Theoretical Computer Science 38 (1985) 69-82
North-Holland

LOWER BOUNDS ON PROBABILISTIC LINEAR DECISION TREES

Marc SNIR
Department of Computer Science, The Hebrew University of Jerusalem, Jerusalem, Israel

Communicated by M.S. Paterson
Received January 1983
Revised September 1984

Abstract. The power of probabilistic linear decision trees is examined. It is shown that the standard arguments used to prove lower bounds on deterministic linear decision trees apply to probabilistic linear decision trees as well. Examples are next given of problems which randomization helps solving.

1. Introduction

Randomizing algorithms have drawn much attention in the last years. Problems that seem hard to solve when conventional algorithms are used, are easily solved by probabilistic methods.

Many algorithms are usefully represented by the *decision tree* model. This model captures the control flow of the algorithm, that is the decisions taken during execution. One represents the tests made and the branch statements executed, and ignores the other computations. Randomizing steps can be added to this model, as tests whose outcome depend on independent probabilistic experiments, that is 'coin flipping' tests.

Manber and Tompa [6] considered nondeterministic and probabilistic decision trees. These are decision trees that have guessing nodes, aside from comparison nodes. At a guessing node, a choice is made between two alternatives, nondeterministically for nondeterministic decision trees, or with equal probability for probabilistic decision trees.

Manber and Tompa give nonlinear $\Omega(n \log n)$ lower bounds on the depth of nondeterministic decision trees needed to solve several problems. Interestingly, their upper bounds show that each of these problems can be solved either with $O(n \log n)$ comparisons and no guesses, or with $O(n)$ comparisons and $O(n \log n)$ guesses. Thus, the nonlinear lower bounds do not apply either to comparisons or to guesses alone, but only to their sum.

These lower bounds apply to the weaker, probabilistic model as well, with cost being defined as the length of the longest path in the decision tree.

0304-3975/85/$3.30

Their paper leaves open many interesting questions. In particular, can we obtain nonlinear lower bounds on the cost of probabilistic decision trees (using the usual, 'expected time' measure) when only comparisons are counted? Can we extend lower bounds known for deterministic decision trees to probabilistic decision trees? And, most important of all, can we exhibit problems where randomization yields a provable speedup?

This paper gives a partial answer to these problems. We show that known methods for proving lower bounds on deterministic linear decision trees apply to probabilistic linear decision trees as well. We consider two methods: the 'region counting' argument of Dobkin and Lipton [1] and the 'face counting' argument of Yao and Rivest [10] (see also [8] for an exposition of their methods). Lower bounds obtained by these methods for deterministic linear decision trees turn out to be valid, within a constant factor, for probabilistic linear decision trees as well. This holds true, even if only comparisons are counted. Thus, decision problems that are known to require $\Omega(n \log n)$ comparisons in the deterministic case, are shown to require $\Omega(n \log n)$ comparisons on the average, in a probabilistic algorithm. This includes all the problems for which nonlinear lower bounds are proven in [6].

These results indicate an interesting distinction between nondeterministic and probabilistic decision trees: For the problems considered in [6], the use of randomization does not decrease the number of comparisons needed, whereas comparisons can be traded off for guesses in the nondeterministic case.

Nevertheless, randomization can reduce the complexity of linear decision problems. We show that it is possible to build a family of problems P_n such that P_n can be solved by a probabilistic linear decision tree of cost 3^n, but cannot be solved by a deterministic linear decision tree of cost less than 4^n. The problem P_n is defined by a formula with n alternations of existential and universal quantifiers.

The proof of this gap involves two interesting methods. The proof of the lower bound for deterministic complexity uses an invariance argument: If a problem is invariant under a set of transformations, then an optimal solution to this problem uses only tests which are invariant under the 'dual' set of transformations. The upper bound argument for probabilistic complexity uses a very simple form of randomization: when the solution to a problem will be obtained by solving either the first or the second of two subproblems, then one betters the average running time by randomly choosing the order in which these subproblems are attacked.

2. Definitions

Let $\mathbb{R}^n$ be the space of real n-tuples, and let $P \subset \mathbb{R}^n$. The *decision problem* $D(P)$ is to determine for any input $x \in \mathbb{R}^n$ whether $x \in P$. An algorithm for this decision problem is represented by a *decision tree*. This is a labeled binary tree. The internal nodes are either

(1) *test nodes* which are labeled with binary tests; the two emanating edges are labeled by the possible outcomes of that test; or

(2) *randomizing nodes* which are unlabeled.

The leaves are labeled either *accept* or *reject.*

For each input $x \in \mathbb{R}^n$ the algorithm proceeds by moving down the tree. At test nodes the test is performed on x, and one of the two branches is taken according to the outcome of the test. At randomizing nodes one of the two branches is taken randomly, with probability $\frac{1}{2}$.

We define the *acceptance (rejection) probability* $\pi_a(x)$ ($\pi_r(x)$) of an input x to be the probability that x reaches an accepting (rejecting) leaf. The decision tree *solves the problem $D(P)$ with threshold α* if:

(1) if $x \in P$, then x is accepted with probability $\geq \alpha$, and

(2) if $x \notin P$, then x is accepted with probability zero (i.e., reaches only rejecting leaves).

The *time* $\tau(x)$ required for an input x is the expected number of tests on a path followed by x. (We do not charge for randomizing nodes.) The *acceptance (rejection) time* $\tau_a(x)$ ($\tau_r(x)$) required for an input x is the expected number of tests on a path followed by x, given that the path leads to an accepting (rejecting) leaf. The acceptance (rejection) time is defined to be equal to zero if x never reaches an accepting (rejecting) leaf.

Note that

$$\tau(x) = \pi_a(x)\tau_a(x) + \pi_r(x)\tau_r(x). \tag{2.1}$$

The *cost* $c(T)$ (*acceptance cost* $c_a(T)$, *rejection cost* $c_r(T)$) of a decision tree T is defined to be the maximum, over all possible inputs x, of the time (acceptance time, rejection time) required for x.

If T solves the problem $D(P)$ with threshold α, then $\pi_a(x) \geq \alpha$ for each $x \in P$ and $\pi_r(x) = 1$ for each $x \notin P$. This implies, by (2.1), that $c(T) \geq (1/\alpha)c_a(T)$ and $c(T) \geq c_r(T)$. Thus, lower bounds on acceptance (rejection) cost imply lower bounds on cost.

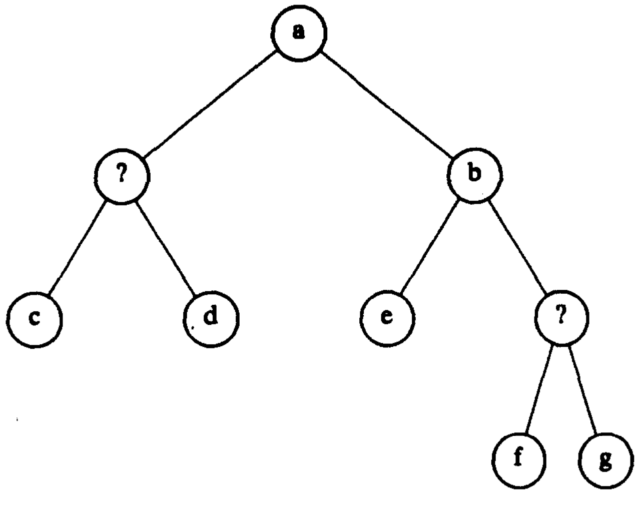
A decision tree is *deterministic* if it does not contain randomizing nodes. In that case the previous definitions agree with the usual ones for deterministic decision trees: each input follows a unique path; the decision tree solves the problem $D(P)$ iff each input x reaches an accepting leaf iff $x \in P$; and the cost of a decision tree is equal to the depth of the tree. Note that the acceptance cost and rejection cost of a minimal deterministic decision tree are equal to its cost. Indeed, if T is minimal, then each nontrivial subtree of T contains both accepting leaves and rejecting leaves (otherwise the subtree can be replaced by a unique leaf). In particular, T has both accepting leaves and rejecting leaves at maximum depth.

3. Main theorem

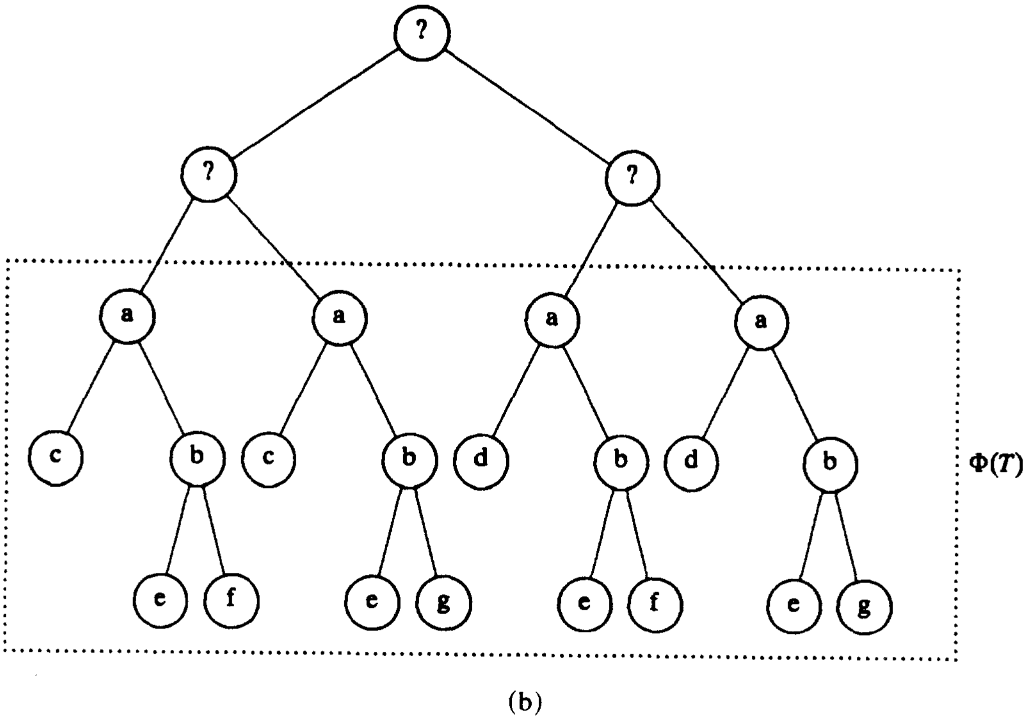
Theorem 3.1. *Let T be a probabilistic decision tree that solves a decision problem $D(P)$ with threshold α. Let $x_1, \ldots, x_m$ be inputs from P that are accepted in time $\leq t$. Then for each $\lambda > 1$ there exists a deterministic decision tree T' with the following properties:*

(1) *All the tests occurring in T' are tests from T.*
(2) *T' accepts at least a fraction $\alpha(1-1/\lambda)$ of the inputs $x_1, \ldots, x_m$ in time $\leqslant \lambda t$.*
(3) *T' accepts only inputs belonging to P.*

Proof. Assume that k randomizing nodes occur in T. Let $\Phi(T)$ be the set of 2^k deterministic decision trees obtained from T by replacing in all possible ways each of the k random choices by a deterministic choice. Consider the decision procedure whereby a tree is randomly chosen from $\Phi(T)$, next applied to the input. This procedure is represented by a probabilistic decision tree $\hat{T}$ consisting of k levels of randomizing nodes, followed by the 2^k trees of $\Phi(T)$. The construction is illustrated in Fig. 1.



(a)



(b)

Fig. 1. (a) Tree T. (b) Tree $\hat{T}$.

It is easily seen that the decision tree $\hat{T}$ defines the same probability distribution as T. For each input x the acceptance probability of x in T is equal to the acceptance probability of x in $\hat{T}$. This, in turn, is equal to the fraction of trees in $\Phi(T)$ that accept x. The acceptance (rejection) time of x in T is equal to the acceptance (rejection) time of x in $\hat{T}$. This is equal to the average length of the path followed by x in those trees in $\Phi(T)$ that accept (reject) x.

If $x \notin P$, then x does not reach an accepting leaf in T, and therefore does not reach an accepting leaf in any tree from $\Phi(T)$. Thus, trees in $\Phi(T)$ accept only inputs from P.

Each input x_j is accepted by a fraction α of the trees in $\Phi(T)$, and the average length of the path followed in these trees by x_j is at most t. Thus, a fraction of $(1-1/\lambda)$ of the trees in $\Phi(T)$ that accept x_j, accept it in no more than λt steps. Each input x_j is accepted by at least a fraction $\alpha(1-1/\lambda)$ of the trees in $\Phi(T)$ in no more than λt steps. This implies that there exists a tree in $\Phi(T)$ that accepts a fraction $\alpha(1-1/\lambda)$ of the inputs $x_1, \ldots, x_m$ in no more than λt steps. □

4. Region counting

We restrict ourselves now to *linear decision trees*, i.e., decision trees using only tests of the form $f(x)\ R\ 0$, where f is an affine functional and R is one of the relations $<$, $>$, $\leqslant$, or $\geqslant$. The set of inputs reaching a leaf of a linear decision tree is convex. This implies immediately the following theorem (see [1, Theorem 2], [6, Theorem 1], and [8, Corollary 3.2]).

Theorem 4.1. *Let $P \subset \mathbb{R}^n$, and let $x_1, \ldots, x_m \in P$ satisfy the condition that for any pair x_i, x_j with $i \neq j$ the segment $\overline{x_i x_j}$ contains a point $y_{ij} \notin P$. Then if T is a linear decision tree solving $D(P)$, each accepting node in T is reached by at most one input from the set $x_1, \ldots, x_m$. In particular, T contains at least m accepting nodes, and has depth at least* $\log_2 m$.

We can use Theorem 3.1 to extend this result to probabilistic decision trees.

Theorem 4.2. *Assume that the conditions of Theorem* 4.1 *are satisfied. Then any probabilistic linear decision tree solving $D(P)$ with threshold α has acceptance cost greater than* $(1/\lambda)\log_2 m + (1/\lambda)\log_2 \alpha(1-1/\lambda)$, *for any* $\lambda > 1$.

Proof. If T is a probabilistic linear decision tree with acceptance cost t, then one can build a deterministic linear decision tree $\hat{T}$ that accepts at least $m\alpha(1-1/\lambda)$ of the inputs $x_1, \ldots, x_m$ in time $\leqslant \lambda t$ and accepts none of the inputs y_{ij}. No leaf of $\hat{T}$ is reached by two distinct inputs x_i. It follows that $\lambda t \geqslant \log_2(\alpha(1-1/\lambda)m)$. □

Corollary 4.3. *Let $\alpha > 0$ be a fixed threshold. Then any probabilistic linear decision tree that solves one of the following problems has cost $\Omega(n \log n)$.*

(1) *Element uniqueness* [1]: *Decide whether* $x_1, \ldots, x_n$ *are distinct.*

(2) *Set disjointness* [7]: *Given two sets* S_1 *and* S_2 *of n numbers each, decide whether* S_1 *and* S_2 *are disjoint.*

(3) *Set equality* [7]: *Given two sets* S_1 *and* S_2 *of n numbers each, decide whether they are equal.*

(4) ε*-distance* [3]: *Decide whether the distance between every two distinct numbers in* $x_1, \ldots, x_n$ *is larger than* ε.

(5) *Set maximality in plane* [5]: *Given* $2n$ *inputs* $x_1, y_1, \ldots, x_n, y_n$, *decide whether the n points in the plane with these coordinates form a maximal set (the set is maximal if for no* $i \neq j$ *both* $x_i \leq x_j$ *and* $y_i \leq y_j$*).*

Proof. For each of these problems it is possible to exhibit $n!$ distinct inputs that fulfil the conditions of Theorem 4.1. □

5. Face counting

We assume now that P is a polyhedral convex set in $\mathbb{R}^n$ (this restriction can be relaxed, see [8, Section 4]). Let T be a linear decision tree that solves the problem $D(P)$. To each leaf e of T is associated the polyhedral convex set $S(e)$ of inputs that reach e. This set is defined by the linear inequalities labeling the edges on the path leading to e. We have

$$P = \bigcup_{e \text{ accepting leaf of } T} S(e).$$

If T is deterministic, the union is disjoint. This decomposition of P induces a decomposition of the faces of P: Each k-dimensional face of P is the union of faces of the sets $S(e)$ of dimension k or less. In particular, each k-dimensional face of P contains a k-dimensional face of some set $S(e)$. A k-dimensional face of $S(e)$ is obtained by choosing $n-k$ of the inequalities defining $S(e)$, that is $n-k$ of the relations labeling edges on the path to e, and replacing the inequalities by equalities. There are at most 2^d leaves, and for each leaf at most $\binom{d}{n-k}$ such possible choices, if d is the depth of T. We obtain therefore the following theorem.

Theorem 5.1. *Let* $f_k(P)$ *be the number of k-dimensional faces of a polyhedral convex set P. If T is a linear decision tree of depth d that solves* $D(P)$, *then*

$$2^d \binom{d}{n-k} \geq f_k(P)$$

for any $0 \leq k \leq n$.

This theorem is proven in [10] (for deterministic linear decision trees) using an adversary argument.

Once again we can use Theorem 3.1 to extend this lower bound to the cost of probabilistic decision trees. We assume that P is closed so that it contains all its faces. Let T be a probabilistic linear decision tree that solves $D(P)$ with threshold α and acceptance cost t. Let Ψ be the family of linear equalities $(f\cdot x)=a$ corresponding to linear tests occurring in T. We pick from each k-dimensional face F of P an input $x_F\in F$ with the property that x does not fulfil any equality from Ψ unless this equality is identically satisfied on F. According to Theorem 3.1 there exists a deterministic linear decision tree $\hat{T}$ that accepts at least a fraction of $\alpha(1-1/\lambda)$ of these inputs in time λt, and accepts only inputs from P. Let e be the leaf reached in $\hat{T}$ by x_F, and let $S(e)$ be the set of inputs reaching e in $\hat{T}$. Then $S(e)\subset P$ and x_F is contained therefore in a face $\hat{F}$ of $S(e)$ such that $\hat{F}\subset F$. The choice of x_F ensures that each of the equalities defining $\hat{F}$ is identically satisfied on F. It follows that $\dim\hat{F}\geqslant\dim F$, so that $\dim\hat{F}=k$. The sets $S(e)$ associated with accepting leaves at depth λt or less in $\hat{T}$ have therefore at least $\alpha(1-1/\lambda)f_k(P)$ distinct k-dimensional faces. We have proven the following theorem.

Theorem 5.2. *Let $P\subset\mathbb{R}^n$ be a closed polyhedral convex set with $f_k(P)$ k-dimensional faces, $k=0,\ldots,n$. Let T be a probabilistic linear decision tree that solves $D(P)$ with threshold α and acceptance cost t. Then*

$$2^{\lambda t}\binom{\lfloor\lambda t\rfloor}{k}\geqslant\alpha\left(1-\frac{1}{\lambda}\right)f_{n-k}(P)$$

for any $\lambda>1$ and any $0\leqslant k\leqslant n$.

Corollary 5.3. *Let $P=\{x\in\mathbb{R}^n\colon x_1\geqslant x_i,\ i=2,\ldots,n\}$. Any probabilistic linear decision tree that solves $D(P)$ with threshold α (i.e., decides whether $x_1=\max_i x_i$) has acceptance cost at least $n-1$.*

Proof. P has a 1-dimensional face defined by the equalities $x_1=\cdots=x_n$. Thus, if t is the acceptance cost of a tree which solves $D(P)$, then $\binom{\lfloor\lambda t\rfloor}{n-1}>0$ for any $\lambda>1$, so that $t\geqslant n-1$. □

6. Invariance argument

We wish to introduce in this section a new principle, which is familiar from other fields in mathematics. One way of showing that some problem has a simple structure is to show that it is invariant under a large family of transformations. If this is the case then we can restrict the search for an efficient solution to algorithms that are invariant under the 'dual' family of transformations. For example, if x_n does not occur in the definition of $P\subset\mathbb{R}^n$, a fact that can be expressed by saying that P is invariant under translations along the x_n axis, then comparisons involving x_n do not help in solving $D(P)$. This invariance principle will be later used to build a family of problems that can be solved faster using randomization.

Let V_1 and V_2 be linear spaces over $\mathbb{R}$. $G: V_1 \to V_2$ is an *affine transformation* if it is of the form $G(\boldsymbol{v}) = H(\boldsymbol{v}) + \boldsymbol{w}$, where $H: V_1 \to V_2$ is linear, and $\boldsymbol{w} \in V_2$. We denote by V^a the linear space of affine functionals, e.g., the space of affine transformations from V to $\mathbb{R}$. If F is an affine transformation in V we denote by F^* the dual linear transformation in V^a defined by the identity $(F^*f)(\boldsymbol{v}) = f(F\boldsymbol{v})$. The affine functional f is *sign invariant* under the affine transformation F if $f(F\boldsymbol{v}) \geq 0 \Leftrightarrow f(\boldsymbol{v}) \geq 0$. Thus f is sign invariant under F iff F maps the hyperplane defined by the equation $f(\boldsymbol{v}) = 0$, and each of the two halfspaces defined by it, into themselves.

We recall the following result, which is due to Farkas [2].

Lemma 6.1. *Let V be a finite-dimensional vector space, f and g be two affine functionals over V. The following two assertions are equivalent*:

(1) $\forall \boldsymbol{v} f(\boldsymbol{v}) \geq 0 \Leftrightarrow g(\boldsymbol{v}) \geq 0$.

(2) $\exists \lambda > 0$ *such that* $f = \lambda g$.

The affine functional f is sign invariant under the affine transformation F iff $(F^*f)(\boldsymbol{v}) \geq 0 \Leftrightarrow f(\boldsymbol{v}) \geq 0$. The last condition is fulfilled, according to Lemma 6.1, iff there exists a $\lambda > 0$ such that $F^*f = \lambda f$. Thus the affine functional f is sign invariant under the affine transformation F iff f is a *positive eigenvector* of F^*, i.e., an eigenvector of F^* with positive eigenvalue.

Let U be a set of functionals on $\mathbb{R}^n$. The set P is of *type U* if it can be obtained from sets of the form $\{\boldsymbol{x} : f(\boldsymbol{x}) > 0\}$, where $f \in U$, using complements, finite unions, and finite intersections (P is the truth set of some Boolean combination of assertions of the form $f(\boldsymbol{x}) > 0$). In particular, P is of *affine type* if P is of type U, where U is the set of affine functionals in $\mathbb{R}^n$, and P is of *linear type* if U is the set of linear functionals on $\mathbb{R}^n$. Note that P is of affine type iff p is the finite union of polyhedral convex sets, and of linear type iff it is the finite union of polyhedral convex cones.

Let P be a set of type U. If each functional $f \in U$ is sign invariant under the affine transformation F, then F maps the set P onto itself. Let U^* be the set of such affine transformations. Our claim is that in order to solve the problem $D(P)$ it is sufficient to use comparisons involving affine functionals that are sign invariant under U^*. In order to prove this claim we must first characterize these functionals. This is done in the next theorem and the following corollary.

Theorem 6.2. *Let U be a family of nonzero vectors in a finite-dimensional linear space V, and for each* $\boldsymbol{u} \in U$ *let* $S(\boldsymbol{u})$ *be a set of nonzero scalars. Let* U^* *be the set of linear transformations F on V with the property that each vector* $\boldsymbol{u} \in U$ *is an eigenvector of F with eigenvalue in* $S(\boldsymbol{u})$; *let* U^{**} *be the set of vectors* $\boldsymbol{v}$ *that are eigenvectors of each transformation in* U^*. *Then there exists a finite family of subspaces* $V_1, \ldots, V_{r+1}$ *of V such that*

(1) $V = V_1 \oplus \cdots \oplus V_{r+1}$.

(2) $F \in U^*$ *iff* $\exists \lambda_1, \ldots, \lambda_r$ *such that* $\lambda_i \in \bigcap_{\boldsymbol{u} \in V_i \cap U} S(\boldsymbol{u})$ *and* $\boldsymbol{v} \in V_i \Rightarrow F\boldsymbol{v} = \lambda_i \boldsymbol{v}$, $i = 1, \ldots, r$.

(3) $U^{**}=\bigcup_{i=1}^{r} V_i$, *and an eigenvector from* V_i *can only be associated with eigenvalues from* $\bigcap_{u\in V_i\cap U} S(u)$.

Proof. For $v\in U^{**}$, let G_v be the mapping from U^* to $\mathbb{R}$ defined by

$$G_v(F)=\lambda, \quad \text{where } F(v)=\lambda v.$$

For each $v\in U^{**}$, the set $\{w\in U^{**}: G_w=G_v\}\cup\{\mathbf{0}\}$ is a linear subspace of V. Two such distinct sets intersect only in $\mathbf{0}$. Since V is finite-dimensional, there exists a finite number of such subspaces, say $V_1,\ldots,V_r$, so that $U^{**}\subset V_1\oplus\cdots\oplus V_r$.

Let $v\in U^{**}$. Then $v=v_1+\cdots+v_r$, where $v_i\in V_i$. Assume w.l.o.g. that $v_1\neq\mathbf{0}$. Let $F\in U^*$. Then $F(v_i)=\lambda_i v_i$, $i=1,\ldots,r$, and $F(v)=\lambda v$. Thus

$$\sum_{i=1}^{r}\lambda v_i=F(v)=\sum_{i=1}^{r}\lambda_i v_i.$$

It follows that $\lambda=\lambda_1$. As this holds true for any $F\in U^*$, $G_v=G_{v_1}$, so that $v\in V_1$. We have shown that $U^{**}\cup\{\mathbf{0}\}=\bigcup_{i=1}^{r} V_i$.

If $V_i\cap U=\mathbf{0}$, then transformations $F\in U^*$ can be defined so that their restriction on V_i be an arbitrary linear mapping. Thus

$$V_i\cap U=\mathbf{0} \;\Rightarrow\; V_i\cap U^{**}=\mathbf{0}.$$

This implies that $V_i\cap U\neq\mathbf{0}$, $i=1,\ldots,r$. As all nonzero vectors in V_i are associated with the same eigenvalues, the set of eigenvalues associated with V_i is at most $\bigcap_{u\in U\cap V_i} S(u)$. This proves (3).

The 'if' part of (2) follows from the fact that $U\subset U^{**}=\bigcup_{i=1}^{r} V_i$; the 'only if' part follows from the definition of V_i.

Take V_{r+1} to be the orthogonal complement to $V_1\oplus\cdots\oplus V_r$ in V; then (1) is satisfied. □

Corollary 6.3. *Let* V *be a finite-dimensional vector space and let* V^{a} *be the linear space of affine functionals over* V. *Let* U *be a subset of* V^{a}. *Let* U^* *be the set of affine transformations under which each functional in* U *is sign invariant, and let* U^{**} *be the set of affine functionals that are sign invariant under each transformation in* U^*. *Then there exists a direct sum decomposition* $V^{\mathrm{a}}=V_1\oplus\cdots\oplus V_{r+1}$ *such that*:

(1) *If* $F\in U^*$, *then* $\exists\lambda_2,\ldots,\lambda_r>0$ *such that* $F^*f=f$ *if* $f\in V_1$ *and* $F^*f=\lambda_i f$ *whenever* $f\in V_i$, $i=2,\ldots,r$; *conversely any linear transformation in* V^{a} *that fulfils these two conditions is the dual of an affine transformation in* U^*.

(2) $U^{**}=\bigcup_{i=1}^{r} V_i$.

Proof. Note that a linear transformation in V^{a} is the dual of an affine transformation in V iff it maps the affine functional $\iota(x)\equiv 1$ onto itself. Thus, the set of linear transformations on V^{a} which are dual to affine transformations in U^* are precisely all those linear transformations in V^{a} that map each $f\in U$ to a positive multiple of itself, and map ι to itself. The claim now follows from the previous theorem. □

The next theorem shows that there exists a minimal cost deterministic linear decision tree for $D(P)$ that uses only tests which are sign invariant under those affine transformations that preserve P. The proof uses the decomposition for U^{**} that was obtained in the last corollary: we replace each test on the tree by a test derived from an affine functional that acts only on one subspace V_i. In order to show that the derived tree is still correct, we show that each input vector x can be 'streched' separately in each subspace V_i by an affine transformation that preserves P, thus obtaining a vector y, so that the answer of the new tree on x is identical to the answer of the old tree on y.

Theorem 6.4. *Let P be a set of type U, and let T be a probabilistic linear decision tree that solves $D(P)$ with threshold α. Let U^* be the set of affine transformations in $\mathbb{R}^n$ under which each affine functional in U is sign invariant, and let U^{**} be the set of affine functionals that are sign invariant under each transformation in U^*. Then the tests in T can be replaced by tests from U^{**}, so that the resulting tree $\hat{T}$ solves $D(P)$ with threshold α, and cost (accepting cost, rejecting cost) no greater than the cost (accepting cost, rejecting cost) of T. If T is deterministic, then $\hat{T}$ is deterministic too.*

Proof. Let $V_1 \oplus \cdots \oplus V_{r+1}$ be a direct sum decomposition of V^a that fulfils the condition of Corollary 6.3. Let $f_1, \ldots, f_k$ be the affine functionals used in the tests occurring in T. Each functional has a decomposition of the form $f_i = \sum_{j=1}^{r+1} g_j^i$, where $g_j^i \in V_j$. Let j_i be the index of the first nonzero term in the decomposition of f_i ($=\infty$ if no such term exists), so that $f_i = \sum_{j=j_i}^{r+1} g_j^i$.

Let $\hat{T}$ be the decision tree obtained from T by replacing each test $f_i(x)\ R\ 0$ by the test $\hat{f}_i(x)\ R\ 0$, where $\hat{f}_i = g_{j_i}^i$ if $i_j \leq r$, or 0 otherwise. $\hat{T}$ tests only affine functionals from U^{**}, and is deterministic if T is so. We claim that $\hat{T}$ solves $D(P)$ at a cost no greater then the cost of T.

Let $x \in \mathbb{R}^n$ be a fixed input. Let $\delta_1, \ldots, \delta_{r+1}$ be a sequence of numbers fulfilling the following conditions:

(1) $\delta_{r+1} = 0$.

(2) δ_j, $j = r, r-1, \ldots, 1$ fulfils the inequality $\delta_j |g_j^i(x)| > \sum_{k>j} \delta_k |g_k^i(x)|$, for any i, whenever $g_j^i(x) > 0$.

Let $\lambda_j = \delta_j / \delta_1$. We then have:

(1) $\lambda_{r+1} = 0$.

(2) $\lambda_1 = 1$.

(3) $\sum_i \lambda_i g_i^i(x) = 0$ only if $i_j > r$.

(4) If $\sum \lambda_j g_j^i(x) \neq 0$, then $\operatorname{sign}(\sum \lambda_j g_j^i(x)) = \operatorname{sign}(g_{j_i}^i(x))$.

Let G^* be the linear transformation on $(\mathbb{R}^n)^a$ defined by the conditions

$$G^* f = \lambda_i f \quad \text{if } f \in V_i,\ i = 1, \ldots, r+1.$$

According to the previous theorem, G^* is the dual of an affine transformation G in $\mathbb{R}^n$, and each affine functional in U is sign invariant under G. It follows that $x \in P$ iff $G(x) \in P$. On the other hand for each affine functional f_i occurring in T

we have

$$f_i(Gx) = (G^*f_i)(x) = \sum_{j=i_j}^{r} \lambda_j g_j^i(x).$$

It follows that $\text{sign}(f_i(Gx)) = \text{sign}(\hat{f}_i(x))$. Thus x follows the same path in $\hat{T}$ as Gx follows in T, and x is accepted (rejected) by $\hat{T}$ in time t iff Gx is accepted (rejected) by T in time t. As T solves $D(P)$, Gx is accepted by T iff $Gx \in P$, and $Gx \in P$ iff $x \in P$. This concludes the proof. □

Corollary 6.5. *Let P be of type U where U is a linear subspace of affine functionals. Then there exists a minimal cost probabilistic (deterministic) linear decision tree T that uses only tests from U. In particular, if P is defined by linear, homogeneous inequalities, then T uses only linear, homogeneous inequalities.*

Proof. If U is a linear subspace, then $U^{**} = U$. □

Corollary 6.6 ([9]). *Let P be a set defined by inequalities of the form $x_i > x_j$ (P is defined in terms of the relative order of $x_1, \ldots, x_n$). There exists a minimal cost probabilistic (deterministic) linear decision tree that uses only tests of the form $\sum \alpha_i x_i \mathrel{R} 0$, where $\sum \alpha_i = 0$.*

Proof. The linear closure of the functionals $f(x) = x_i - x_j$ consists exactly of all the functionals of the form $f(x) = \sum \alpha_i x_i$, where $\sum \alpha_i = 0$. □

7. Composite problems

Let P_1 and P_2 be subsets of $\mathbb{R}^m$ and $\mathbb{R}^n$ respectively. The subset $P_1 \otimes P_2$ of $\mathbb{R}^{n+m}$ is defined to be the set $\{\langle x, y\rangle : x \in P_1 \text{ and } y \in P_2\}$. We define similarly $P_1 \oplus P_2 = \{\langle x, y\rangle : x \in P_1 \text{ or } y \in P_2\}$. Note that if P_1 and P_2 are of affine (linear) type, then both $P_1 \otimes P_2$ and $P_1 \oplus P_2$ are of affine (linear) type.

Define $\text{dc}(P)$ to be the minimum cost of a deterministic linear decision tree that solves $D(P)$. We have the following theorem.

Theorem 7.1. *Let P_i be of linear type. Then*

$$\text{dc}(P_1 \oplus P_2) = \text{dc}(P_1 \otimes P_2) = \text{dc}(P_1) + \text{dc}(P_2).$$

Proof. Note that $P_1 \oplus P_2 = (P_1^c \otimes P_2^c)^c$, where P^c is the complement of P, and that $\text{dc}(P) = \text{dc}(P^c)$. Thus, the first equality follows from the second one.

The set $P_1 \otimes P_2$ is invariant under linear transformations of the form $\langle x, y\rangle \to \langle \alpha x, \beta y\rangle$, where $\alpha, \beta > 0$. Let $f(x, y) = (u \cdot x) + (v \cdot y) + a$ be an affine functional on $\mathbb{R}^{m+n}$, and assume that $u \neq 0$, $v \neq 0$. Pick x and y so that $(u \cdot x) > 0$ and $(v \cdot y) < 0$.

Then it is clear that for $\alpha \gg \beta \gg 0$, $f(\alpha \boldsymbol{x}, \beta \boldsymbol{y}) = \alpha(\boldsymbol{u} \cdot \boldsymbol{x}) + \beta(\boldsymbol{v} \cdot \boldsymbol{y}) + a > 0$, whereas, for $0 \ll \alpha \ll \beta$, $f(\alpha \boldsymbol{x}, \beta \boldsymbol{y}) < 0$. It follows that if f is sign invariant under these transformations, then either $\boldsymbol{u} = \boldsymbol{0}$ or $\boldsymbol{v} = \boldsymbol{0}$. A similar reasoning shows that if f is sign invariant, and not constant, then $a = 0$. Thus f is sign invariant under these transformations iff it is linear, and either depends only on $\boldsymbol{x}$ or depends only on $\boldsymbol{y}$.

The proof can now be completed using an adversary argument, which is, in fact, a game theoretic argument. We associate with each problem $D(P)$ a game $\boldsymbol{G}$ played between *decider* and *adversary*. Alternately, the decider chooses to perform a linear test on the input and the adversary chooses an answer to this test which is consistent with its previous answers. The game ends if the answers to the successive tests determine membership in P. The deterministic complexity of $D(P)$ is c iff c is the number of moves the adversary can force in the game.

Now let $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$ be the games corresponding to the problems $D(P_1)$ and $D(P_2)$, and let c_1 and c_2 be their respective deterministic complexity. In order to solve $D(P_1 \otimes P_2)$ for the input $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ we have to determine whether $x \in P_1$ and $y \in P_2$, that is solve $D(P_1)$ and $D(P_2)$. According to Theorem 6.4 we can restrict our attention in solving the problem $D(P_1 \otimes P_2)$ to unmixed comparisons involving only $\boldsymbol{x}$ or only $\boldsymbol{y}$. The deterministic complexity of the problem $D(P_1 \otimes P_2)$ is therefore equal to the number of moves the adversary can force in the game $\boldsymbol{G}_1 \times \boldsymbol{G}_2$ which is the Cartesian product of $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$, where moves are either moves of $\boldsymbol{G}_1$ or moves of $\boldsymbol{G}_2$, and this is equal to $c_1 + c_2$ [4]. □

The results of the previous section were used in the last proof to show that mixed comparisons involving both x's and y's do not help. This claim is trivial for decision trees using only simple comparisons (we can assume that $x_i < y_j$, for each i and j). In that case the last result can be proved without using the heavy machinery of the previous section.

The last result does not extend to sets defined by affine inequalities. In [8] we give an example of two sets P_1, P_2 in $\mathbb{R}$ such that $\mathrm{dc}(P_1 \otimes P_2) < \mathrm{dc}(P_1) + \mathrm{dc}(P_2)$. Thus, in general, it might be easier to solve two independent problems together rather than solving each one separately.

The last result is not valid either for probabilistic linear decision trees. We have the following theorem.

Theorem 7.2. *Let P_1 and P_2 be two sets such that $D(P_i)$ can be solved by a probabilistic linear decision tree T_i with threshold α, acceptance cost* ac *and rejection cost* rc. *Then*:

(1) *$D(P_1 \otimes P_2)$ can be solved by a probabilistic linear decision tree with threshold α^2, acceptance cost* 2ac *and rejection cost* $\mathrm{rc} + \frac{1}{2}\mathrm{ac}$.

(2) *$D(P_1 \oplus P_2)$ can be solved by a probabilistic linear decision tree with threshold α, acceptance cost* $\mathrm{ac} + \frac{1}{2}\mathrm{rc}$ *and rejection cost* 2rc.

Proof. (1) Membership in $P_1 \otimes P_2$ can be decided by performing first the tests in T_1 and, if an accepting node has been reached, performing next the tests from T_2.

The decision tree $\hat{T}_1$ thus defined solves $D(P_1 \otimes P_2)$ with threshold α^2, accepts each input from the set in no more than 2ac steps on the average, rejects any input whose first component is not in P_1 in rc steps on the average, and rejects any input with a first component in P_1 and a second component not in P_2 in no more than ac+rc steps on the average. Let $\hat{T}_2$ be the decision tree obtained by reversing the order of T_1 and T_2. Then $\hat{T}_2$ accepts each input in no more than 2ac steps, rejects any input whose second component is not in P_2 in no more than rc steps, and rejects the remaining inputs in no more than ac+rc steps on the average. Finally, let $\hat{T}$ be the decision tree consisting of a randomizing node at the root, followed by $\hat{T}_1$ and $\hat{T}_2$ as left and right subtrees. Then $\hat{T}$ accepts any input in $P_1 \otimes P_2$ in no more than 2ac steps on the average and rejects any input not in $P_1 \otimes P_2$ in no more than $\frac{1}{2}((\mathrm{ac}+\mathrm{rc})+(\mathrm{rc}))=\mathrm{rc}+\frac{1}{2}\mathrm{ac}$ steps on the average.

The proof of (2) is similar. □

Let $[\varepsilon_1, \ldots, \varepsilon_n]$ denote the number with binary representation $\varepsilon_1, \ldots, \varepsilon_n$ ($\varepsilon_i \in \{0, 1\}$). The set P_n is defined in $\mathbb{R}^{4^n}$ by a sequence of $2n$ alternating quantifiers:

$$P_n = \{x : \forall \alpha_n \exists \beta_n, \ldots, \forall \alpha_1 \exists \beta_1 \; x_{[\alpha_1,\ldots,\alpha_n,\beta_1,\ldots,\beta_n]} = 0\} (\alpha_i, \beta_i \in \{0, 1\}).$$

We have the following corollary.

Corollary 7.3. (1) *Any deterministic linear decision tree solving $D(P_n)$ has cost at least 4^n, and*

(2) *P_n can be solved with threshold 1 by a probabilistic linear decision tree with cost $\leqslant 3^n$.*

Proof. Note that $P_{n+1} = (P_n \oplus P_n) \otimes (P_n \oplus P_n)$, where $P_0 = \{0\}$. The deterministic complexity of P_0, $\mathrm{dc}(P_0) = 1$, and Theorem 7.1 implies that $\mathrm{dc}(P_{n+1}) = 4\mathrm{dc}(P_n)$. This proves the first half of the claim. The second half is proven by induction, using Theorem 7.2. P_0 can be solved by a probabilistic linear decision tree with threshold 1 and cost 1. Assume the claim is valid for P_n. Then $P_n \oplus P_n$ can be solved by a probabilistic linear decision tree with threshold 1, acceptance cost $\leqslant 1.5 \times 3^n$ and rejection cost $\leqslant 2 \times 3^n$, and P_{n+1} can be solved by a probabilistic linear decision tree with threshold 1, acceptance cost $\leqslant 3^{n+1}$ and rejection cost $\leqslant 2 \times 3^n + \frac{3}{4} \times 3^n < 3^{n+1}$. □

A similar result can be obtained even if randomizing steps are accounted for. The construction of Theorem 7.2 shows that if P can be solved by a probabilistic linear decision tree with threshold 1 and cost $k > 2$, then $(P \oplus P) \otimes (P \oplus P)$ can be solved by a probabilistic linear decision tree with threshold 1 and cost $\leqslant 3k + 2$, where the term of 2 accounts for the randomizing steps. Iterating this construction one can build a sequence P_n of problems such that the deterministic complexity of P_n is $4^n k$, yet P_n can be solved by a probabilistic linear decision tree with threshold 1 and cost $\leqslant 3^n(k+1)$.

8. Concluding remarks

The definition of deterministic decision trees is symmetric with respect to acceptance and rejection. In particular, the problem of deciding membership in P is as hard as the problem of deciding membership in the complement P^c. We considered in this paper one-sided probabilistic decision trees, where rejection may be wrong, but not acceptance. The results, therefore, are not symmetric any more. It is quite easy to build examples of problems such that the probabilistic complexity of $D(P^c)$ is lower than the probabilistic complexity of $D(P)$. We do not know how large the gap can be. In particular, are $\Omega(n \log n)$ lower bounds valid for the complements of each of the problems listed in Corollary 4.3?

We gave in Corollary 7.3 an ad hoc example of a linear decision problem that can be solved faster using randomization. It would be of interest to exhibit such speedup with respect to a 'natural' problem.

The example we built uses randomization only in a restricted sense, as a threshold of 1 is used: An input always reaches a leaf with the correct label. When the threshold is below one we are allowed (not too frequent) mistakes with respect to accepted inputs. Do there exist problems that exhibit a tradeoff between accepting cost and acceptance threshold?

Acknowledgment

The present version of Theorem 6.2 is due to Mike Paterson; his help, and the help of an anonymous referee in improving and simplifying this paper are gratefully acknowledged.

References

[1] D.P. Dobkin and R.J. Lipton, On the complexity of computations under varying sets of primitives, *JCSS* **18** (1979) 86–91.
[2] J. Farkas, Uber die Theorie der einfachen Ungleichungen, *Jour. für die Reine u. Angewandte Math.* **124** (1902) 1–27.
[3] M.L. Fredman and B. Weide, On the complexity of computing the measure of $\bigcup [a_i, b_i]$, *Comm. ACM* **21** (1978) 540–544.
[4] J.C. Holaday, Cartesian products of termination games, in: M. Dresher, A.W. Tucker and P. Wolfe, eds., *Contributions to the Theory of Games, Vol. 3* (Princeton University Press, Princeton, NJ, 1957) 189–199.
[5] H.T. Kung, F.L. Luccio and F.P. Preparata, On finding the maxima of a set of vectors, *J. ACM* **22** (1975) 469–476.
[6] U. Manber and M. Tompa, Probabilistic, nondeterministic, and alternating decision trees, *Proc. 14th Ann. ACM Symp. on Theory of Computing* (1982) 234–244.
[7] E.M. Reingold, On the optimality of some set algorithms, *J. ACM* **19** (1972) 649–659.
[8] M. Snir, Proving lower bounds for linear decision trees, *Proc. 8th ICALP Colloquium, Lecture Notes in Computer Science* **115** (*Springer, Berlin*), 1981) 305–315.
[9] C. Yap, On lifted problems, *Proc. IEEE 19th Ann. Symp. on Foundations of Computer Science* (1978) 267–279.
[10] A.C. Yao and R.L. Rivest, On the polyhedral decision problem, *SIAM J. Comput.* **9** (1980) 343–347.