The Minimum Feature Set Problem

Kevin S. Van Horn and Tony Martinez
Computer Science Department
Brigham Young University
Provo, UT

This paper appeared in *Neural Networks* 7 (1994), no. 3, pp. 491–494.

The Minimum Feature Set Problem

Kevin S. Van Horn and Tony Martinez
Computer Science Department
Brigham Young University
Provo, UT

Acknowledgments: We wish to acknowledge the support of Wordperfect and Novell in this research, and thank the referees for their helpful comments.

Send reprint requests to: Kevin S. Van Horn, 3361 TMCB, Computer Science Dept., Brigham Young University, Provo, UT 84602. Phone: (801) 378-6914.

Running title: Minimum Feature Set.

The Minimum Feature Set Problem

Abstract

One approach to improving the generalization power of a neural net is to try to minimize the number of non-zero weights used. We examine two issues relevant to this approach, for single-layer nets. First we bound the VC dimension of the set of linear-threshold functions that have non-zero weights for at most s of n inputs. Second, we show that the problem of minimizing the number of non-zero input weights used (without misclassifying training examples) is both NP-hard and difficult to approximate.

Keywords: linear threshold, minimization, learning, complexity, approximation algorithm.

1 Introduction

Consider the task of learning from examples an (unknown) linear threshold function f of n real-valued features. Theoretical results (Blumer, Ehrenfeucht, Haussler & Warmuth, 1989) give an upper bound on the number of examples needed for learning that is linear in the *VC dimension* of the hypothesis space used, which in this case is $n+1$. If f depends on only a small subset of the features, this argues for removing the superfluous features; but often it is not known beforehand on which specific features f depends. Thus there has long been interest in learning algorithms which minimize the number of features actually used (i.e. the number of non-zero input weights), while still correctly classifying all the training examples. (Note that Littlestone (1988) gives an alternative approach to the problem of many unnecessary features, for a restricted class of linear threshold functions with Boolean inputs.)

In section 2 we quantify the benefits of minimizing the number of features used, by giving an upper bound on the VC dimension of the set of linear threshold functions with n inputs and at most s non-zero input weights. For $s \ll n$ this bound is much less than $n + 1$. This VC dimension bound improves on the bound that follows from the (more general) results of Baum and Haussler (1989).

In section 3 we examine the time-complexity of this feature-minimization problem, which we call MINIMUM FEATURE SET (MIN FS). We show that MIN FS is both NP-hard and difficult to approximate. In particular, we show the following:

- No polynomial-time algorithm can approximate MIN FS to within a constant factor, unless P = NP.
- No polynomial-time algorithm can approximate MIN FS to within an $o(\log m)$ factor, where m is the number of training examples, unless NP $\subseteq$ DTIME$[n^{\log \log n}]$.

2 Benefits of minimization

2.1 Preliminaries

We begin with a quick review of some notions used in the PAC (probably approximately correct) model of learning. The definitions and theorems of this subsection are from Blumer et al. (1989).

Definition 1 *A* hypothesis space *H over a set X is a set of $\{0,1\}$-valued functions with domain X. We say that a hypothesis $h \in H$ is* consistent *with a set E of training examples if $h(x) = y$ for all examples $(x, y) \in E$. We say that a learning algorithm* uses H consistently *if it returns a hypothesis $h \in H$ that is consistent with the set of training examples given it, whenever such an h exists.*

Definition 2 *Let H be a hypothesis space over a set X. For any finite $I \subseteq X$ we define*

$$\Pi_H(I) = \{h \cap I : h \in H\},$$

where we regard the elements of H as subsets of X. For any integer m, $1 \leq m \leq |X|$, we define

$$\Pi_H(m) = \max_{I \subseteq X, |I|=m} |\Pi_H(I)|.$$

$\Pi_H(m)$ is just the maximum number of ways a set of m elements of X can be dichotomized by elements of H.

Note that $\Pi_H(m) \leq 2^m$, since an m-element set has only 2^m distinct subsets.

Definition 3 *Let H be a hypothesis space over a set X. The* Vapnik-Chervonenkis dimension *of H, denoted* VCdim(H), *is the largest integer m s.t. $\Pi_H(m) = 2^m$. I.e.,* VCdim(H) *is the cardinality of the largest subset of X which can be arbitrarily dichotomized by elements of H.*

In general, VCdim(H) and $\Pi_H(m)$ are related as follows:

Theorem 1 *If* VCdim(H) $= d$ *and $m \geq d \geq 1$, then $\Pi_H(m) \leq (em/d)^d$, where e is the base of the natural logarithm.*

The VC dimension is important in computing upper bounds on a learning algorithm's sample complexity (a measure of the number of examples needed for learning). For learning algorithms that use a hypothesis space H consistently, these bounds are linear in VCdim(H) (Blumer et al., 1989).

2.2 Effect of minimization on VC dimension

Let H be the set of linear threshold functions on n real-valued features. We will identify $h \in H$ with its defining weight vector $\mathbf{w}$, i.e. the vector $\mathbf{w} \in \mathcal{Q}^{n+1}$ s.t.

$$h(\mathbf{x}) = \begin{cases} 1 & \text{if } w_{n+1} + \sum_{i=1}^{n} w_i x_i > 0 \\ 0 & \text{if } w_{n+1} + \sum_{i=1}^{n} w_i x_i < 0. \end{cases}$$

($\mathcal{Q}$ is the set of rational numbers.) We have VCdim(H) $= n+1$ (Blumer et al., 1989).

Definition 4 *The* cost *of $\mathbf{w} \in \mathcal{Q}^{n+1}$ (or the corresponding $h \in H$) is the number of nonzero weights other than the bias weight w_{n+1}, i.e. $|\{i : w_i \neq 0, 1 \leq i \leq n\}|$.*

Definition 5 *Let $S \subseteq \mathcal{N} \stackrel{\text{def}}{=} \{1, \ldots, n\}$ and $1 \leq s \leq n$; then we define*

- *$H[S]$ is the set of $h \in H$ which have nonzero weights (other than the bias) only for the features in S, and*

- *$H[s]$ is the set of $h \in H$ whose cost is at most s.*

Suppose a learning algorithm $\mathcal{A}$ uses H consistently and also minimizes the cost of the hypothesis it returns. If s is the cost of the function being learned, then $\mathcal{A}$ also uses $H[s]$ consistently, and we have an upper bound on sample complexity that is linear in $\mathrm{VCdim}(H[s])$. Even if $\mathcal{A}$ only does approximate minimization, the notion of an *effective hypothesis space* (Blumer et al., 1989) can be used to obtain sample-complexity bounds that use $\mathrm{VCdim}(H[s])$ rather than $\mathrm{VCdim}(H)$. Thus it is important to have good bounds on the former.

Theorem 2 *Let $1 \leq s < n$. Then $\mathrm{VCdim}(H[s]) < 2.71(s+1)\log_2(en/(s+1))$.*

Proof. We can assume that $s+1 \leq n/2$. If not, i.e. $s+1 > n/2$, then $n \geq s+1 \geq (n+1)/2$ (since s and n are integers), and writing lg for $\log_2$ we have

$$2.71(s+1)\lg\left(\frac{en}{s+1}\right) \geq 2.71(s+1)\lg e \geq \frac{2.71}{2}(n+1)\lg e > n+1.$$

But since $H[s] \subseteq H$ we have $\mathrm{VCdim}(H[s]) \leq \mathrm{VCdim}(H) = n+1$, which is less than our bound.

We now compute $\Pi_{H[s]}(m)$. The set of linear threshold functions of s inputs has VC dimension $s+1$. Thus $\mathrm{VCdim}(H[S]) = s+1$ for all $S \subseteq \mathcal{N}$, $|S| = s$, and by Theorem 1 this gives $\Pi_{H[S]}(m) \leq (em/(s+1))^{s+1}$ for all $m \geq s+1$. Using this inequality we obtain

$$\Pi_{H[s]}(m) \leq \sum_{|S|=s} \Pi_{H[S]}(m) \leq \binom{n}{s}(em/(s+1))^{s+1}.$$

By the assumption that $s+1 \leq n/2$ we have $\binom{n}{s+1} > \binom{n}{s}$. A variant of Stirling's formula states that $\binom{j}{k} \leq (ej/k)^k$, for all j and k (Cormen, Leiserson & Rivest, 1990, p. 102). Combining these we obtain

$$\Pi_{H[s]}(m) < (en/(s+1))^{s+1}(em/(s+1))^{s+1} \text{ for all } m \geq s+1.$$

Let $x_0 \stackrel{\text{def}}{=} \lg(en/(s+1))$ and $m_0 \stackrel{\text{def}}{=} 2.71(s+1)x_0$. To prove the theorem it suffices to show that $\Pi_{H[s]}(m_0) < 2^{m_0}$, by the definition of the VC dimension. This inequality will hold if $m_0 - \lg(\Pi_{H[s]}(m_0)) > 0$. Dividing this inequality by $s+1$, and using the fact that $m_0 \geq s+1$, we find

$$\frac{m_0 - \lg(\Pi_{H[s]}(m_0))}{s+1} > \frac{m_0}{s+1} - x_0 - \lg\left(\frac{em_0}{s+1}\right) = 1.71x_0 - \lg(2.71ex_0)$$

Thus it suffices to show that $f(x_0) \geq 0$, where $f(x) \stackrel{\text{def}}{=} 1.71x - \lg(2.71ex)$. We have $f'(x) = 1.71 - (x \ln 2)^{-1}$, hence $f(x)$ is non-decreasing for all $x \geq x_{min} \stackrel{\text{def}}{=} (1.71 \ln 2)^{-1}$.

Thus it suffices to show that $f(x) \geq 0$ for some $x_{min} \leq x \leq x_0$. Let us choose $x = \lg(2e)$. Since $s + 1 \leq n/2$ we have $x_0 \geq \lg(2e)$. We also have

$$\lg(2e) = \ln(2e)/\ln 2 > 1.71^{-1}/\ln 2 = x_{min}.$$

Finally, by computation we find $f(\lg(2e)) \approx 7.55 \times 10^{-3} > 0$. □

The above VC dimension bound is a significant improvement over $n+1$ when $s \ll n$. It is also an improvement over the $\Theta(s \log(sn))$ bound that can be obtained from Corollary 5 of Baum and Haussler (1989), which is a more general result applicable to multi-layered neural nets.

3 The complexity of minimization

We can formally state the minimization problem we have been discussing as follows:

Definition 6 MINIMUM FEATURE SET (MIN FS)
Instance: *Integers $m \geq 1$, $n \geq 1$, and an $m \times (n+1)$ matrix $\mathbf{A}$ of rational numbers s.t. $a_{i,n+1} \in \{1, -1\}$ for all i.*
Problem: *Find a $\mathbf{w} \in \mathcal{Q}^{n+1}$ satisfying $\mathbf{Aw} > 0$, of minimum cost. (Recall that the cost of $\mathbf{w}$ is the number of nonzero w_i, $i \neq n+1$.)*

In the above, m is the number of training examples and n is the number of features. $\mathbf{A}_i$, the i-th row of the matrix $\mathbf{A}$, is obtained from the i-th training example $(\mathbf{x}_i, y_i)$ as follows: first, augment $\mathbf{x}_i$ with 1 to obtain an $(n+1)$-element vector $\alpha(\mathbf{x}_i)$; then, if it is a negative example ($y_i = 0$), negate this vector. The requirement that $\mathbf{Aw} > 0$ is equivalent to $\mathbf{A}_i\mathbf{w} > 0$ for all i, so this gives us $\alpha(\mathbf{x}_i) \cdot \mathbf{w} > 0$ for positive examples and $\alpha(\mathbf{x}_i) \cdot \mathbf{w} < 0$ for negative examples.

The requirement that $a_{i,n+1} \in \{1, -1\}$ arises from the above construction, in which the example vectors were augmented with 1 to provide for the bias term; thus $a_{i,n+1} = 1$ for positive examples and $a_{i,n+1} = -1$ for negative examples.

We now give some standard definitions that will be needed in our investigation of the difficulty of solving or approximating MIN FS.

Definition 7 *A* minimization *problem $\mathcal{M}$ has the following form, for some pair of predicates P and Q and integer-valued* cost *function κ: Given x satisfying $P(x)$ (a problem* instance*), find some y satisfying $Q(x, y)$ (a* solution *for x) that minimizes $\kappa(x, y)$.*

Definition 8 *A* polynomial-time approximation algorithm (PTAA) *$\mathcal{A}$ for a minimization problem $\mathcal{M}$ is a polynomial-time algorithm which takes as input some instance x satisfying $P(x)$ and outputs a y satisfying $Q(x, y)$. We say that $\mathcal{A}$* approximates *$\mathcal{M}$* to within a factor *$\phi(x)$ if, additionally, $\kappa(x, y) \leq \phi(x)\kappa(x, z)$ for any z such that $Q(x, z)$. We say that $\mathcal{M}$* can be approximated to within a factor *$\phi(x)$ if there is a PTAA that approximates it to within a factor $\phi(x)$.*

Definition 9 *A* cost-preserving *polynomial transformation from minimization problem $\mathcal{M}$ to minimization problem $\mathcal{M}'$ is a pair of functions (t_1, t_2) with the following properties, where x is taken to be an instance of $\mathcal{M}$:*

1. *t_1 maps instances of $\mathcal{M}$ to instances of $\mathcal{M}'$. t_2 maps pairs (y, x), where y is a solution for $t_1(x)$, to solutions for x.*
2. *t_1 and t_2 are both computable in polynomial time.*
3. *If x has a solution of cost k, then $t_1(x)$ has a solution of cost at most k.*
4. *If y is a solution for $t_1(x)$ of cost k, then $t_2(y, x)$ has cost at most k.*

Suppose there exist both a cost-preserving polynomial transformation from $\mathcal{M}$ to $\mathcal{M}'$ and a PTAA that approximates $\mathcal{M}'$ to within a factor $\phi(x')$. Then there is a PTAA that approximates $\mathcal{M}$ to within a factor $\phi(t_1(x))$: compute $x' = t_1(x)$, run the PTAA for $\mathcal{M}'$ on x' to get y, and output $t_2(y, x)$. Thus any limits on the approximability of $\mathcal{M}$ give corresponding limits on the approximability of $\mathcal{M}'$. We will show that MIN FS is both NP-hard and difficult to approximate via a transformation from MIN SET COVER.

Definition 10 MINIMUM SET COVER (MIN SC)
Instance: *A finite set S and a collection C of subsets of S.*
Problem: *Find a cover of S (a set $C' \subseteq C$ s.t. $\bigcup C' = S$) of minimum cardinality.*

Theorem 3 *There is a cost-preserving polynomial transformation from MIN SC to MIN FS in which instances (S, C) of MIN SC are mapped to instances $(m, n, \mathbf{A})$ of MIN FS with $m = |S| + 1$.*

Proof. In the following we shall use the notation (Φ), where Φ is a logical formula, for (if Φ then 1 else 0). This notation is due to Graham, Knuth and Patashnik (1989).

Let (S, C) be an instance of MIN SC. Define $m = |S| + 1$ and $n = |C|$. Enumerate the elements of S as s_1, s_2, ..., s_{m-1} and the elements of C as c_1, c_2, ..., c_n. Define $t_1(S, C) = (m, n, \mathbf{A})$, where $\mathbf{A}$ is an $m \times (n + 1)$ matrix constructed from the column vectors of 1's and 0's corresponding to each c_j, as follows:

c_1	c_2	$\cdots$	c_n	$\vdots$ -1 $\vdots$
$\cdots 0 \cdots$				1

(Thus, $a_{ij} = (s_i \in c_j)$ for $i < m$ and $j \leq n$.)

Note that the inequality $\mathbf{A}\mathbf{w} > 0$ is equivalent to the set of inequalities $\mathbf{A}_i\mathbf{w} > 0$, $1 \leq i \leq m$. This set of inequalities is in turn equivalent to

$$\sum_{j \leq n} (s_i \in c_j) w_j > w_{n+1} > 0, \quad 1 \leq i < m \tag{1}$$

Suppose that (S, C) has a solution of cost k, i.e. there exists some $C' \subseteq C$ which is a cover of S, and $|C'| = k$. Define the vector $\mathbf{w}$ by

$$w_j = \begin{cases} (c_j \in C') & \text{if } j \leq n \\ 0.5 & \text{if } j = n+1. \end{cases}$$

Note that $\mathbf{w}$ has cost k and $w_{n+1} > 0$. Using the fact that C' is a cover of S we have also that, for $1 \leq i < m$,

$$\sum_{j \leq n} (s_i \in c_j) w_j = \sum_{j \leq n} (s_i \in c_j)(c_j \in C') > 1 > w_{n+1},$$

so $\mathbf{w}$ satisfies (1). Thus we have shown that $t_1(S, C) = (m, n, \mathbf{A})$ has a solution of cost k.

We now consider the reverse mapping. For all $\mathbf{w} \in \mathcal{Q}^{n+1}$, define

$$t_2(\mathbf{w}, S, C) = \{c_j : w_j > 0, 1 \leq j \leq n\}.$$

Suppose that $\mathbf{w}$ is a solution for $t_1(S, C) = (m, n, \mathbf{A})$, of cost k. Let $C' = t_2(\mathbf{w}, S, C)$. It is clear from the definition of t_2 that $C' \subseteq C$ and $|C'| \leq k$. We have furthermore that $\mathbf{A}\mathbf{w} > 0$ and hence $\mathbf{w}$ satisfies the inequalities (1). Thus for each i there is some j s.t. $s_i \in c_j$ and $w_j > 0$. But $w_j > 0$ implies $c_j \in C'$, so C' is a cover for S. Thus we have shown that $\mathbf{w}$ is a solution for (S, C) of cost at most k. □

Corollary 4 *MIN FS is NP-hard. In addition, we have the following approximability limits:*

1. *For every constant $c \geq 1$, MIN FS cannot be approximated to within a factor c, unless* P = NP.
2. *MIN FS cannot be approximated to within an $o(\log m)$ factor, unless* NP $\subseteq$ DTIME$[n^{\log \log n}]$.

Proof. MIN SC is NP-hard, hence by Theorem 3, so is MIN FS.

From Theorem 3, if MIN FS can be approximated to within some constant factor c, so can MIN SC. But Bellare, Goldwasser, Lund and Russel (1993) have shown that MIN SC cannot be approximated to within a constant factor unless P = NP.

Suppose that MIN FS can be approximated to within a factor $g(m, n, \mathbf{A}) = f(m)$, where $f(m) = o(\log m)$. Then MIN SC can be approximated to within a factor $g(t_1(S, C)) = f(|S| + 1) = o(\log(|S| + 1)) = o(\log |S|)$. But Bellare et al. have also

shown that MIN SC cannot be approximated to within an $o(\log |S|)$ factor unless NP $\subseteq$ DTIME$[n^{\log \log n}]$. □

Haussler (1988) looks at the problem of finding a pure conjunctive concept consistent with a set of examples, with the minimum number of conjuncts, and shows that it is NP-hard via a reduction from MIN SC. His reduction is similar to ours, and defines a cost-preserving polynomial transformation that produces $|S| + 1$ examples. Thus the conclusions of Corollary 4 also apply to Haussler's problem.

We note in closing that Corollary 4 should *not* be taken as evidence against the possibility of obtaining improved sample-complexity bounds by approximating MIN FS. Lemma 5.6 of Haussler (1988) and Theorem 3.2.1 of Blumer et al. (1989) can be used to establish improved upper bounds on sample complexity when the approximation factor grows slowly with the number of examples.

References

Baum, E. B., & Haussler, D. (1989). What size net gives valid generalizations? *Neural Computation*, **1**, 151–160.

Bellare, M., Goldwasser, S., Lund, C., & Russell, A. (1993). Efficient probabilistically checkable proofs and applications to approximation. *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*, 294–304.

Blumer, A., Ehrenfeucht, A., Haussler, D., & Warmuth, M. K. (1989). Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, **36**, 929–965.

Cormen, T. H., Leiserson, C. E., & Rivest, R (1990). **Introduction to algorithms**. Cambridge, MA: MIT Press.

Graham, R. L., Knuth, D. E., & Patashnik, O. (1989). **Concrete mathematics: a foundation for computer science**. Reading, MA: Addison-Wesley.

Haussler, D. (1988). Quantifying inductive bias: AI learning algorithms and Valiant's learning framework. *Artificial Intelligence*, **36**, 177–221.

Littlestone, N. (1988). Learning quickly when irrelevant attributes abound: a new linear-threshold algorithm. *Machine Learning*, **2**, 285–318.