m \geq q_1^{2k^2} n/2$ such that the submatrix M of $H[y]$ induced on $A_1 \times A_2$ has rank at most $2Sr + 2q_1^{1/2}m + 10$. For sufficiently large n and any k satisfying the restrictions at the beginning of the proof, $m = q_1^{2k^2} n \geq \sqrt{n} \geq (1024 + 64\log_2 n)^2$ and since $y \in Y$, Corollary 6.5 implies that M has rank at least $b = \frac{1}{2}m/(1024 + 64\log_2(n/m))^2 - 2$. Again, for sufficiently large n, it is easy to see $2q_1^{1/2}m + 10 < b/2$ so we conclude that $2Sr > b/2 > 2q_1^{1/2}m$ which implies $S \geq q_1^{1/2}m/r$ which is at least n/k^{ck^2} for some constant $c > 0$. It follows that for some constant $c' > 0$ and sufficiently large n, $k \geq \sqrt{\log(n/S)/\log\log(n/S)}$. $\square$

Corollary 6.7. *For any $\epsilon < 1/2$, 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 ϵ requires $T \geq cn\log^{1/2}(n/S)/(\log\log(n/S))^{3/2}$.*

Proof. Let $\mathcal{B}$ be such a branching program. As in standard probability amplification if one run of a randomized algorithm has error at most $\epsilon < 1/2$, taking the majority answer from some $c_\epsilon \log\log(n/S)$ independent copies of the algorithm run on the same input suffices to reduce the error to less than $1/\log(n/S)$. This can be computed by chaining together $c_\epsilon \log\log(n/S)$ copies of $\mathcal{B}$ where each node is also replicated $c_\epsilon \log\log(n/S)$ times at each time step to store the running tally of the number of copies of $\mathcal{B}$ in which 1 has been produced so far. The resulting branching program will have time $T' = c_\epsilon T \log\log(n/S)$ and space $S' \leq S + c'_\epsilon \log\log\log(n/S)$ for some constant $c'_\epsilon > 0$. Applying Theorem 6.6 implies a lower bound on T' (and thus the desired lower bound on T) since if T is smaller than $c'n\log^{1/2}(n/S)/(\log\log(n/S))^{3/2}$ the error $c'n/T'$ permitted there is at least $\sqrt{\log\log(n/S')/\log(n/S')}$ which is larger than $1/\log(n/S)$. $\square$

We also can define quadratic form functions over large domains for which we can obtain even larger lower bounds using Corollary 5.2. We could apply the deterministic part of this corollary (implicit in [BST98]) to the problem of determining if $F_M(x) = 0$ with M and x over $GF(p)$ for $p \geq n$ to derive lower bounds of the form $T = \Omega(n\log(\frac{n\log n}{S}))$. To be able to show bounds for constant-error randomized algorithms we define related functions that are more balanced between outputs 0 and 1.

For M and $n \times n$ matrix over $GF(p)$ and $G \subseteq GF(p)$, define $F_{M,G} : GF(p)^n \to \{0, 1\}$ by $F_{M,G}(x) = 1$ iff $x^T Mx \in G$. Natural examples of sets G to choose include elements with low order bit equal 1, the quadratic residues modulo p in the case p is prime, or if $p = 2^{p'}$ some linear map $\phi : GF(p) \to GF(2)$.

Theorem 6.8. *Let $n \geq r \geq k$ be positive integers with $k \geq 4$ and $r \geq 2^{k+6}k^2$. Let M be an $n \times n$ matrix with entries in $GF(p)$ for $p > 2$ a prime power, and let $G \subset GF(p)$ be any set of size $\lfloor p/2 \rfloor$. Suppose that $\mathcal{B}$ is an n-variate branching program over $GF(p)$ of length at most $(k-2)n$ and size 2^S that disagrees with $F_{M,G}$ on at most a $1/50$ fraction of inputs. Then there are two disjoint subsets $A_1, A_2 \subseteq [n]$ with $|A_1| = |A_2| = m$ where $m \geq n/2^{k+1}$ such that the submatrix of $P = M + M^T$ induced by $A_1 \times A_2$ has rank at most $(2Sr + 24(k+1)m(R) + 8)/\log_2 p + 6$.*

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

Applying the second part of Corollary 5.2 to f and $\mathcal{B}'$ with $\delta \geq 1/2$ and $\epsilon = 1/50$, we get a balanced rectangle $R = (R, A_1, A_2)$ contained in $(\mathcal{B}')^{-1}(1)$ satisfying $m(R) = |A_1| = |A_2| \geq n/2^{k+1}$ and $\alpha(R) \geq 2^{-12(k+1)m(R)-Sr-2}(\frac{1}{2} - \frac{1}{50}) \geq 2^{-12(k+1)m(R)-Sr-4}$ such that f is 0 on at most a $2(1/50)/(1/2 - 1/50) =$

$1/12$ fraction of points of R. There are at least $(p-1)/2$ elements $c \in GF(p)$ such that $x^T Mx = c$ implies $f(x) = 0$. Therefore there is some value $c \in GF(p)$ such that $x^T Mx = c$ for at most a $\frac{1}{6(p-1)} \le 1/(4p)$ fraction of points of R. By Lemma 6.2, $\alpha(R)$ must be less than $p^{3-\mathrm{rank}(P)/2}$. Combining the upper and lower bounds on $\alpha(R)$ we deduce $\mathrm{rank}(P) \le 6 + (2Sr + 24(k+1)m(R) + 8)/\log_2 p$. ☐

We now apply this to the functions $J^n_{p,G} : GF(p)^{3n-1} \to \{0,1\}$ based on Hankel matrices over $GF(p)$ given by $J^n_{p,G}(x,y) = F_{L(H[y]),G}(x)$; i.e. $J^n_{p,G}$ is 1 if and only if $x^T L(H[y])x \in G$.

Theorem 6.9. *Let $p \ge n$ be a prime power. Let $G \subset GF(p)$ with $|G| = \lfloor p/2 \rfloor$. Any randomized $GF(p)$-way branching program computing $J^n_{p,G}$ in time T and size 2^S with probability of error at most $1/100$ requires $T = \Omega(n \log(\frac{n \log n}{S}))$.*

Proof. Choose n to be a sufficiently large integer. Let $\widetilde{\mathcal{B}}$ be a randomized $GF(p)$-way branching program with input variables $x_1, \ldots, x_n, y_1, \ldots, y_{2n-1}$ of length at most $(k-2)n$ and size 2^S and suppose that the probability $\widetilde{\mathcal{B}}(x,y) \ne J_{p,G}(x,y)$ is less than $1/100$. We want to show that for some constant c', $k \ge c' \log(\frac{n \log n}{S})$.

We will apply Theorem 6.8 and to this end, we assume without loss of generality that $k \ge 4$ and define $r = 2^{k+6}k^2$. Assume that $k \le 2^{-29} \log_2 n$ since otherwise the theorem follows immediately.

By Proposition 2.1, we can fix a deterministic branching program $\mathcal{B}$ that disagrees with $J_{p,G}(x,y)$ on at most a $1/100$ fraction of all inputs. For each y, let $\mathcal{B}[y]$ be the branching program obtained from $\mathcal{B}$ by hardwiring the values of y, and let $\epsilon(y)$ be the fraction of inputs x such that $\mathcal{B}[y](x)$ does not compute the value of the predicate $x^T L(H[y])x \in G$. Let Y be the set of all $y \in Y$, such that $H[y]$ has the rigidity property specified in the conclusion of Corollary 6.5. By that corollary, half of all y belong to Y. Therefore, there must exist a $y \in Y$ such that $\epsilon(y) \le 1/50$.

Fix such a y and apply Theorem 6.8 for the matrix $H[y]$ and the branching program $\mathcal{B}[y]$. We conclude that there are disjoint subsets A_1, A_2 of $[n]$, of equal size $m \ge n/2^{k+1}$, such that the submatrix M_R of $L(H[y]) + L(H[y])^T$ induced on $A_1 \times A_2$ has rank at most $(2Sr + 24(k+1)m + 8)/\log_2 p + 6$. Observe that, since Hankel matrices are symmetric and M_R contains no elements from the diagonal of $L(H[y]) + L(H[y])^T$, M_R is a submatrix of $H[y]$. Therefore by Corollary 6.5, since $m \ge n/2^{k+1}$ the submatrix M_R has rank at least $\frac{1}{2}m/(1024 + 64\log_p(n/m))^2 - 2 \ge m/2^{23} - 2$. Thus $(2Sr + 24(k+1)m + 8)/\log_2 p + 6 \ge m/2^{23} - 2$ and so

$$
\begin{aligned}
S &\ge \frac{1}{r}([2^{-24}\log_2 p - 12(k+1)]m - 4\log_2 p - 4) \\
&\ge \frac{1}{k^2 2^{k+6}}([2^{-24}\log_2 p - 12(k+1)]n/2^{k+1} - 4\log_2 p - 4) \\
&\ge C\frac{1}{k^2 2^{2k}} n \log_2 n
\end{aligned}
$$

for some constant C and n sufficiently large since $\log_2 p \ge \log_2 n > 2^{29}(k+1)$ in this case. This implies that $k \ge c' \log_2(\frac{n \log n}{S})$ for some constant $c' > 0$. ☐

It is easy to see that, by applying probability amplification using a constant number of independent copies of the branching program as in Corollary 6.7, the asymptotic bound of Theorem 6.9 also applies for any error $\epsilon < 1/2$.

We can also applying Theorem 6.8 to the *modified Sylvester matrices* studied in [BST98]; for example, these matrices include the standard Hadamard matrices over $\{1, -1\}$ with their diagonals replaced by the value 0. This result extends the bound of [BST98, BJS01] to randomized branching programs.

Theorem 6.10. *Let $p \geq n$ be a prime power. Let M be a modified Sylvester matrix over $GF(p)$ and $G \subset GF(p)$ with $|G| = (p-1)/2$. Any randomized $GF(p)$-way branching program computing $F_{M,G}$ in time T and size 2^S with probability of error at most $1/50$ requires $T = \Omega(\min\{n \log\log n, n \log(\frac{n \log n}{S}\}))$.*

6.2 Element Distinctness and Hamming Closeness

Ajtai [Ajt99a] gave the first general time-space tradeoff lower bounds for element distinctness and also gave lower bounds for a related problem he defined which we call the Hamming closeness problem.

The bounds for Hamming closeness use significantly simpler techniques than those for element distinctness. While Ajtai only claimed non-trivial time-space tradeoff lower bounds for Hamming closeness when time T is linear in n, Pagter observed [Pag01] that by optimizing the technique of [Ajt99a], one can derive non-trivial lower bounds for Hamming closeness when $T \leq cn \log n / \log\log n$ for some $c > 0$ and these bounds apply in the presence of 1-sided error (but not nondeterminism, which is stronger).

However, the technique of [BST98] mentioned in Table 1 and predating [Ajt99a] yields even better tradeoffs for Hamming closeness. It applies when $T \leq cn \log n$ for some constant $c > 0$ and even applies to nondeterministic branching programs. Corollary 5.2 extends this technique to 2-sided error randomized computation and we show how it applies to the Hamming closeness problem. Since the deterministic lower bound is considerably simpler we present it first.

To derive lower bounds for element distinctness, Ajtai developed most of the machinery required for his Boolean branching program lower bounds. Again he only claimed to produce lower bounds when time T is linear in n but, as in the Boolean case, a careful analysis of his arguments show that they apply even when T is as large as $cn \log\log n / \log\log\log n$ for some $c > 0$. Our results improve this range up to $T \leq cn\sqrt{\log n / \log\log n}$ and generalize the bounds to randomized branching programs with 2-sided error.

Deterministic Branching Programs

To prove a lower bound for a deterministic branching program all we need is a lower bound on the fraction of inputs on which the function takes on a given value as well as an upper bound on the sizes of embedded rectangles on which the function can take on that value.

Element Distinctness

Define the element distinctness function $ED : D^n \rightarrow \{0, 1\}$ is 1 if and only if there is no pair $i \neq j \in X$ such that $x(i) = x(j)$.

Proposition 6.11. *If $|D| \geq n^2$, at least a $1/e$ fraction of all inputs $x \in D^n$ have $ED(x) = 1$.*

Proof. It is easy to check that for $N \geq n^2$, $N!/(N-n)! \geq N^n/e$. □

Lemma 6.12. *Let $ED : D^n \rightarrow \{0, 1\}$. 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 6.13. *Any $[n^2]$-way deterministic branching program computing $ED : [n^2]^n \to \{0,1\}$ 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 $\mathcal{B}$ of length $(k-2)n$ and size 2^S for ED. Apply Corollary 5.4(i) with $q_1 = 2^{-40}k^{-8}$ and $r = \lceil q_1^{-5k^2} \rceil$. We obtain an embedded rectangle on which $\mathcal{B}$ 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 6.12, 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 $(k-2)n$ requires space at least $k^{-ck^2}n$. Substituting $T = (k-2)n$, and re-arranging we obtain the claimed tradeoff. □

Corollary 6.14. *For any $\epsilon \geq 0$ there is a constant c_ϵ, such that any RAM algorithm for element distinctness on inputs in $[n^2]$ taking at most $c_\epsilon n\sqrt{\log n/\log\log n}$ time requires at least $n^{1-\epsilon}$ space.*

Hamming Closeness

We now define the Hamming closeness problem HAM_γ for $0 < \gamma < 1/2$. For $0 < \gamma < 1/2$ let $c = c(\gamma) \geq 0$ be minimum such that for all $b \geq 1$, $2^{2b}/S(\gamma cb, cb) \leq 1$ where $S(d,n)$ is the size of the Hamming ball of radius d about a vector of length n. Recall that $S(d,n) = \sum_{j \leq d}\binom{n}{j} \leq 2^{H_2(d/n)n}$ where H_2 is the binary entropy function, $H_2(q) = -q\log_2 q - (1-q)\log_2(1-q)$. Observe by this bound on $S(d,n)$ that $c(\gamma) \leq 2/(1-H_2(\gamma))$.

Given two elements $u, v \in [N]$ we say that u and v are *γ-close* to each other if the Hamming distance between u and v represented in binary, $\Delta_H(u,v) \leq \gamma\log_2 N$ and *γ-far* from each other otherwise. (Also given two subsets $U, V \subset [N]$ we say that U and V are *γ-close* if there is a pair of elements $u \in U$ and $v \in V$ that are γ-close to each other, and U and V are *γ-far* if all such pairs of elements are γ-far from each other.) The Hamming closeness problem $HAM_\gamma : [N]^n \to \{0,1\}$ where N is a power of 2 and $0 < \gamma < 1/2$. $HAM_\gamma(x_1,\ldots,x_n) = 1$ if and only if there is some pair of indices $i \neq j$ such that x_i and x_j are γ-close to each other.

The following propositions are minor variants of those shown by Ajtai [Ajt99a].

Proposition 6.15. *For $0 < \gamma < 1/2$, there is a constant $c(\gamma) = 2/(1-H_2(\gamma))$ such that for $c \geq c(\gamma)$ and $N = n^c$, $HAM_\gamma^{-1}(0)$ contains at least $1/2$ of all inputs in $[N]^n$.*

Proof. Let $c \geq 2/(1-H_2(\gamma))$ and $b = \log_2 N = c\log_2 n$ be the number of bits in each x_i. The Hamming ball of radius γb about any element of $[N]$ contains $S(\gamma b, b) \leq 2^{H_2(\gamma)b}$ elements, where we recall that $S(d,n) = \sum_{j\leq d}\binom{n}{j} \leq 2^{H_2(d/n)n}$. Therefore the fraction of pairs x and y from $[N]$ that are γ-close to each other is at most

$$2^{H_2(\gamma)b}/2^b = 2^{-(1-H_2(\gamma))b} = 2^{-(1-H_2(\gamma))c\log_2 n} \leq n^{-2}.$$

There are $\binom{n}{2}$ pairs (x_i, x_j) with $i \neq j$ so HAM_γ has value 1 for at most $\binom{n}{2}n^{-2} \leq 1/2$ of the inputs in $[N]^n$. □

Proposition 6.16. *For $0 < \gamma < 1/2$, there is a constant $\beta = \beta(\gamma) > 0$ such that any two sets $U, V \subseteq [N]$ with $|U|, |V| \geq N^{1-\beta}$ are γ-close.*

Proof. Let $b = \log_2 N$ and choose β as large as possible such that $S((\frac{1-\gamma}{2})b, b) < 2^{(1-\beta)b}$. (Thus $\beta > 0$ will be roughly $1 - H_2(\frac{1-\gamma}{2})$ since $\log_2 S(d,n)$ is asymptotic to $H_2(d/n)n$ as $n \to \infty$ and d/n is fixed.)

By the classic isoperimetric bound of Harper [Har66], the γb-Hamming neighborhood of U will contain a set of size $> S((\frac{1+\gamma}{2})b, b)$ and thus will only miss a set of size $< S((\frac{1-\gamma}{2})b, b) < 2^{(1-\beta)b} \leq |V|$. $\square$

Lemma 6.17. *Let $0 < \gamma < 1/2$ and $HAM_\gamma : [N]^n \to \{0,1\}$. Then there is a constant $\beta = \beta(\gamma) > 0$ such that any embedded rectangle $R \subseteq [N]^n$ with $HAM_\gamma(x) = 0$ for all $x \in R$ has $\alpha(R) \leq N^{-\beta m(R)}$.*

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). Let $\beta = \beta(\gamma) > 0$ be the constant from Proposition 6.16. $HAM_\gamma(x) = 0$ for all $x \in R$ implies S_1 and S_2 are not γ-close, so by Proposition 6.16 for some index h $|S_h| \leq N^{1-\beta}$. Thus $\alpha_h(R) \leq (|S_h|/N)^{m_h(R)} \leq N^{-\beta m(R)}$. $\square$

Theorem 6.18. *Let $\gamma < 1/2$ and $c \geq 2/(1 - H_2(\gamma))$. Any $[n^c]$-way deterministic branching program computing $HAM_\gamma : [n^c]^n \to \{0,1\}$ in time T and size 2^S requires $T = \Omega(n \log(\frac{n \log n}{S}))$.*

Proof. Let $\mathcal{B}$ be a deterministic branching program of length $(k-2)n$ and size 2^S computing HAM_γ. Therefore there is branching program $\mathcal{B}'$ of the same length and size computing $\neg HAM_\gamma$ on $[N] = [n^c]$. By Proposition 6.15, $\mathcal{B}'$ outputs 1 on at least $1/2$ of the inputs in $[N]^n$. Assume without loss of generality that $k \leq \frac{1}{8} \log_2 n$ and apply Corollary 5.2(i) with $r = 2^{k+6}k^2$ to $\mathcal{B}'$ to obtain an embedded rectangle (R, A_1, A_2) on which HAM_γ is 0 satisfying $m(R) = m_1(R) = m_2(R) \geq n/2^{k+1}$ and $\alpha(R) \geq 2^{-12(k+1)m(R)-Sr-3}$. Using Lemma 6.17, this means $2^{-12(k+1)m(R)-Sr-3} \leq N^{-\beta m(R)}$ for some constant $\beta = \beta(\gamma) > 0$. Therefore $S > ((\beta \log_2 N - 3)m(R) - 12(k+1))/r \geq (C2^{-3k} n \log_2 n)$ for some $C > 0$ for n sufficiently large and the theorem follows. $\square$

Corollary 6.19. *For any $\epsilon \geq 0$ and γ with $0 < \gamma < 1/2$ and $c \geq 2/(1 - H_2(\gamma))$ there is a constant $c_{\epsilon,\gamma} > 0$, such that any RAM algorithm for HAM_γ on inputs in $[n^c]$ taking at most $c_{\epsilon,\gamma} n \log n$ time requires at least $n^{1-\epsilon}$ space.*

Randomized Branching Programs

We now consider randomized branching programs for both element distinctness and Hamming closeness. We already have lower bounds on the fraction of inputs for which these functions take on particular values. To apply the second parts of Corollaries 5.2 and 5.4, respectively, we will now need to show that any rectangle on which the functions mostly have those values cannot be very dense. The two arguments are very similar to each other but unfortunately will only be able to tolerate rather small error.

Element Distinctness

Lemma 6.20. *If (R, A_1, A_2) is an n-variate embedded rectangle over $[N]$ with $|A_1| = |A_2| = m$ such that at most an $\epsilon < \frac{1}{24}$ fraction of $x \in R$ have $ED(x) = 0$ then*

$$\alpha(R) \leq 8(8/9)^{m/2} \max\{1, (8/9)^{m/2} 2^{\frac{H_2(8\epsilon)}{3} N/m}\}.$$

Proof. Call a point of D^m *non-repetitive* if all the coordinates are distinct and *repetitive* otherwise. For $j \in \{1,2\}$, let Q_j be the set of non-repetitive points of R_{A_j}. Let Q be the subrectangle of R having $Q_{A_j} = Q_j$ for each j. Clearly $\alpha_j(Q) \geq \frac{\alpha_j(R)}{2}$ since otherwise R has too many points x with $ED(x) = 0$. Thus rectangle $R' \subseteq R$ with legs Q_1 and Q_2 has $\alpha(R') \geq \alpha(R)/2$. Observe that each element of Q_j

corresponds to an m-subset of $[N]$ and that for any point $x \in R'$ with $ED(x) = 1$ the sets corresponding to the $x_{Q_1} \in Q_1$ and $x_{Q_2} \in Q_2$ must be disjoint.

We will now apply an argument used by Babai, Frankl, and Simon [BFS86] to derive a lower bound on ϵ-error communication complexity for this set-disjointness problem. (Note that the arguments used later by Kalayanasundaram and Schnitger [KS87] or Razborov [Raz90] to get optimal communication complexity bounds are not useful to us because these require precise linear relationships between the set and universe sizes.) However, because different m-subsets of $[N]$ may correspond to different numbers of the m-permutations in the Q_j, we will need to argue that our translation into the set-disjointness problem preserves not only the size of the rectangle involved but also its relative error. For this we apply the following easy lemma.

Lemma 6.21. *Let Y and Z be sets such that $|Y| = M \cdot |Z|$; let $S : Y \to Z$ be such that for all $z \in Z$, $|S^{-1}(z)| = M$ and let $f : Z \to [0,1]$. If $Y' \subseteq Y$ satisfies $|Y'| \geq \beta |Y|$ and $\mathrm{E}_{y \in Y'}[f(S(y))] \leq \epsilon$ then there is a $Z' \subseteq Z$ such that $|Z'| \geq \frac{\beta}{2}|Z|$ and for every $z \in Z'$, $f(z) \leq 2\epsilon$.*

Proof. By Markov's inequality there is a set $Y'' \subseteq Y'$ such that $|Y''| \geq |Y'|/2 \geq \frac{\beta}{2}|Y|$ such that for all $y \in Y''$, $f(S(y)) \leq 2\epsilon$. Define $Z' = S(Y'')$. Then $|Z'| \geq |Y''|/M \geq \frac{\beta}{2}|Y|/M = \frac{\beta}{2}|Z|$ and for $z \in Z'$ there is some $y \in Y''$ such that $f(z) = f(S(y)) \leq 2\epsilon$. ☐

Given sets T and T' define the indicator variable $\chi_{T,T'}$ to be 1 if $T \cap T' \neq \emptyset$ and 0 otherwise. We first apply Lemma 6.21 with Y the set of m-permutations of $[N]$, $Z = \binom{[N]}{m}$, the set of m-subsets of $[N]$, $Y' = Q_2$, S the map from an m-permutation to its corresponding m-subset, and, for $T' \in \binom{[N]}{m}$, $f(T')$ equal to $\mathrm{E}_{q \in Q_1}[\chi_{S(q),T'}]$. Since $ED(x) = 0$ for at most an ϵ fraction of elements of R', $\mathrm{E}_{y \in Q_2}[f(S(y))] \leq \epsilon$. Therefore by Lemma 6.21 we obtain a set G of m-subsets of $[N]$ such that $|G| \geq \frac{\alpha(R)}{4}\binom{N}{m}$ and for every $T' \in G$, $\mathrm{E}_{q \in Q_1}[\chi_{S(q),T'}] \leq 2\epsilon$ so

$$\mathrm{E}_{q \in Q_1} \mathrm{E}_{T' \in G}[\chi_{S(q),T'}] \leq 2\epsilon.$$

We now apply Lemma 6.21 again with the same values of S, Y, and Z but with $Y' = Q_1$ and $f(T) = \mathrm{E}_{T' \in G}[\chi_{T,T'}]$ We obtain a set F of m-subsets of $[N]$ such that $|F|, |G| \geq \frac{\alpha(R)}{4}\binom{N}{m}$ and for every $T \in F$, $\mathrm{E}_{T' \in G}[\chi_{T,T'}] \leq 4\epsilon \leq 1/6$; i.e. each element of F intersects at most $4\epsilon \leq 1/6$ of all elements of G. The following is a simple generalization of part of the argument in [BFS86].

Proposition 6.22. *Let $d \geq 3$ and let F be a collection of m-subsets of $[N]$. If $|F| > 2(4(d-1)/d^2)^{m/2}\binom{N}{m}$ then F contains a sequence of $p = \lceil N/(dm) \rceil$ sets $S_1, \ldots, S_p$ such that $|S_j \cap \bigcup_{i<j} S_i| \leq m/2$ for $j = 1, \ldots, p$, i.e. at least half the elements of S_j do not occur in earlier sets.*

Proof. We construct $S_1, \ldots, S_p$ inductively. Select $S_1 \in F$ arbitrarily. For $j > 1$, having chosen $S_1, \ldots, S_{j-1}$, we show that for $j \leq p$, the number of sets that have more than half their elements in earlier sets is less than $|F|$ and so we can select $S_j \in F$ as required. Let $U_j = \bigcup_{i<j} S_i$. Since $j \leq \lceil N/(dm) \rceil$, $|U_j| \leq N/d$, the number of m-subsets of $[N]$ having more than half their elements in U_j is at most $\sum_{h \geq m/2} \binom{|U_j|}{h}\binom{n-|U_j|}{m-h} \leq \sum_{h \geq m/2} \binom{N/d}{h}\binom{(1-1/d)N}{m-h}$. It is easy to check that since $d \geq 3$ as h increases, each successive term is at most half the previous so the sum is at most $2\binom{N/d}{\lceil m/2 \rceil}\binom{(1-1/d)N}{\lfloor m/2 \rfloor}$. Using the easily verifiable inequalities that for $b \geq a \geq c \geq d$, $\binom{a}{c}\binom{b}{d} < (\frac{4ab}{a+b})^c \binom{(a+b)/2}{c}\binom{(a+b)/2}{d}$ and $\binom{N/2}{c}\binom{N/2}{d} \leq \binom{N}{c+d}$ we upper bound this strictly by $2(4(d-1)/d^2)^{m/2}\binom{N}{m}$ which is less than $|F|$. ☐

If $\alpha(R) \leq 8(8/9)^{m/2}$ then we are done. Otherwise, applying the proposition with $d = 3$ to our set F, we can find $p = \lceil N/(3m) \rceil$ sets $S_1, \ldots, S_p$ in F each of which contains at least $m/2$ elements not occurring in earlier sets. For $T \in G$, let $w(T)$ be the number of S_j that intersect it. Since each $S_j \in F$, $\frac{1}{|G|}\sum_T w(T) \leq 4\epsilon p$, so at most half of the $T \in G$ have $w(T) > 8\epsilon p$. Let G' be the set of $T \in G$ with $w(T) \leq 8\epsilon p$. Thus $|G'| \geq |G|/2$.

We now upper bound the number of elements in G' and thus G. An element T of G' can be described by giving a subset $J \subseteq [p]$ of $(1-8\epsilon)p$ indices such that $T \cap S_j = \emptyset$ for all $j \in J$ and then specifying T as an m-subset of the elements outside these subsets. By the claim, any collection of $(1-8\epsilon)p$ of the sets has a total of $m(1-8\epsilon)p \geq N/9$ elements since $\epsilon \leq \frac{1}{24}$. Therefore $|G| \leq 2|G'| \leq 2\binom{p}{8\epsilon p}\binom{8N/9}{m} < 2^{1+H_2(8\epsilon)N/(3m)}(8/9)^m\binom{N}{m}$ and thus $\alpha(R) \leq 2^{3+H_2(8\epsilon)N/(3m)}(8/9)^m$. □

Theorem 6.23. *There is a constant $c > 0$ such that any randomized $[n^2]$-way branching program computing $ED : [n^2]^n \to \{0,1\}$ in time T and size 2^S with probability of error at most $(T/n)^{-c(T/n)^2}$ requires $T = \Omega(n\sqrt{\log(n/S)/\log\log(n/S)})$. Furthermore, any randomized $[n^2]$-way branching program computing $ED : [n^2]^n \to \{0,1\}$ in time T and size 2^S with probability of error at most S/n requires $T = \Omega(n\sqrt{\log(n/S)\log\log(n/S)})$.*

Proof. By Proposition 2.1, 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 5.4 and to this end, we assume without loss of generality that $k \geq 8$ and define for $q_1 = 2^{-40}k^{-8}$, and $r = \lceil q_1^{-5k^2}\rceil$. There is a constant $c > 0$ such that for all positive $\epsilon \leq (k-2)^{-c(k-2)^2}$ and $k \geq 8$, $H_2(8\epsilon) \leq 16\epsilon\log_2\frac{1}{\epsilon} \leq \frac{1}{16}(32k)^{-32k^2}$ and $\frac{4\epsilon k}{1/e-\epsilon} \leq \frac{1}{24}$. Fix such a c and assume that $\epsilon \leq (k-2)^{-c(k-2)^2}$.

Let $\mathcal{B}$ be a deterministic branching program of length at most $(k-2)n$ and size 2^S that approximates ED within ϵ. 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 5.4(ii) and the fact that ED is 1 for least a $1/e$ fraction of all inputs in $[n^2]$ and $1/e - \epsilon > 1/4$, we obtain a balanced rectangle R with $m = m(R) \geq q_1^{2k^2}n/2$ and $\alpha(R) \geq 2^{-Sr-q_1^{1/2}m-5}$ such that ED is 0 for at most a fraction $\frac{4k\epsilon}{(1/e-\epsilon)} \leq \frac{1}{24}$ of inputs in R by our assumption on ϵ. Applying Lemma 6.20, we have that $\alpha(R) \leq 8(8/9)^{m/2}\max\{1, (8/9)^{m/2}2^{\frac{H_2(8\epsilon)}{3}n^2/m}\}$. Now $n^2/m \leq 4mq_1^{-4k^2} \leq 4m(2^{40}k^8)^{4k^2} = 4m(32k)^{32k^2}$ so

$$
\begin{aligned}
(8/9)^{m/2}2^{\frac{H_2(8\epsilon)}{3}n^2/m} &\leq 2^{-m/12+\frac{4H_2(8\epsilon)}{3}m(32k)^{32k^2}} \\
&\leq 2^{-\frac{m}{3}(1/4-4H_2(8\epsilon)(32k)^{32k^2})} \\
&\leq 1
\end{aligned}
$$

since $H_2(8\epsilon) \leq \frac{1}{16}(32k)^{-32k^2}$. Therefore $\alpha(R) \leq 8(8/9)^{m/2} \leq 2^{3-m/12}$. Combining the upper and lower bounds on $\alpha(R)$ and simplifying we get $2^{Sr} \geq 2^{m/12-q_1^{1/2}m-8}$, which, for n sufficiently large and k satisfying the restrictions above, is at least $2^{m/13}$. From this we deduce $S \geq \frac{m}{13r} \geq r^{-1}q_1^{2k^2}n/26$ which is at least $c_0 n/k^{c_1k^2}$ for some c_0, c_1 independent of n and k. It follows that for some constant $c' > 0$ and sufficiently large n, $k \geq c'\sqrt{\log(n/S)/\log\log(n/S)}$.

Also observe that our conditions on ϵ hold for $\epsilon = r^{-1}q_1^{2k^2}/26$ which is our lower bound on S/n and from this the second part of the theorem follows. □

Corollary 6.24. *For any $\delta \geq 0$ there is a constant c_δ such that for n sufficiently large any randomized RAM* *algorithm for element distinctness on inputs in $[n^2]$ taking at most $c_\delta n\sqrt{\log n/\log\log n}$ time and having at most $n^{-\delta}$ error requires at least $n^{1-\delta}$ space.*

Hamming Closeness

Lemma 6.25. *If (R, A_1, A_2) is an n-variate embedded rectangle over $[N]$ with $|A_1| = |A_2| = m$ such that at most an $\epsilon < \frac{1}{24}$ fraction of $x \in R$ have $HAM_\gamma(x) = 1$ then*

$$\alpha(R) \le 8(36N^{-\beta})^{m/2} \max\{1, (N^{-\beta}/36)^{m/2} 2^{3H_2(8\epsilon)N^{1-\beta}/m}\}.$$

Proof. As in the proof of Lemma 6.20 but replacing the condition $ED(x) = 0$ with $HAM_\gamma(x) = 1$ we find two collections of m-subsets of $[N]$, F and G, such that $|F|, |G| \ge \alpha(R)/4\binom{N}{m}$ and for each $T \in F$ at most a 4ϵ fraction of $T' \in G$ have the property that there exist $t \in T$ and $t' \in T'$ such that $\Delta_H(t, t') \le \gamma \log_2 N$.

Let $d = N^\beta/3$ where the value of $\beta = \beta(\gamma)$ is given by Proposition 6.16. By Proposition 6.22, if $\alpha(R) > 8(36N^{-\beta})^{m/2} > 8(4(d-1)/d^2)^{m/2}$ then F contains $p = \lceil N/(dm)\rceil$ sets $S_1, \ldots, S_p$ such that each set S_j has at most $m/2$ elements occurring in S_i for $i < j$.

As in the proof of Lemma 6.20, prune G to obtain a set G' with $|G'| \ge |G|/2$ such that each element $T' \in G$ is γ-close to at most $8\epsilon p$ of $S_1, \ldots, S_p$. We can describe a set T' in G' by naming the $(1-8\epsilon)p$ sets S_i from which it is γ-far and then specifying each element in T' from among the elements that are γ-far from these sets. The $(1-8\epsilon)p$ sets together contain at least $mp/3 \ge N/(3d) = N^{1-\beta}$ elements of $[N]$ and so by Proposition 6.16 at most $N^{1-\beta}$ elements of $[N]$ are γ-far from these sets. Therefore

$$|G| \le 2|G'| \le 2\binom{p}{8\epsilon p}\binom{N^{1-\beta}}{m} \le 2^{1+3H_2(8\epsilon)N^{1-\beta}/m} N^{-\beta m}\binom{N}{m}$$

and thus $\alpha(R) \le 8N^{-\beta m} 2^{3H_2(8\epsilon)N^{1-\beta}/m}$ from which the lemma follows. □

Although the parameters in this lemma make it appear stronger than Lemma 6.20, the value of N for which we will need to use it is much larger than in the case of element distinctness and we will only be able to obtain lower bounds for much smaller error. It is conceivable that a much stronger result holds since (1) the collection of sets $S_1, \ldots, S_p$ in Proposition 6.22 was chosen to maximize the unions of subcollections rather than to maximize the number of elements that are γ-close to these subcollections, and (2) no use was made of the property that each set in F and G must only contain elements that are γ-far from each other.

Theorem 6.26. *Let $0 < \gamma < 1/2$ and $c \ge 2/(1 - H_2(\gamma))$. Any randomized $[n^c]$-way branching program computing $HAM_\gamma : [n^c]^n \to \{0, 1\}$ in time T and size 2^S with probability of error at most n^{2-c} requires $T = \Omega(n \log(\frac{n \log n}{S}))$.*

Proof. By Proposition 2.1, it suffices to prove the lower bound for deterministic branching programs that approximate HAM_γ within error $\epsilon = n^{2-c}$.

Choose n to be a sufficiently large integer. Let $k \ge 4$ and let $\mathcal{B}$ be a deterministic branching program of length at most $(k-2)n$ and size 2^S approximates HAM_γ within ϵ. Therefore the branching program $\mathcal{B}'$, set to be $\mathcal{B}$ with the labels of its two sink nodes swapped, approximates $\neg HAM_\gamma$ within ϵ.

We will show that for some $c' > 0$, $k \ge c' \log((n \log n)/S)$. Let $\beta = \beta(\gamma) > 0$ be the constant from Proposition 6.16. If $k \ge (c\beta \log_2 n)/32$ then we are done so assume without loss of generality that $2^k < n^{c\beta/32}$. Let $r = 2^{k+6}k^2$ and apply Corollary 5.2(ii) to $\mathcal{B}'$ to obtain a balanced rectangle R

with $m = m(R) \geq n/2^{k+1}$ and $\alpha(R) \geq 2^{-12(k+1)m-Sr-5}$ such that HAM_γ is 1 for at most a fraction $\frac{2\epsilon}{(1/2-\epsilon)} \ll \frac{1}{24}$ of inputs in R.

Applying Lemma 6.25 with $N = n^c$, we have

$$\alpha(R) \leq 8(36n^{-c\beta})^{m/2} \max\{1, (n^{-c\beta}/36)^{m/2} 2^{3H_2(8\epsilon)n^{c(1-\beta)}/m}\}.$$

Now since $m^2 \geq n^2/2^{2k+2}$, $n^{c(1-\beta)}/m \leq 2^{2k+2} n^{c(1-\beta)-2} m$ so

$$\begin{aligned}(n^{-c\beta}/36)^{m/2} 2^{3H_2(8\epsilon)n^{c(1-\beta)}/m} &\leq (n^{-c\beta/2} 2^{12H_2(8\epsilon)2^{2k}n^{c-2-c\beta}}/6)^m \\ &\leq (n^{-c\beta/2} 2^{12H_2(8\epsilon)n^{c-2-c\beta/2}}/6)^m \\ &\leq 1\end{aligned}$$

where the second line follows from our assumption on k and the third follows since $H_2(8\epsilon) \leq 16(c-2)n^{2-c}\log_2 n \leq \frac{1}{12} n^{2-c+c\beta/2}$ for n sufficiently large.

Therefore $\alpha(R) \leq 8(36n^{-c\beta})^{m/2} \leq 2^{3-m(\frac{c\beta}{2}\log_2 n - 3)}$. Combining the upper and lower bounds on $\alpha(R)$ and simplifying we get $2^{Sr} \geq 2^{m(\frac{c\beta}{2}\log_2 n-(12k+15))-8} \geq 2^{\frac{c\beta}{8}m\log_2 n}$ by our assumptions on k. From this we deduce $S \geq c\beta 2^{-2k-2\log_2 k-10} n\log_2 n \geq c\beta 2^{-3k-14} n \log_2 n$ since $k \geq 4$. Rearranging, we obtain $k \geq c' \log(\frac{n\log n}{S})$ for some positive constant c'. $\square$

7 Discussion

The time-space tradeoff lower bounds we obtain for decision problems on general randomized branching programs are nearly as good as the best lower bounds known even for the much simpler oblivious deterministic branching programs. The best lower bounds in the oblivious case have all been obtained using some form of communication complexity. Using two-party communication complexity, Alon and Maass [AM88] derived lower bounds of the form $T = \Omega(n \log(n/S))$ and using multi-party communication complexity, Babai, Nisan, and Szegedy [BNS92] derived the best current lower bounds which are of the form $T = \Omega(n \log^2(n/S))$.

The use of rectangles in our results as well as all those referenced in Table 1 is related to 2-party communication complexity (see e.g. [KN97]) and most of the difficulty in these arguments is in extending the bounds from the oblivious to the general case. In fact, the basic approach provides an alternate way to obtain the same bounds as [AM88] for oblivious branching programs (see the discussion prior to Lemma 4.4). Recently, these methods have been extended [BV02] to include multi-party communication complexity ideas which yield an alternate way to obtain the bounds of [BNS92] for oblivious branching programs. These results also extend the technique of [BJS01] using multi-party communication complexity ideas to obtain lower bounds over large domains. However, it is not at all clear if it is possible to extend results to include multi-party communication complexity ideas in the Boolean case using the ideas of [Ajt99b]; a key stumbling block seems to be the lack of a multi-party analogue of Lemma 4.10 in that case.

A larger goal would be to extend these lower bounds to apply when time is $n(\log n)^{\omega(1)}$ and even achieve tradeoffs for decision problems such as $T = \Omega(n^2/S)$, a bound we already can prove for multi-output problems such as sorting. Attempting to show this first for oblivious branching programs seems like a good way to start.

Finally, we remark that in our lower bounds the error bounds on the randomized algorithms that our arguments tolerate vary a great deal from problem to problem. We are able to obtain time-space tradeoff lower bounds for branching programs whose error approaches 1/2 when solving the quadratic form problems. However, for the element distinctness and Hamming closeness problems, the bounds we prove are for error that is inverse polynomial in the input size; probability amplification of these bounds does not yield non-trivial lower bounds for error approaching 1/2.

Acknowledgements

Many thanks to Jeff Edmonds and Michal Koucky for their helpful comments on drafts of this paper. Also many thanks to Martin Sauerhoff for his comments on technical points of the paper.

References

[Abr90] Karl 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, October 1990. IEEE.

[Abr91] Karl R. Abrahamson. Time–space tradeoffs for algebraic problems on general sequential models. *Journal of Computer and System Sciences*, 43(2):269–289, October 1991.

[Ajt98] 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.

[Ajt99a] 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*, pages 632–641, Atlanta, GA, May 1999.

[Ajt99b] M. Ajtai. A non-linear time lower bound for boolean branching programs. In *Proceedings 40th Annual Symposium on Foundations of Computer Science*, pages 60–70, New York,NY, October 1999. IEEE.

[AM88] Noga Alon and Wolfgang Maass. Meanders and their applications in lower bounds arguments. *Journal of Computer and System Sciences*, 37:118–129, 1988.

[BC82] Allan Borodin and Stephen 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.

[Bea91] Paul W. Beame. A general time-space tradeoff for finding unique elements. *SIAM Journal on Computing*, 20(2):270–277, 1991.

[BFK$^+$81] Allan Borodin, Michael J. Fischer, David G. Kirkpatrick, Nancy A. Lynch, and Martin Tompa. A time-space tradeoff for sorting on non-oblivious machines. *Journal of Computer and System Sciences*, 22(3):351–364, June 1981.

[BFMadH$^+$87] Allan Borodin, Faith E. Fich, Friedhelm Meyer auf der Heide, Eli Upfal, and Avi Wigderson. A time-space tradeoff for element distinctness. *SIAM Journal on Computing*, 16(1):97–99, February 1987.

[BFS86] László Babai, P. Frankl, and Janos Simon. Complexity classes in communication complexity theory. In *27th Annual Symposium on Foundations of Computer Science*, pages 337–347, Toronto, Ontario, October 1986. IEEE.

[BJS01] Paul Beame, T. S. Jayram, and Michael Saks. Time-space tradeoffs for branching programs. *Journal of Computer and System Sciences*, 63(4):542–572, December 2001.

[BNS92] László Babai, Noam Nisan, and Márió Szegedy. Multiparty protocols, pseudorandom generators for logspace, and time-space trade-offs. *Journal of Computer and System Sciences*, 45(2):204–232, October 1992.

[Bor93] Allan Borodin. Time space tradeoffs (getting closer to the barrier?). In *4th International Symposium on Algorithms and Computation*, pages 209–220, Hong Kong, December 1993.

[BRS93] Allan Borodin, A. A. Razborov, and Roman Smolensky. On lower bounds for read-k times branching programs. *Computational Complexity*, 3:1–18, October 1993.

[BST98] Paul W. Beame, Michael Saks, and Jayram S. Thathachar. Time-space tradeoffs for branching programs. In *Proceedings 39th Annual Symposium on Foundations of Computer Science*, pages 254–263, Palo Alto, CA, November 1998. IEEE.

[BV02] Paul Beame and Erik Vee. Time-space tradeoffs, multiparty communication complexity, and nearest-neighbor problems. In *Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing*, pages 688–697, Montreal, Quebec, Canada, May 2002.

[For97] L. Fortnow. Nondeterministic polynomial time versus nondeterministic logarithmic space: Time-space tradeoffs for satisfiability. In *ProceedingsTwelfth Annual IEEE Conference on Computational Complexity (formerly: Structure in Complexity Theory Conference)*, pages 52–60, Ulm, Germany, June 1997.

[FvM00] L. Fortnow and D. van Melkebeek. Time-space tradeoffs for nondeterministic computation. In *ProceedingsFifteenth Annual IEEE Conference on Computational Complexity*, pages 2–13, Florence, Italy, July 2000.

[Har66] L.H. Harper. Optimal numberings and isoperimetric problems on graphs. *Journal of Combinatorial Theory*, 1:385–394, 1966.

[HLP52] G. H. Hardy, J. E. Littlewood, and G. Polya. *Inequalities*. Cambridge University Press, 1952.

[KN97] E. Kushilevitz and N. Nisan. *Communication Complexity*. Cambridge University Press, Cambridge [England] ; New York, 1997.

[KS87] B. Kalyanasundaram and Georg Schnitger. The probabilistic communication complexity of set intersection. In *Proceedings, Structure in Complexity Theory, Second Annual Conference*, pages 41–49, Cornell University, Ithaca, NY, June 1987. IEEE.

[LV99] R. Lipton and A. Viglas. Time-space tradeoffs for SAT. In *Proceedings 40th Annual Symposium on Foundations of Computer Science*, pages 459–464, New York,NY, October 1999. IEEE.

[MNT93] Y. Mansour, N. Nisan, and P. Tiwari. The computational complexity of universal hashing. *Theoretical Computer Science*, 107:121–133, 1993.

[Oko93] E. Okol'nishnikova. On lower bounds for branching programs. *Siberian Advances in Mathematics*, 3(1):152–166, 1993.

[Pag01] J. Pagter. *Time-Space Tradeoffs*. PhD thesis, BRICS, March 2001.

[Raz90] A. A. Razborov. On the distributional complexity of disjointness. In Michael S. Paterson, editor, *Automata, Languages, and Programming: 17th International Colloquium*, volume 443 of *Lecture Notes in Computer Science*, pages 249–253, Warwick University, England, July 1990. Springer-Verlag.

[Yao88] 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, October 1988. IEEE.