ON THE DIFFICULTY OF DESIGNING GOOD CLASSIFIERS[*]

MICHELANGELO GRIGNI[†], VINCENT MIRELLI[‡], AND CHRISTOS H. PAPADIMITRIOU[§]

Abstract. We consider the problem of designing a near-optimal linear decision tree to classify two given point sets B and W in $\Re^n$. A linear decision tree defines a polyhedral subdivision of space; it is a classifier if no leaf region contains points from both sets. We show hardness results for computing such a classifier with approximately optimal depth or size in polynomial-time. In particular, we show that unless NP=ZPP, the depth of a classifier cannot be approximated within any constant factor, and that the total number of nodes cannot be approximated within any fixed polynomial. Our proof uses a simple connection between this problem and graph coloring, and uses the result of Feige and Kilian on the inapproximability of the chromatic number. We also study the problem of designing a classifier with a single inequality that involves as few variables as possible, and point out certain aspects of the difficulty of this problem.

Key words. linear decision tree, hardness of approximation, parameterized complexity

AMS subject classifications. 68Q17, 62H30

1. Introduction.

Classifying point sets in $\Re^n$ by linear decision trees is of great interest in pattern analysis and many other applications [4, 5, 14]. Typically, in such a problem we are given a set W of *white* points and a set B of *black* points in $\Re^n$, and we must produce a linear decision tree which classifies them. That is, the tree defines a linear decision at each internal node, such that for each leaf ℓ of this tree, either only white or only black points lead the algorithm to ℓ. We call such a linear decision tree a *classifier*. In many situations W and B are not given explicitly, but implicitly in terms of concepts, images of objects, etc.

The problem is already well-studied. Constructing a size-optimal classifier is NP-complete even in three dimensions [10]; in high dimensions it is NP-complete even for constant size trees [2, 16]. There is much algorithmic work towards computing classifiers that meet various local optimality conditions [4, 17], but very little is known about how well such local optima approximate the optimal solution. An exception is the use of random sampling to find near-optimal splitting planes in low dimensions [10].

In this paper we prove some very strong negative results on high-dimensional classifying trees (the important case in practice). We point out a simple connection between the problem of designing optimal linear classifying trees and the classical problem of *coloring a graph*. Given a graph G, we construct its geometric realization; roughly speaking, the white points are the vertices of the graph arranged at the corners of a simplex, and the black points correspond to the edges of the graph, with each black point placed at the midpoint between the two endpoints of its edge. It is not hard to prove then that the optimum size of any classifier is the chromatic number of the graph $\chi(G)$, while the optimum depth is $\log_2(\chi(G)+1)$. We then use the result of Feige and Kilian [8] on the inapproximability of the chromatic number, to obtain these two results:

[*] This work was supported by the Army Research Laboratory.

[†] Department of Mathematics and Computer Science, Emory University, Atlanta, Georgia, `mic@mathcs.emory.edu`. Work performed at UCSD, supported in part by NSF grant No. DMS-9206251.

[‡] U.S. Army Research Lab., Fort Belvoir, Virginia, `vmirelli@arl.mil`.

[§] Computer Science Division, University of California at Berkeley, Berkeley, California, `christos@cs.berkeley.edu`.

THEOREM 1.1. *Unless NP=ZPP, no polynomial-time algorithm for optimizing the number of nodes in a classifier can approximate the optimum within any fixed polynomial. For $\epsilon > 0$ and large enough dimension n, the approximation ratio is no better than $n^{1-\epsilon}$.*

THEOREM 1.2. *Unless NP=ZPP, no polynomial-time algorithm for optimizing the depth a classifier can have approximation ratio better than any fixed constant.*

Here ZPP is the class of problems solved by polynomial expected-time *randomized* algorithms with neither false negatives nor false positives. NP=ZPP is a situation almost as unthinkable as NP=P. In the next Section we prove these two results.

Finally, in Section 3 we look at another aspect of the difficulty of optimizing classifiers: Suppose that the two point sets can be separated by a *single* linear inequality, but we want to find the inequality that separates them *and involves as few variables as possible.* This situation is of interest when we use functions of the points as additional coordinates to facilitate classification [4, 11]. We point out that variants of this problem are hard for various levels of the *W hierarchy* [3, 6], which implies that (unless an unlikely collapse occurs), they cannot be solved in polynomial-time even if the optimum sought is small (bounded by any very slowly growing function).

2. Definitions and proofs. Let $W, B \subseteq \Re^n$ be two point sets. A *linear classifying tree* for W and B is a decision tree with internal nodes of the form $\sum_{i=1}^n a_i x_i > b$, each with two branches, the *true* branch and the *false* branch. A leaf ℓ of such a tree corresponds in a straightforward way to a convex cell in a subdivision of $\Re^n$, call it $C(\ell)$, containing all points that satisfy (or falsify) the inequality in each internal node I that is an ancestor of ℓ in the tree, and such that ℓ is in the *true* (respectively, *false*) subtree of I.

There are two important measures of the difficulty of such a classifier. The first is the *number of internal nodes* of the tree, and corresponds to the program size of the classifier. The other is the *depth* of the tree, and corresponds to the running time of the decision algorithm. We denote by $d(W, B)$ the depth of the classifier for W and B that has the smallest possible depth among all such classifiers; similarly, $n(W, B)$ is the optimum number of internal nodes.

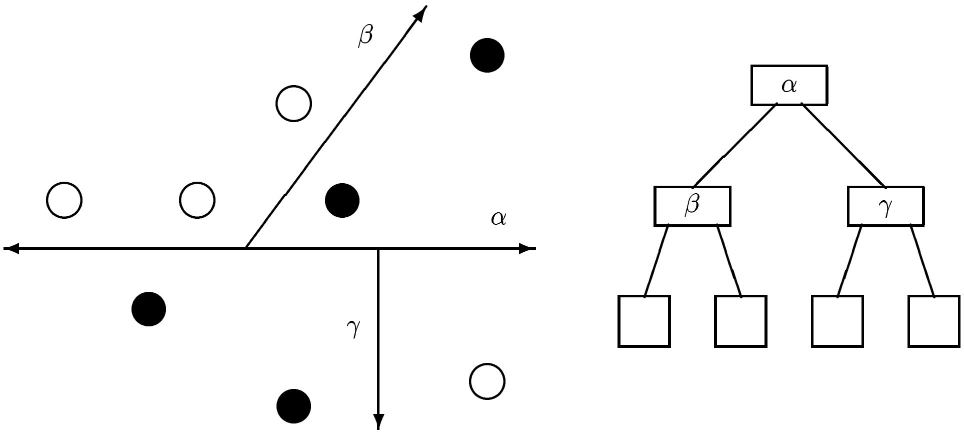

FIG. 2.1. *A two dimensional classifier.*

For example, a classifier for the two 2-dimensional point sets W and B shown in Figure 2.1(a) is shown in Figure 2.1(b). The subdivisions corresponding to the leaves are also shown in Figure 2.1(a). It has depth two, and a total of three nodes. Here it is easy to see that $d(W, B) = 2$ and $n(W, B) = 2$; thus the tree shown is optimal with respect to depth, but not with respect to the number of nodes.

Let $G = (V, E)$ be any graph, with vertices $V = \{v_1, \ldots, v_n\}$ and edges $E = \{e_1, \ldots, e_m\}$. Consider the following two point sets in $\Re^n$ (indeed, on the $(n-1)$-dimensional hyperplane $\sum_{i=1}^n x_i = 1$): the white set $W(G) = \{w_1, \ldots, w_n\}$, where w_i is the ith elementary basis vector (that is, $(w_i)_i = 1$ and all other coordinates are zero); and the black set $B(G) = \{b_1, \ldots, b_m\}$, with $b_k = \frac{1}{2}(w_i + w_j)$ where $e_k = \{v_i, v_j\}$. In other words, the white points are the nodes of G placed at the vertices of the simplex, while the black points are the edges of G, each placed at the midpoint of its two endpoints.

The chromatic number of G, $\chi(G)$, is the smallest number of colors that can be used to color the nodes of G so that no two adjacent nodes have the same color; equivalently, it is the smallest number of independent sets that can be used to cover all nodes of G.

The following two lemmata now characterize the complexity of classifying $W(G)$ and $B(G)$ in terms of $\chi(G)$.

LEMMA 2.1. $n(W(G), B(G)) = \chi(G)$.

Proof. Consider any white leaf ℓ in any decision tree for $W(G), B(G)$. Since its cell $C(\ell)$ is convex, it follows that the nodes of G it contains share no edge, because otherwise the corresponding black midpoint would also be in $C(\ell)$. Thus, $C(\ell)$ contains an independent set of G. Since the leaves of the decision tree must cover all nodes of G, there are at least $\chi(G)$ white leaves in any decision tree. In addition there must be at least one black leaf, and hence there are at least $\chi(G) + 1$ leaves overall, and at least $\chi(G)$ internal nodes. It follows that $n(W(G), B(G)) \geq \chi(G)$.

For the other direction let $S_1, \ldots, S_{\chi(G)}$ be the independent sets in the optimum coloring of G. We can construct a decision tree with $\chi(G)$ internal nodes, of which the kth has the inequality $\sum_{v_i \in S_k} x_i \geq \frac{2}{3}$, with the *true* branch leading to a white leaf and the *false* branch leading to either the $k+1$st internal node, or a black leaf if $k = \chi(G)$. It is easy to see that this is a classifier for $W(G), B(G)$, and hence $n(W(G), B(G)) \leq \chi(G)$. □

LEMMA 2.2. $\lceil \log_2(\chi(G) + 1) \rceil \leq d(W(G), B(G)) \leq \lceil \log_2(\chi(G) + 1) \rceil + 1$.

Proof. The lower bound follows from the previous lemma, since $d(W, B) \geq \lceil \log_2(n(W, B) + 1) \rceil$. For the upper bound, consider the optimum coloring of G with $\chi(G)$ colors. We let V_1 be the union of the first $\lfloor \frac{\chi(G)}{2} \rfloor$ color classes, and let V_2 be the remaining nodes of G. Our first inequality is $\sum_{v_i \in V_1} x_i \geq \frac{1}{3}$, and it separates the white nodes in two subgraphs, each with about half the chromatic number. Continuing the same way we arrive at nodes that contain white nodes that are independent, plus certain black nodes; these can be separated with one more internal node. The total depth is thus $\lceil \log_2 \chi((G) + 1) \rceil + 1$. □

To prove Theorems 1.1 and 1.2 from the lemmata, we now only need the following result of Feige and Kilian [8], building on earlier results of Lund and Yannakakis [15] and Fürer [9]:

THEOREM 2.3. *Unless NP=ZPP, no polynomial-time algorithm for approximating the chromatic number of a graph with n nodes can have an approximation ratio better than $n^{1-\epsilon}$, for a fixed $\epsilon > 0$ and large enough n.*

In other words, for a given efficient algorithm and $\epsilon > 0$, there are graphs with chromatic number n^ϵ, such that the algorithm cannot find a coloring better than $n^{1-\epsilon}$. Theorem 1.1 then follows from Lemma 2.1 and Theorem 2.3, and Theorem 1.2 follows from Lemma 2.2 and Theorem 2.3.

3. Single linear decisions. In this section we point out aspects of the difficulty of classifier optimization which hold even in the case in which W and B are *separable*, that is, *there is a single linear inequality that separates W from B* (in other words, the optimum classifying tree has just one internal node). In this case we are interested in minimizing *the number of variables that are actually needed in the decision node.*

Naturally, the interesting classification problems are not linearly separable; however, the separable case is practically interesting because it arises when we introduce "extra variables" to make classification possible. For example, one may introduce low-degree monomials (products of variables) or radial basis functions (simple functions of the distance from a point) [11, 13], and then construct a linear decision tree treating the outputs of these functions as new variables. Or one could even allow more costly special-purpose classifying heuristics, and also treat their outputs as variables. It is clear that any disjoint finite sets W and B may be separated given enough such extra functions, so the real question is how to minimize their number and cost. Besides the obvious consideration of computational efficiency, by the principle of *Occam's razor* we expect that optimal classifiers of this sort are in some sense "better-quality" classifiers.

We wish thus to solve the following problem: We are given two point sets $W, B \subseteq \Re^n$, that we know are separable by a single hyperplane. We are asked to find the hyperplane $\sum_{i=1}^n a_i x_i \geq b$ that separates W from B, and such that $|\{i : a_i \neq 0\}|$ is minimized. In another version (better suited for modeling the case of extra functions), the first $m < n$ variables are *free*, and we wish to minimize $|\{i > m : a_i \neq 0\}|$.

We next make a very useful simplification: We assume that $B = \{0\}$ (that is, there is only one black point, the origin): Given any classification problem W, B we can transform it into an equivalent classification problem $W - B, \{0\}$ where $W - B = \{w - b : w \in W \text{ and } b \in B\}$ is the *Minkowski difference.* Thus, we seek the hyperplane that separates a given point-set W from the origin and has the smallest number of nonzero coefficients (respectively, excluding the coefficients of the first m variables). We call these problems the *smallest separating inequality problem*, and its *version with free variables.*

Both versions of this problem are easily seen to be NP-complete. In this section we point out their high *parameterized complexity.* In [3, 6] a theory of parameterized complexity has been initiated. The issue is whether a minimization problem of the form "given instance x and integer parameter k, is the optimum k or less?" can be solved in time, say $O(n^p)$, where n is the size of the input x, and the hidden constants (but not p) may depend on k. For some problems, such as bandwidth and node cover, such algorithms are possible; for others, no such algorithms are known. These latter problems classify into a hierarchy of classes, denoted $W[1], W[2], \ldots$, plus an ultimate class $W[P]$. Hardness of a problem (via "parameterized reductions" appropriate for these problems, see [3]) for such a class is evidence that the problem does not have a polynomial algorithm even when the parameter is severely bounded. The higher the class, the more devastating the evidence of intractability.

THEOREM 3.1. *The smallest separating inequality problem is hard for $W[2]$, and its version with free variables is hard for $W[P]$.*

Proof. For $W[2]$-hardness we shall reduce the $W[2]$-complete *hitting set problem* [1, 12] to the minimum separating hyperplane problem. In the hitting set problem we are given a family $F = \{S_1, \ldots, S_k\}$ of subsets of some set $\{1, 2, \ldots, n\}$, and a parameter p, and we are asked to determine whether there is a set H, $|H| \leq p$, such that $H \cap S_i \neq \emptyset$ for all i. From F we construct a set of points $W = \{w_1, \ldots, w_k\} \subseteq \Re^n$, where w_i is the characteristic vector of S_i. Let $\sum_{i=1}^{n} a_i x_i = 1$ be a hyperplane separating W from the origin, and let $H = \{i : a_i \neq 0\}$. If $H \cap S_i = \emptyset$ for some i, then the hyperplane fails to separate w_i from the origin, and hence the nonzero coordinates of the hyperplane must be a hitting set. Conversely, for any hitting set H, the hyperplane $\sum_{i \in H} x_i = \frac{1}{2}$ separates W from the origin. This completes the proof of the first part.

For the second part, we shall reduce to the version of the problem with free variables the $W[P]$-complete *minimum monotone circuit value problem* [7]. In it we are given a monotone circuit, and a parameter k, and we wish to determine whether there is an input vector with k or fewer 1's that makes the output of the circuit 1. Given such a circuit with n gates, of which all but the first m are input gates, we construct the following point set W in $\Re^n$: If i is the output gate, we add to W the point $-e_i$ —recall that e_i is the unit vector in the ith coordinate. If i is an OR gate with inputs j and ℓ, then we add to W the point $e_i - e_j - e_\ell$. If i is an AND gate with inputs j and ℓ, then we add to W the points $e_i - e_j$ and $e_i - e_\ell$. This completes the construction. It is not very hard to argue that there is a hyperplane separating W from the origin with k or fewer nonzero coefficients in its last $n - m$ coordinates, if and only if the given circuit has a satisfying truth assignment with k or fewer positive inputs. □

Acknowledgments. We wish to thank Mihalis Yannakakis for an interesting discussion on this problem.

REFERENCES

[1] G. Ausiello, A. D'Atri, and M. Protasi, *Structure preserving reductions among convex optimization problems*, Journal of Computer and System Sciences, 21 (1980), pp. 136–153.

[2] A. L. Blum and R. L. Rivest, *Training a 3-node neural network is NP-complete*, Neural Networks, 5 (1992), pp. 117–127.

[3] H. L. Bodlaender, M. R. Fellows, and M. T. Hallett, *Beyond NP-completeness for problems of bounded width: Hardness for the W hierarchy*, in 26th Annual ACM Symposium on Theory of Computing (STOC), 1994, pp. 449–458.

[4] B. E. Boser, I. M. Guyon, and V. N. Vapnik, *A training algorithm for optimal margin classifiers*, in Proc. of the 5th Annual ACM Workshop on Computational Learning Theory (COLT), 1992, pp. 144–52.

[5] L. Breiman, J. J. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*, Wadsworth, 1984.

[6] L. Cai, J. Chen, R. Downey, and M. Fellows, *On the structure of parameterized problems in NP*, Information and Computation, 123 (1995), pp. 38–49. Preliminary version in STACS'94.

[7] R. G. Downey, M. R. Fellows, B. M. Kapron, M. T. Hallett, and H. T. Wareham, *The parameterized complexity of some problems in logic and linguistics*, in 3rd Intl. Symposium on Logical Foundations of Computer Science, Lecture Notes in Computer Science, vol. 813, Springer-Verlag, 1994, pp. 89–100.

[8] U. Feige and J. Kilian, *Zero knowledge and the chromatic number*, in Proc. 11th Ann. IEEE Conf. on Computational Complexity (CCC), 1996, pp. 278–287.

[9] M. Fürer, *Improved hardness results for approximating the chromatic number*, in Proc. 36th Annual IEEE Symposium on Foundations of Computer Science (FOCS), Milwaukee, Wisconsin, 1995, pp. 414–421.

[10] M. T. Goodrich, V. Mirelli, M. Orletsky, and J. Salowe, *Decision tree construction in fixed dimensions: Being global is hard but local greed is good*, Tech. Report TR-95-1, Johns Hopkins Univ., Dept. of Computer Science, May 1995.

[11] I. Guyon, B. Boser, and V. Vapnik, *Automatic capacity tuning of very large VC-dimension classifiers*, in Advances in Neural Information Processing Systems, S. J. Hanson, J. D. Cowan, and C. L. Giles, eds., vol. 5, Morgan Kaufmann, 1993, pp. 147–155.

[12] M. T. Hallett and H. T. Wareham, *A compendium of parameterized complexity results*, SIGACT News (ACM Special Interest Group on Automata and Computability Theory), 25 (1994). Also online at `ftp://ftp.csc.uvic.ca/pub/W_hierarchy/`.

[13] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Macmillan College Publishing, 1994.

[14] D. Heath, S. Kasif, and S. Salzberg, *Learning oblique decision trees*, in Proc. 13th International Joint Conference on Artificial Intelligence, Morgan Kaufmann, 1993, pp. 1002–1007. Chambery, France.

[15] C. Lund and M. Yannakakis, *On the hardness of approximating minimization problems*, Journal of the ACM, 41 (1994), pp. 960–981. Preliminary version in STOC'93.

[16] N. Megiddo, *On the complexity of polyhedral separability*, Discrete Computational Geometry, 3 (1988), pp. 325–337.

[17] S. K. Murthy, S. Kasif, and S. Salzburg, *A system for induction of oblique decision trees*, Journal of Artificial Intelligence Research, 2 (1994), pp. 1–33.