1-1-1991

Selection Networks

Nicholas Pippenger
Harvey Mudd College

SIAM J. COMPUT.
Vol. 20, No. 5, pp. 878–887, October 1991

005

SELECTION NETWORKS*

NICHOLAS PIPPENGER†

Abstract. An upper bound asymptotic to $2n \log_2 n$ is established for the number of comparators required in a network that classifies n values into two classes, each containing $n/2$ values, with each value in one class less than or equal to each value in the other. (The best lower bound known for this problem is asymptotic to $(n/2) \log_2 n$.)

Key words. comparator, classifier, expanding graph, random walk

AMS(MOS) subject classifications. 68E05, 94C10

1. Introduction. The selection networks of which we speak in this paper are comparator networks (see Knuth [K]) that classify a set of n values into two classes, with each of the values in one class being at least as large as all of those in the other. In this paper we shall confine our attention to the simplest case, in which n is even and the two classes each contain $n/2$ values, but similar methods apply to classes of unequal cardinality, as well as to the problem of selecting the value having a prescribed rank, such as the median.

We shall present an upper bound asymptotic to $2n \log_2 n$ for the number of comparators needed to construct such a network. Alekseev [A1] has given a lower bound asymptotic to $(n/2) \log_2 n$. Some perspective on the gap between these bounds is gained by considering the analogous problem of determining the median of n values with an adaptive sequence of comparisons: here the best upper bound known is asymptotic to $3n$ (see Schönhage, Paterson, and Pippenger [S]), and the best lower bound known is asymptotic to $2n$ (see Bent and John [Be]).

The classifying problem has traditionally been considered in connection with the problem of sorting n values into order. In 1983, Ajtai, Komlós, and Szemerédi [Aj] showed that $O(n \log n)$ comparators are sufficient for sorting, and this bound obviously applies to classifying as well. The constant factor implicit in their original proof is enormous, however, and further efforts to refine their ideas have not brought it below 1000 (see Paterson [Pa]). Our classifiers are based on the same fundamental idea as their sorters; our only contribution is to show that in the context of classifiers, it yields both a much simpler proof and a much smaller constant.

Though we shall confine ourselves to proving the result stated above, two additional points should be mentioned. First, we prove the existence of classifying networks without giving an explicit construction. This situation arises from the use of expanding graphs; by exploiting known explicit constructions for expanding graphs (see Pippenger [Pi, § 3.2]), and by accepting a somewhat larger bound (the best we have been able to obtain is slightly less than $6n \log_2 n$), we could give a completely explicit construction. Second, the networks we describe have depth $\Omega((\log n)^2)$; with more care in the construction and proof, we could establish a bound of $O(\log n)$. Our method does not seem well suited to optimizing the depth, however, and we have not made any attempt to obtain the sharpest possible result in this direction.

* Received by the editors April 30, 1990; accepted for publication December 5, 1990. This research was partially supported by Natural Sciences and Engineering Research Council of Canada operating grant and a British Columbia Advanced Systems Institute fellowship award.

† Department of Computer Science, University of British Columbia, Vancouver, British Columbia V6T 1W5, Canada.

2. Expanding graphs. We shall need some results concerning expanding graphs; these will be obtained as special cases of a general result due to Bassalygo [B], to whom we refer for the proof.

A bipartite graph with n "left" vertices and m "right" vertices will be called an (α, β)-*expander* if any k of its left vertices ($k \leqq \lfloor \alpha n \rfloor$) are connected to at least $\lfloor \beta k \rfloor$ right vertices (the set A of left vertices is connected to the set B of right vertices if at least one edge from A leads to each right vertex b, $b \in B$; $\lfloor x \rfloor$ is the integer part of x).

LEMMA 2.1 (Bassalygo). *For any positive integers q and p, any reals α and β $(0 < \alpha < p/\beta q < 1)$, and any sufficiently large n $(n \geqq n_0(\alpha, \beta, q, p))$, there exists an (α, β)-expander with qn left vertices and pn right vertices, for which the number of edges does not exceed $spqn$, where s is any integer greater than*

$$\frac{H(\alpha)+(p/q)H(\alpha\beta q/p)}{pH(\alpha)-\alpha\beta qH(p/\beta q)}; \qquad H(x)=-x\log x-(1-x)\log(1-x), \qquad 0<x<1.$$

The proof of Lemma 2.1 considers only graphs in which every left vertex meets sp edges and every right vertex meets sq edges. This observation will be important when we consider the depth, rather than merely the size, of networks.

We shall use this lemma with $p = q = 1$. We shall let α and β depend on a new parameter ϑ by $\alpha = \vartheta$ and $\beta = (1-\vartheta)/\vartheta$. For every $\vartheta > 0$, there is a value of s that satisfies the hypothesis of Lemma 2.1.

The proof of Lemma 2.1 may be regarded as considering a probability distribution over graphs, and when $p = q$, this distribution is invariant under the exchange of left and right vertices. The proof also shows, not merely that there exists a graph with the prescribed expansion property, but that almost all considered graphs have this property. In particular, a majority of the considered graphs have this property. It follows that there exists a graph such that both it and the graph obtained from it by exchanging left and right vertices have the prescribed expansion property.

Combining these elaborations of Lemma 2.1, we obtain the following corollary.

COROLLARY 2.2. *For every $\vartheta > 0$, there exists an s such that for all sufficiently large n (depending on ϑ) there exists a bipartite graph with n left vertices, n right vertices, and s edges meeting each vertex such that* (1) *every set of $k \leqq \vartheta n$ left vertices is connected to at least $(1-\vartheta)k/\vartheta$ right vertices, and* (2) *every set of $k \leqq \vartheta n$ right vertices is connected to at least $(1-\vartheta)k/\vartheta$ left vertices.*

Returning to Lemma 2.1 with $p = q = 1$, if we take $s = 4$ and choose $\beta < 3$, then the hypothesis is satisfied for all sufficiently small $\alpha > 0$ (depending on β). Thus we also obtain the following corollary.

COROLLARY 2.3. *For every $\beta < 3$, there exists an $\alpha > 0$ such that for all sufficiently large n (depending on β), there exists a bipartite graph with n left vertices, n right vertices, and four edges meeting every vertex such that* (1) *every set of $k \leqq \alpha n$ left vertices is connected to at least βk right vertices, and* (2) *every set of $k \leqq \alpha n$ right vertices is connected to at least βk left vertices.*

3. Classifiers. A comparator network with $2m$ inputs and two sets of m outputs, the "left" outputs and the "right" outputs, is an α-*weak approximate classifier with tolerance* ϑ (or simply an α-*weak* ϑ-*classifier*) if, for any assignment of values to the inputs and positive integer $k \leqq \alpha m$, (1) at most ϑk of the k smallest values appear at right outputs, and (2) at most ϑk of the k largest values appear at left outputs. A 1-weak approximate classifier with tolerance ϑ will be called an *approximate classifier with tolerance* ϑ (or simply a ϑ-*classifier*). An approximate classifier with tolerance 0 will be called a *classifier.*

Classifiers are the goal of our construction. Approximate classifiers and weak approximate classifiers are the ultimate building blocks of our construction. These building blocks are secured by a lemma that is a slight generalization of the most basic lemma of Ajtai, Komlós, and Szemerédi [Aj], from which the proof is easily adapted (see also Pippenger [Pi, § 3.2]).

Let G be a bipartite graph with n left vertices and n right vertices, in which every vertex meets s edges. The edges of G may be decomposed into s perfect matchings $E_1, \cdots, E_s$ between the left and right vertices. We may regard each perfect matching E_r as a comparator network, by taking a comparator for each edge in E_r, labelling the inputs of the comparator with the vertices met by the edge, labelling the smaller output of the comparator with the left vertex met by the edge, and labelling the larger output of the comparator with the right vertex met by the edge. We may then combine the comparator networks $E_1, \cdots, E_s$ into a single comparator network by identifying the outputs of E_r with the corresponding inputs of E_{r+1} for each $1 \leqq r \leqq s-1$. We shall denote the resulting comparator network by G; this notation is ambiguous, since different decompositions of the bipartite graph yield different comparator networks, but this ambiguity will not be important to us.

LEMMA 3.1 (Ajtai, Komlós, and Szemerédi). *Let G be a bipartite graph with n left vertices and n right vertices and s edges meeting every vertex in which* (1) *any set of $k \leqq \alpha\vartheta n$ left vertices is connected to at least $(1-\vartheta)k/\vartheta$ right vertices, and* (2) *any set of $k \leqq \alpha\vartheta n$ right vertices is connected to at least $(1-\vartheta)k/\vartheta$ left vertices. Let the left and right outputs of the comparator network G be those labelled by the left and right vertices, respectively. Then G is an α-weak ϑ-classifier.*

4. Recursive construction. A comparator network with $2m$ inputs, l outputs labelled as "low," l outputs labelled as "high," and $2m-2l$ outputs labelled as "middle" is a *strong partial classifier* if, for any assignment of values to the inputs, only values among the $m/2$ smallest appear at low outputs and only values among the $m/2$ largest appear at high outputs. A strong partial classifier is less than a classifier in that there are some outputs, the middle outputs, at which any value may appear; but it is more than a classifier in that fewer values can appear at the low and high outputs. Our goal in this section is to show how strong partial classifiers can be assembled to form a classifier.

It will be convenient to use strong partial classifiers for which the number of inputs is of the form 2^ν or $3 \cdot 2^\nu$, with ν a positive integer; numbers of this form will be called *magic*. If n is any even positive integer, the largest magic number not exceeding n will be called the *magic part* of n; it is even and at least $2n/3$.

Consider the following recursive construction for a classifier with n inputs. Let $2m$ denote the magic part of n. Feed $2m$ of the inputs into a strong partial classifier with $2m$ inputs. Feed the remaining $n-2m$ inputs, together with the $2m-2l$ middle outputs of the strong partial classifier into a classifier with $2n-2l$ inputs. The left and right outputs of the combined network will be the low and high outputs, respectively, of the strong partial classifier, together with the left and right outputs, respectively, of the constituent classifier.

A value appearing at a low output of the strong partial classifier must be among the $m/2$ smallest of the $2m$ values at its inputs, and thus among the $(m/2)+(n-2m) = n-3m/2$ smallest of all n values. Since $2m \geqq 2n/3$, it must be among the $n/2$ smallest of all n values. Similarly, a value appearing at a high ouput of the strong partial classifier must be among the $n/2$ largest of all n values. Thus, of the $n/2$ largest and $n/2$ smallest values, equal numbers appear at the inputs of the classifier with $n-2l$ inputs. It follows that any value appearing at a left output of this classifier must be

among the $n/2$ smallest, and any value appearing at a right output must be among the $n/2$ largest. Thus the combined network is indeed a classifier.

LEMMA 4.1. *Let $C>0$ be a constant. Suppose that, for every $\varepsilon>0$ and all sufficiently large m (depending on ε), there exists a strong partial classifier with $2m$ inputs, l low outputs, and l high outputs, and size at most $(C+\varepsilon)l \log_2 m$. Then for every $\varepsilon>0$ and all sufficiently large n (depending on ε), there exists a classifier with n inputs and size at most $(C/2+\varepsilon)n \log_2 n$.*

Proof. Apply the recursive construction until the number of inputs of the strong partial classifier that is needed is too small for the hypothesis to apply. Terminate the recursion with a sorting network using $\binom{m}{2}$ comparators. The size of this final sorting network depends only on ε, and thus is at most $(\varepsilon/2)n \log_2 n$ for all sufficiently large n (depending on ε).

Let $2m_1, \cdots, 2m_s$ denote the numbers of inputs of the strong partial classifiers, and let $l_1, \cdots, l_r$ denote the numbers of low outputs. We have $2m_r \leqq n$ for all $1 \leqq r \leqq s$ and, since each left output of the combined network is a low output of at most one strong partial classifier, $\sum_{1\leqq r\leqq s} l_r \leqq n/2$. Thus the total size of all the strong partial classifiers is at most $\sum_{1\leqq r\leqq s} (C+\varepsilon)l_r \log_2 m_r \leqq (C/2+\varepsilon/2)n \log_2 n$. Adding the bound $(\varepsilon/2)n \log_2 n$ for the final sorter yields $(C/2+\varepsilon)n \log_2 n$. □

5. Crude classification trees. This section introduces classification trees, the basic tool we shall use to construct strong partial classifiers. We shall begin with a crude version of the construction, and later refine it to obtain our final bound.

Set $\vartheta = \frac{1}{8}$. Corollary 2.2 and Lemma 3.1 then yield constants s_0 and n_0 such that for all $n \geqq n_0$, there is a ϑ-classifier with $2n$ inputs and depth s_0. (A simple calculation shows that $s_0 = 28$. The determination of n_0 would require scrutiny of the proof of Lemma 2.1, but this proof consists of explicit estimates, so that n_0 is at least effectively calculable. The actual values of these constants will not be important to us.)

Suppose that we wish to construct a strong partial classifier with $2m$ inputs, where $2m$ is a magic number. Feed the $2m$ inputs into a ϑ-classifier with $2m$ inputs. This approximate classifier has m left outputs and m right outputs. Feed each of these sets of outputs into a ϑ-classifier with m inputs. These two approximate classifiers have four sets of outputs. Feed each of these sets into a ϑ-classifier with $m/2$ inputs, and continue in this way until the sets of outputs of the approximate classifiers have cardinality less than $2n_0$. The result is a tree of approximate classifiers that we shall call a *classification tree.* At its root are $2m$ inputs, and at its leaves are sets of outputs each containing fewer than $2n_0$ outputs.

The next step will be to label the outputs as low, high, and middle in such a way that the result is a strong partial classifier. When, as we do this, we assign the same label to all the outputs in a subtree, we may prune away that subtree, and affix the label to the outputs of the approximate classifier feeding the subtree. A large fraction of the tree will be eliminated in this way.

We begin by labelling as middle the right outputs of the left child of the root, and the left outputs of the right child of the root (and pruning away the subtrees below). We shall label as low some of the outputs in the subtree fed by the left outputs of the left child, and as high some of the outputs in the subtree fed by the right outputs of the right child. We shall now describe which outputs are to be labelled as low. The mirror image of this procedure will label an equal number of outputs as high.

Consider the $m/2$ smallest values assigned to the inputs, since it is these that are eligible to appear at an output labelled as low. We shall call these $m/2$ values *good*, and the other $3m/2$ values *bad.*

At most, a fraction $\vartheta = \frac{1}{8}$ of the good values can appear at right outputs of the approximate classifier at the root, and at most a fraction $\frac{1}{8}$ can appear at right outputs of the approximate classifier that is its left child. Thus at least a fraction $1 - (\frac{1}{8}) - (\frac{1}{8}) = \frac{3}{4}$ of the good values appear at left outputs of the left child. Since the number of good values equals the number of left outputs of the left child, at most a fraction $1 - (\frac{3}{4}) = \frac{1}{4}$ of the values appearing at these outputs are bad.

We may characterize the set of left outputs of the left child by its cardinality $m/2$ and its "impurity" $\frac{1}{4}$ (the largest possible fraction of its values that could be bad). Suppose now that we have a set of outputs of some approximate classifier with cardinality k and impurity η. First, if $\eta > \frac{1}{2}$, we shall label these outputs as middle (and prune away the subtree below). Second, if $\eta k < 1$, then not a single bad value can appear at one of these outputs; thus we shall label them as low (and prune away the subtree below). Finally, if $\eta \leqq \frac{1}{2}$ and $\eta k \geqq 1$, then we shall consider the sets of outputs of the child. The set of left outputs has cardinality $k/2$ and (by Lemma 3.1) impurity $2\vartheta\eta = \eta/4$, and the set of right outputs has cardinality $k/2$ and impurity 2η (the factors of 2 in the impurities arise because we are considering a fraction of half as many things). We may continue in this way along each path in the tree until we assign a label or reach a leaf. If we reach a leaf, we shall label its outputs as middle if $\eta k \geqq 1$, and as low if $\eta k < 1$.

The first question we shall ask is: what fraction of the outputs are labelled as middle by being in a set with impurity exceeding $\frac{1}{2}$? To answer this question, we shall consider the following random walk on the integers. Start at the position 2, $Z_0 = 2$. At each step, independently move to the position one smaller, $Z_{t+1} = Z_t - 1$, or two larger, $Z_{t+1} = Z_t + 2$, with equal probabilities. What is the probability of ever reaching a position smaller than 1? Since the walk is confined to the integers, this is the probability of ever reaching the position 0, $Z_t = 0$. The answer to this question is an upper bound to the fraction of the outputs that are labelled as middle by being in a set with impurity exceeding $\frac{1}{2}$, as can be seen by considering the correspondence between paths in the tree and walks, where the number of levels from the root corresponds to time in the walk, and the negative of the logarithm (to base 2) of the impurity corresponds to position in the walk.

In the present instance, the probability of ever reaching the position 0 can be determined explicitly and is $(3-\sqrt{5})/2 = 0.382\cdots$. To see this, let $f(x)$ denote the power series in x in which the coefficient of x^t is the number of walks that start at 1 and reach 0 for the first time at time t. Then $f(x)^2$ is the power series for walks that start at 2 and reach 0 for the first time at time t, since each such walk can be uniquely parsed into two subwalks according to the time at which it first reaches 1, and the numbers of possibilities for both subwalks are counted by $f(x)$. Thus the probability we seek is $f(\frac{1}{2})^2$, so it will suffice to show that $f(\frac{1}{2}) = (-1+\sqrt{5})/2 = 0.618\cdots$. Let $g(x)$ count the number of walks that start at 1 and return to 1 for the first time at time t. Then $g(x) = xf(x)^2$, since such a walk must go to 3 on the first step, then return to 1 for the first time in $t-1$ more steps. On the other hand, $f(x) = x + xg(x) + xg(x)^2 + \cdots = x/(1-g(x))$, since the walks counted by $f(x)$ may be classified according to the number of times they visit 1 before reaching 0. Thus $f(x)$ satisfies the equation $xf(x)^3 - f(x) + x = 0$, so that $f(\frac{1}{2})$ satisfies $f(\frac{1}{2})^3 - 2f(\frac{1}{2}) + 1 = 0$, which yields the stated result.

Next we shall ask: what fraction of the outputs are labelled as middle by being in a leaf that is not pruned away? Such a leaf has cardinality at most $2n_0$, and thus it must have impurity at least $\frac{1}{2}n_0$ to avoid being labelled as low. Let d denote the number of levels of approximate classifiers in the tree. Rephrased in terms of random walks, our question becomes: what is the probability of being at a position at most $c_0 = \log_2 n_0$

at time $d-2$? (The first two levels of the tree do not correspond to steps of the random walk.) We shall answer this question with a lemma that goes beyond our present needs, but which will be applied repeatedly later.

We shall consider random walks with discrete time indexed by the natural numbers and discrete positions indexed by the integers. We shall assume that the steps are independent and identically distributed, but we shall allow the steps to have any probability distribution on a finite set of integers. We shall say that such a random walk is *positively biased* if the expectation of a step is positive. (In the present instance, the step is uniformly distributed on the set $\{-1, 2\}$, and the expectation is $(-1+2)/2 = \frac{1}{2} > 0$.)

LEMMA 5.1. *Let Z_t, $t = 0, 1, 2, \cdots$, be a positively biased random walk starting at 0, and let c be any position. Then there exist constants A and $b < 1$ such that for all t, the probability that Z_t is at most c does not exceed Ab^t.*

Proof. Let $\Phi(\xi) = \mathrm{Ex}(\exp-(\xi Z_1))$. The power series expansion of $\Phi(\xi)$ is $1-\xi\,\mathrm{Ex}\,(Z_1)+O(\xi^2)$. Since $\mathrm{Ex}\,(Z_1)>0$, we can choose $\xi_0>0$ sufficiently small so that $\Phi(\xi_0)<1$. Since the steps are independent and identically distributed, we have $\mathrm{Ex}\,(\exp-(\xi_0 Z_t)) = \Phi(\xi_0)^t$. If $Z_t \leqq c$, then $\exp-(\xi_0 Z_t) \geqq \exp-(\xi_0 c)$. Thus, by Markov's inequality, the probability that $Z_t \leqq c$ is at most $\Phi(\xi_0)^t/\exp-(\xi_0 c)$, so we may take $A = \exp(\xi_0 c)$ and $b = \Phi(\xi_0)$. □

We may now apply Lemma 5.1 with $t = d-2 \geqq \log_2(m/4n_0)$ and conclude that the fraction of outputs that are labelled as middle by being in a leaf that is not pruned away is at most $Ab^d \leqq Cm^{-e}$, where C and $e>0$ are constants. The only feature of this bound that is relevant to our present purposes is that it tends to zero, even when multiplied by $d \leqq \log_2 m$.

Finally, we shall ask: what is the size of the approximate classifier constructed in this way? To answer this question, we shall again transform it into a question about random walks. In a "synchronous" comparator network (in which the two inputs of any comparator are at the same depth), each comparator contributes 2 to the sum of the depths of the outputs. Thus the number of comparators is $n/2$ times the average depth of the outputs. Since the depth of each approximate classifier is s_0, the number of comparators is $s_0 n/2$ times the average level at which the outputs are labelled. For outputs labelled as low or labelled as middle by being in a leaf that is not pruned away, the level at which they are labelled is at most $d \leqq \log_2 m$. (For outputs labelled as low, it is in most cases substantially less than this, but we shall not attempt to exploit this effect, since later optimizations will render it negligible.) For outputs labelled as middle because their impurity exceeds $\frac{1}{2}$, the level at which they are labelled is two more than the number of steps taken by the corresponding walk to reach position 0 for the first time. (Again, the first two levels of the tree do not correspond to steps of the random walk.)

In the present instance, this average number of steps can be calculated explicitly and is $4/\sqrt{5} = 1.788\cdots$. It is obtained from the power series $f(x)^2$ that counts the walks by evaluating $xd(f(x)^2)/dx$ at $x=\frac{1}{2}$ or, equivalently, evaluating $f'(x)f(x)$ at $x=\frac{1}{2}$. This evaluation is most conveniently accomplished by dividing the equation $xf(x)^3 - f(x) + x = 0$ by x, differentiating with respect to x, multiplying by x^2, solving for $f'(x)$ in terms of $f(x)$ and x, multiplying by $f(x)$, and evaluating the result at $x=\frac{1}{2}$. Taking account of the equation $f(\frac{1}{2})^2 = (3-\sqrt{5})/2$, derived earlier, yields the stated result.

We can now sum the contributions to the size of the strong partial classifier. The l outputs labelled as low contribute at most $(s_0 l/2)\log m$, and those labelled as high contribute equally. The outputs labelled as middle by being in a leaf that is not pruned

away contribute at most $Fm^{1-e}\log_2 m$ for some constants F and $e>0$, and the outputs labelled as middle because their impurity exceeds $\frac{1}{2}$ contribute at most Gm for some constant G. Since the $l=\Omega(m)$, we conclude that for every $\varepsilon>0$ and all sufficiently large m (depending on ε), there exists a strong partial classifier with $2m$ inputs, l outputs labelled as low and an equal number labelled as high, and size at most $(s_0+\varepsilon)l\log_2 l$. It follows from Lemma 4.1 that for every $\varepsilon>0$ and all sufficiently large n (depending on ε), there exists a classifier with n inputs and size at most $(s_0/2+\varepsilon)n\log_2 n$. The remainder of this paper is devoted to refining the construction just given to reduce the constant $s_0/2$ to 2.

6. Refined classification trees. If we ask what properties were essential to the construction in the preceding section, we find three. First, the probability that the random walk ever reaches the position 0 is strictly less than 1. Second, the probability that the walk is near position 0 after t steps decreases exponentially with t. Third, the expected number of steps needed to reach 0 (with no contribution from walks that never reach 0) is finite.

The second property is a consequence of the random walk being positively biased. Thus it is natural to seek ways to reduce the number of comparators while preserving the property that the corresponding random walk is positively biased. When this is done, the explicit calculations by which we established the first and third properties will no longer be feasible, but we will see that these properties are consequences of the second property.

The property that the random walk was positively biased follows from the inequality $\vartheta<\frac{1}{4}$ (so that the geometric mean of the factors 2 and 2ϑ, by which impurities change from parent to child, is less than 1). We shall arrange for ϑ to vary from level to level in such a way that the average (again in the sense of the geometric mean) of ϑ is strictly less than $\frac{1}{4}$, but by a very small margin. We shall also exploit the fact that for most of the approximate classifiers in the classification tree, the number of bad elements is very small, so that we may substitute weak approximate classifiers (as defined in § 3).

Let h be a positive integer. Set $\vartheta_h=2^{1/h}/4$ and $\beta_h=(1-\vartheta_h)/\vartheta_h$. Since $\vartheta_h>\frac{1}{4}$, we have $\beta_h<3$. Corollary 2.3 and Lemma 3.1 establish the existence of $\alpha_h>0$ such that, for all sufficiently large n (depending on h), there exists an α_h-weak ϑ_h-classifier with $2n$ inputs and depth 4.

Let us now define a *gadget* to be a tree comprising h levels of α_h-weak approximate classifiers, which therefore approximately classifies the values assigned to its inputs into 2^h classes. The α_h-weak approximate classifier at the root will have tolerance $\frac{1}{8}$, and those at the remaining $h-1$ levels will have tolerance ϑ_h.

Suppose that at most a fraction η of the values assigned to the inputs of the gadget are bad, where $\eta\leqq\alpha_h$. We may determine the impurities of the 2^h sets of outputs by proceeding through successive levels of the tree as before. The root multiplies the impurity by a factor of 2 or $\frac{1}{4}$, and every other node multiplies it by a factor of 2 or $2^{1/h}/2$. A simple calculation shows that the geometric mean of the impurities of the 2^h sets of outputs is $(\frac{1}{2})^{1/2h}$, which is less than 1.

Since all changes to impurities are by factors of $2^{1/h}$, we may consider a random walk on the integers by taking the position to be h times the negative of the logarithm (to base 2) of the impurity, and letting the probability distribution for each step correspond to the changes to impurities for a gadget. The expectation for a step is h times the negative of the logarithm of the geometric mean of the changes, which is $\frac{1}{2}$. Thus the random walk is positively biased.

Since every path through a gadget passes through one weak approximate classifier with tolerance $\frac{1}{8}$ and through $h-1$ with tolerance $\vartheta_h > \frac{1}{4}$, we can construct a gadget with depth at most $4(h-1)+s_0$.

The gadget we have constructed is composed from α_k-weak approximate classifiers, and thus is only useful when the number of bad values is small. We shall call the gadgets described above *cheap* gadgets. We shall also define *dear* gadgets, which have the same tree structure, but with approximate classifiers (rather than weak approximate classifiers) at all nodes, and with all classifiers having tolerance $\frac{1}{8}$. We shall use the same step probability distribution for dear gadgets that we used for cheap gadgets (though of course dear gadgets do much more). The depth of a dear gadget is at most hs_0.

We shall now construct a refined classification tree using gadgets rather than approximate classifiers as nodes below the root and its children (so that the tree is 2^h-ary rather than binary below the first two levels). We assign impurities to each set of outputs of each gadget as before. We shall use a cheap gadget when the impurity of the set of inputs is at most α_h, and a dear gadget when the impurity exceeds α_h.

We shall label outputs as low, high, or middle, and prune away subtrees as before. It remains to estimate the number of outputs labelled as low, high, or middle, and to estimate the size of the resulting strong partial classifier.

The fraction of the outputs that are labelled as middle because their impurity exceeds $\frac{1}{2}$ can be bounded as before by the probability that the corresponding random walk reaches a nonpositive position. (We must consider nonpositive positions, rather than just the position 0, because a single step may now decrease the position by more than 1.) For this purpose we shall use the following lemma.

LEMMA 6.1. *Consider a positively biased random walk starting at a positive position. Then the probability that the walk ever reaches a nonpositive position is strictly less than* 1.

Proof. By Lemma 5.1, the probability that the walk is at a nonpositive position at time t is at most Ab^t for some constants A and $b<1$. The series $\Sigma_{t\geqq 1} Ab^t$ converges, so we may choose T sufficiently large that $\Sigma_{t\geqq T} Ab^t \leqq \frac{1}{2}$. Let $-\Delta$ denote the most negative step that is taken with positive probability, and let $P>0$ denote the probability that a step is positive. Then $Z_{\Delta T} \geqq \Delta T$ with probability at least $P^{\Delta T}$. If this event occurs, the walk cannot reach a nonpositive position in fewer than T additional steps, and the probability of it reaching a nonpositive position in T or more additional steps is no larger than the probability of reaching a position at most ΔT in T or more additional steps, and this is at most $\frac{1}{2}$. Thus the probability of ever reaching a nonpositive position is at most $(1-P^{\Delta T})+P^{\Delta T}/2<1$. □

Using Lemma 5.1 as before, we can again show that the fraction of the outputs that are labelled as middle by being in a leaf that is not pruned away is at most Cm^{-e}, where C and $e>0$ are constants.

To estimate the size of the strong partial classifier we have constructed, we again begin by considering the average level at which outputs are labelled as middle because their impurity exceeds $\frac{1}{2}$. This is bounded by the expectation for the corresponding random walk of the number of steps needed to reach a nonpositive position (with no contribution from walks that never reach a nonpositive position). We shall use the following lemma.

LEMMA 6.2. *Consider a positively biased random walk starting from a positive position. Let U denote the number of steps needed to reach a nonnegative position, if the walk ever reaches a nonpositive position, or* 0 *if the walk never reaches a nonpositive position. Then the expectation of U is finite.*

Proof. Applying Lemma 5.1 with c the negative of the starting position, we have that the probability of being at a nonpositive position at time t is at most Ab^t for some constants A and $b<1$. The convergent series $\Sigma_{t\geq 1} Ab^t t$ bounds the sum over all visits to nonpositive positions of the times at which the visits occur. This in turn bounds the sum over first visits, which is exactly U. □

For refined classification trees, we shall need an additional estimate concerning the total size of dear gadgets. Since a dear gadget is used precisely when the impurity exceeds a threshold α_h, then the total size of dear gadgets can be bounded in terms of the average time spent by the random walk at positions less than a corresponding constant $c_h = -h \log_2 \alpha_h$. By Lemma 5.1, this average time is bounded by a convergent series $\sum_{t\geq 1} Ab^t$, for some constants A and $b<1$, and thus is finite. It follows that the total size of dear gadgets is at most Hm, for some constant H.

Summing the contributions to the size as before, we find that for every h and all sufficiently large m (depending on h), there exists a strong partial classifier with $2m$ inputs, l outputs labelled as low and an equal number labelled as high, and size at most $(4(h-1)+s_0+1)l \log_{2^h} l = (4+(s_0-3)/h)l \log_2 l$. Letting h tend to infinity, we see that for every $\varepsilon>0$ and all sufficiently large m (depending on ε), there exists a strong partial classifier as above with size at most $(4+\varepsilon)l \log_2 l$. It follows from Lemma 4.1 that for every $\varepsilon>0$ and all sufficiently large n (depending on ε), there exists a classifier with n inputs and size at most $(2+\varepsilon)n \log_2 n$. Thus we have achieved the goal of this paper.

7. Embellishments. In this section we shall offer some additional comments on the embellishments to our main result that were mentioned in the introduction.

If one wishes to classify n values into classes of cardinality t and $n-t$, this can be accomplished by modifying the definitions of some of the components in the construction. One redefines "strong partial classifier" so that the numbers of outputs labelled as low and high are in the correct proportion, $t: n-t$, and so that $t/2$ and $(n-t)/2$ values are eligible to appear as low and high outputs, respectively. One must then modify the upper levels of the classification trees to accommodate the new definitions of "good" and "bad" values, which now differ on the two sides of the trees. (Special care is necessary if t or $n-t$ is much smaller than n.) The constant factor in the final result is independent of t, but this should be regarded as a deficiency rather than a merit, since it is to be expected that this constant should decrease as t is varied away from $n/2$.

The problem of obtaining an explicit construction is attacked by replacing Lemma 2.1 by an analogous explicit result (see Pippenger [Pi, § 3.2]). A complication arises from the fact that the explicit results hold only for certain n, rather than for all sufficiently large n. Thus one obtains, for example, weak approximate classifiers with $2(q+1)$ inputs, where q is a prime congruent to 1 modulo 4. In constructing classification trees, one must always settle for using the next smaller weak approximate classifier of this form, and reconcile oneself to labelling the remaining outputs of the parent approximate classifier as middle. Standard results on the distribution of primes, however, allow one to show that only a negligible fraction of the outputs are labelled as middle in this way. Using weak approximate classifiers with depths 8, 12, and 30 results in a final bound slightly less than $6n \log_2 n$.

Finally, if one wishes to ensure that the depth of the final classifier is $O(\log n)$, one must modify the recursive construction so that the various strong partial classifiers are "overlapped" in depth. To do this, it is most convenient to make them "stronger" (so that $m/4$ rather than $m/2$ values are eligible to appear at an output labelled as

low or high). One can then show that enough outputs are available from the shallower levels of the first r strong partial classifiers to supply inputs for the $(r+1)$st strong partial classifier.

8. Conclusion. We have established the existence of classifiers with many fewer comparators than those previously known. For most constructions that rely on expanding graphs (such as those given by Bassalygo [B]), the constant factor depends on the current state of technology of expanding graphs, and can be expected to improve with further advances in the state of this art. The result of this paper, however, will not be improved in this way: the constant in the leading term of the size depends only on the degree needed for expanding graphs to expand very small sets, and this aspect of expanding graphs is understood completely (graphs of degree s can expand small sets by any factor up to $s-1$, but not by more).

It would be of interest to see if a similar situation can be brought about for other applications of expanding graphs, perhaps even for the most celebrated application of all, the sorting networks of Ajtai, Komlós, and Szemerédi.

REFERENCES

[Aj] M. AJTAI, J. KOMLÓS, AND E. SZEMERÉDI, *Sorting in c* log *n parallel steps*, Combinatorica, 3 (1983), pp. 1–19.

[Al] V. E. ALEKSEEV, Sorting algorithms with minimum memory, Kibernetica, 5 (1969), pp. 99–103.

[B] L. A. BASSALYGO, *Asymptotically Optimal Switching Circuits*, Problems Inform. Transmission, 17 (1981), pp. 206–211.

[Be] S. W. BENT AND J. W. JOHN, *Finding the median requires* $2n$ *comparisons*, in Proc. 17th ACM Symposium on Theory of Computing, 1985, pp. 213–216.

[K] D. E. KNUTH, *The Art of Computer Programming*, Vol. 3, *Sorting and Searching*, Addison-Wesley, Reading, MA, 1973.

[Pa] M. S. PATERSON, *Improved sorting networks with* $O(\log n)$ *depth*, Algorithmica, to appear.

[Pi] N. PIPPENGER, *Communication Networks*, in Handbook of Theoretical Computer Science, J. van Leeuwen, ed., North-Holland, Amsterdam, 1990.

[S] A. SCHÖNHAGE, M. PATERSON, AND N. PIPPENGER, *Finding the Median*, J. Comput. System Sci., 13 (1976), pp. 184–199.