On Feature Selection: Learning with Exponentially many Irrelevant Features as Training Examples

Andrew Y. Ng
Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139
ayn@ai.mit.edu

Abstract

We consider feature selection in the "wrapper" model of feature selection. This typically involves an NP-hard optimization problem that is approximated by heuristic search for a "good" feature subset. First considering the idealization where this optimization is performed exactly, we give a rigorous bound for generalization error under feature selection. The search heuristics typically used are then immediately seen as trying to achieve the error given in our bounds, and succeeding to the extent that they succeed in solving the optimization. The bound suggests that, in the presence of many "irrelevant" features, the main source of error in wrapper model feature selection is from "overfitting" hold-out or cross-validation data. This motivates a new algorithm that, again under the idealization of performing search exactly, has sample complexity (and error) that grows *logarithmically* in the number of "irrelevant" features – which means it can tolerate having a number of "irrelevant" features *exponential* in the number of training examples – and search heuristics are again seen to be directly trying to reach this bound. Experimental results on a problem using simulated data show the new algorithm having much higher tolerance to irrelevant features than the standard wrapper model. Lastly, we also discuss ramifications that sample complexity logarithmic in the number of irrelevant features might have for feature design in actual applications of learning.

1 Introduction

In recent years, Feature Selection for classification and regression has been enjoying increasing interest in the Machine Learning community. Impressive performance gains have been reported by numerous authors, and numerous feature subset search heuristics have been proposed. (The literature is too wide to survey here, but see [Langley, 1994] and [Miller, 1990] for overviews.) In view of these significant empirical successes, one central question is: What theoretical justification is there for feature selection? For example, in parametric function approximation schemes such as linear regression, it is often the case that excluding a feature is mathematically identical to setting the coefficient(s) associated with that feature to 0. As feature selection typically runs a risk of misidentifying the "irrelevant" features, why then is it apparently often superior to try to estimate which features are "irrelevant" and set their coefficients to 0, rather than leave them and use the estimated coefficients for these features (which will typically be near 0 anyway)? The theoretical results in this paper will address this question.

Since feature selection attempts to eliminate "irrelevant" features, another central question is: How does the performance of feature selection scale with the number of irrelevant features? The Winnow algorithm of Littlestone for learning Boolean monomials, or more generally also k-DNF formulae and r-of-k threshold functions (over boolean inputs), from noiseless data enjoys worst-case loss logarithmic in the number of irrelevant features [Littlestone, 1988]. Likewise, the EG algorithm for linear regression with quadratic error also has such loss (and indeed sample complexity) that grows logarithmically in the number of irrelevant features [Kivinen and Warmuth, 1994]. For learning from noiseless data, of a representation of a boolean concept (over boolean inputs), Almuallim and Dietterich have also shown that an algorithm that finds the smallest set of features consistent with the training data (such as by exhaustive enumeration) also enjoys loss logarithmic in the number of irrelevant features [Almuallim and Dietterich, 1994]. If it were true in general that feature selection makes sample complexity logarithmic in the number of irrelevant features (though possibly depending more heavily on the number of relevant features), then this would imply, for example, that *squaring* the number of features we have means needing only *twice* as much training data. This could have huge ramifications on the way features are designed for real-world applications. In this paper, we will show that, modulo computational and approximation issues, this ideal of *logarithmic* sample complexity in the number of irrelevant features – which of course means being able to handle *exponentially* many irrelevant features as training examples – can indeed be achieved with a new feature selection algorithm we propose.

Next, the notion of "relevance" is closely related to feature selection. Intuitively, one goal of feature selection is to eliminate all but a small set of "relevant" features, which are then given to an induction algorithm. However, there have been difficulties with a number of definitions of "relevance" [Kohavi and John, 1997], and we take the alternative view, which is quite similar in flavor to those in [Littlestone, 1988] and [Kivinen and Warmuth, 1994], of the goal of feature selection as this: If there exists a hypothesis that, using only a "small" number of features, gives good generalization error, then we want our classifier to achieve close to this level of performance with high probability. This will be made rigorous in subsequent sections, but note in particular that we make no claims towards excluding "irrelevant" features or including all the "relevant" features, so long as the particular set of selected features allows us to have performance close to that of using the "optimal" set of features. [1] In the remainder of this paper, we will use the terms "relevant" and "irrelevant" only when we expect them to be consistent with any reasonable definition of relevance.

Using the terminology introduced by [John et al., 1994], feature selection algorithms broadly fall into the "filter" and the "wrapper" models. The filter model relies on general characteristics of the training data to select some feature subset, doing so without reference to the learning algorithm. In the wrapper model, one generates sets of candidate features, runs them through the learning algorithm, and uses the performance of the resulting hypothesis to evaluate the feature set. While the wrapper model tends to be more computationally expensive, it also unsurprisingly tends to find feature sets better suited to the inductive biases of our learning algorithm, and tends to give superior performance [Langley, 1994]. In this paper, we study only the wrapper model of feature selection, and largely in the context of classification.

Our analysis is largely inspired by [Kearns, 1996], with our theoretical results heavily based on the techniques given there and those outlined in [Kearns et al., 1997]. We also rely heavily on tools from [Vapnik, 1982], that give a very general framework for bounding the deviation of training error from generalization error.

2 Preliminaries

2.1 Feature Selection

Let X be the fixed f-dimensional input space, where f is the number of features in the inputs we are provided. For simplicity, we also assume a fixed binary concept $c : X \longmapsto \{0, 1\}$. We are provided m training examples $S = \{x^i, y^i\}_{i=1}^m$, with each of the f-dimensional input vectors $x^i = [x_1^i \; x_2^i \; \ldots \; x_f^i]^T$ drawn *i.i.d.* from some fixed distribution D_X over X, and corresponding labels $y^i = c(x^i) \in \{0, 1\}$. In this development, we will also briefly consider the case where the labels are independently corrupted by noise with a noise rate $\eta \in [0, 0.5)$, so that $y^i = c(x^i)$ with probability $1 - \eta$, and $y^i = 1 - c(x^i)$ with probability η. Note that c may use all f features, but we hope that it can be approximated well (in the generalization-error sense, to be defined shortly) by a function that depends only on a small subset of the f features.

We will use uppercase F to denote sets of features, and use F_i to identify the i-th feature. For example, the feature set including the 1st, 4th and 10th features may be written $F = \{F_1, F_4, F_{10}\}$. For any input vector x, let $x|_F$ be x with all the features not in F eliminated; sometimes, we will call this "x restricted to F." Analogously, let $X|_F$ denote the input space X with all the dimensions/features not in F eliminated, and $S|_F$ be the data set S with each x^i replaced by $x^i|_F$. In a slight abuse of notation, if we have a hypothesis $h : X|_F \longmapsto \{0, 1\}$ defined only the subspace of features $X|_F$, we extend it to X in

[1] Aside from good generalization error, other goals of feature selection might be user-interpretability and parsimony of hypotheses for fast prediction. We will not address these goals in this paper.

the natural way (with h ignoring features not in F). Thus, for any hypothesis h, we can write the generalization error (with respect to uncorrupted data) as $\varepsilon(h) = \Pr_{x \in D_X}[h(x) \neq c(x)]$ (where the dependence of $\varepsilon(h)$ on D_X has been suppressed for notational brevity,) and the empirical error on a set of data S as $\hat{\varepsilon}_S(h) = \frac{1}{|S|}|\{(x,y) \in S | h(x) \neq y\}|$.

2.2 The wrapper model

In the wrapper model of feature selection suggested by [John et al., 1994], we are given a learning algorithm L that, for any set of features F, takes a training set $S|_F$, and outputs a hypothesis $h : X|_F \longmapsto \{0,1\}$. Given a training set S, an application of feature selection under this model might randomly split S into a training set S' of size $(1-\gamma)m$ and a hold-out set S'' of size γm, and perform a search for a set of features F so that when the learning algorithm is applied to S' restricted to F, the resulting hypothesis $h = L(S'|_F)$ has low empirical error $\hat{\varepsilon}_{S''}(h)$ on the hold-out data S''. Here, $\gamma \in [0,1]$, the fraction of S assigned to the hold-out set, is called the hold-out fraction. A more sophisticated application of feature selection may use n-fold or leave-one-out cross validation rather than hold-out. But as they asymptotically yield at best small-constant improvements over using hold-out and as leave-one-out is at worst little better than training error in estimating generalization error, while rendering the algorithm's performance much less tractable to analysis [Kearns and Ron, 1997], we will not explicitly consider them here, though we believe our results will be suggestive of the performance of these schemes as well.

For any given learning algorithm L, the optimal way to perform feature selection is intimately related to the inductive biases of L. For example, if L is "sufficiently clever" about doing its own feature selection, then one would simply give it S unrestricted to any feature subset, and allow it to select its own features. For this analysis, therefore, we make the (rather strong) assumption that given a particular data set $S|_F$, L chooses the hypothesis h from some class of hypotheses (shortly to be formalized) so as to minimize training error. This closely ties in with the learning framework studied by [Vapnik, 1982], and is also used in [Kearns, 1996] and [Kearns et al., 1997] in proving bounds on generalization error. We believe it to be a very natural model, and that it is a rich enough class of learning algorithms to merit detailed study. (But also see [Kearns et al., 1997] for comments regarding relations to learning algorithms that do not exactly do this; for example, it is not difficult to derive rigorous generalizations of all of our results if L manages to only approximately minimize training error.)

More formally, for any feature set F, we assume that we have a hypothesis class H_F, of hypotheses each with domain $X|_F$. But, with many induction algorithms, each feature is treated in a "similar" manner – for example, when $X = \mathcal{R}^f$, then for two feature sets F and F' of the same size, it makes intuitive sense to identify $X|_F$ and $X|_{F'}$ and therefore H_F and $H_{F'}$, as they are both sets of functions mapping from $\mathcal{R}^{|F|}$ to $\{0,1\}$. For simplicity, let us further make the assumption that the hypothesis class H_F depends on F only through $|F|$, and let H_r be our set of functions with domain X restricted to any set of r features. (This assumption is not really necessary, but it greatly eases our notational burden, and leaving out the assumption does not gain much in terms of theoretical results.) It will always be clear from context which particular set F of features $h \in H_{|F|}$ takes as input. Note also that we have assumed that there is some "uniform" way of handling all features, whether they are discrete/continuous, have different ranges, etc.. For simplicity, one may wish to think of the particular case where all features are real numbers for the remainder of this paper. In this notation then, our previous assumption of error minimization is that when L is given $S|_F$, it outputs the hypothesis $h \in H_F$ (where H_F is identified with $H_{|F|}$) that minimizes training error on $S|_F$. For the remainder of this paper, we will implicitly assume L meets these two assumptions – that it treats features "uniformly," and that it minimizes training error over $H_{|F|}$.

One more definition we need is to let r_{VC} be the Vapnik-Chervonenkis dimension [Vapnik and Chervonenkis, 1971, Vapnik, 1982] of the hypothesis class H_r. Normally, we expect $0_{VC} < 1_{VC} < 2_{VC} < \cdots$, though this is not an assumption we use. For example, if H_r is the class of linear discriminant functions over $\mathcal{R}^r$, then $r_{VC} = r + 1$. We chose this notation so that, to specialize our ensuing bounds on generalization error to linear discriminant functions, which we later use in our experiments, r_{VC} may everywhere be replaced with r (or at least when $r > 0$).

Finally, to obtain the performance bounds, we wish to make statements of the form that "we will, with high probability, find a hypothesis with generalization error no worse than z more than the best hypothesis that uses r features." To formalize this, define the approximation rate function $\varepsilon_g(r)$ to be the *least* generalization error achievable by any hypothesis $h \in H_r$ using any set of r features. In general, we expect

$\varepsilon_g(1) \geq \varepsilon_g(2) \geq \cdots$, though this is also not an assumption we require (except briefly when we summarize our results in terms of sample complexity).

Thus, in the common instantiation of wrapper model feature selection, we search for a feature set F such that when L is applied to $S'|_F$, the resulting hypothesis has low empirical error on the hold-out set. (That is, $\hat{\varepsilon}_{S''}(L(S'|_F))$ is minimized.) Leaving aside details of the actual search, we will call this idealization the STANDARD-WRAP algorithm. Note that in performing the search, enumeration over all the 2^f possible feature sets is usually intractable, and there is no known algorithm for otherwise performing this optimization tractably. Indeed, the Feature Selection problem in general is NP-hard [Garey and Johnson, 1979], but much work over recent years has developed a large number of heuristics for performing this search efficiently. (Again, the literature is too wide to survey here, but examples include [Moore and Lee, 1994, Caruana and Frietag, 1994, Yang and Hoavar, 1997], and [Langley, 1994, Miller, 1990] include overviews.) In this development, we will, in the style of [Kearns, 1996], give bounds for generalization error when this optimization is performed exactly. Of course, the extent to which our bounds predict actual performance will in part depend on the extent to which the optimization algorithms succeed in performing this search on "real life" distributions of data. Alternatively, one can also view these bounds as what the heuristic search/approximation algorithms are (in a rigorous sense, to be discussed later) aspiring to do, with the bounds giving insight into how we might expect the algorithms to perform.

3 Main Results

The ensuing bounds are all given to hold "with high probability." We defer their more detailed versions to the full paper, but note that when we say "with high probability," we mean that the bound holds with at least probability $1-\delta$ for any $\delta > 0$, with constants that depend on δ (through an omitted $\log\frac{1}{\delta}$ term) hidden by the $O(\cdot)$ notation.

Bound for performance without feature selection

The Universal Estimate Rate bound of Vapnik and Chervonenkis [Vapnik and Chervonenkis, 1971, Vapnik, 1982] gives a bound on generalization error when learning using all f features without feature selection.

Theorem 1 (Vapnik and Chervonenkis, 1971) *With high probability, the generalization error of the hypothesis $\hat{h} = L(S)$, given by L applied to S (unrestricted to any feature subset), is bounded by:*

$$\varepsilon(\hat{h}) \leq \varepsilon_g(f) + O\left(\sqrt{\frac{f_{\mathrm{VC}}}{m}\left(\log\frac{m}{f_{\mathrm{VC}}}+1\right)}\right) \qquad (1)$$

Note this is a bound for learning from noiseless data; when the training data labels have independently been corrupted at some noise rate η, the second term in the bound becomes $O\left(\sqrt{\frac{f_{VC}}{(1-2\eta)^2 m}(\log\frac{m}{f_{VC}}+1)}\right)$.

Bound for performance of wrapper model

Applying the proof technique given in [Kearns, 1996] (used to bound the error of hold-out) to feature selection, we obtain the following theorem:

Theorem 2 *Given L, S, γ, the hypothesis $\hat{h}$ output by* STANDARD-WRAP, *given by $\hat{h} = L(S'|_{\hat{F}})$ where $\hat{F} = \mathrm{argmin}_F\, \hat{\varepsilon}_{S''}(L(S'|_F))$, will, with high probability, have generalization error bounded by*

$$\varepsilon(\hat{h}) \leq \min_{0\leq r\leq f}\left\{\varepsilon_g(r) + O\left(\sqrt{\frac{r_{\mathrm{VC}}}{(1-\gamma)m}\left(\log\frac{m}{r_{\mathrm{VC}}}+1\right)}\right)\right\} + O\left(\sqrt{\frac{f}{\gamma m}}\right) \qquad (2)$$

Proof (Sketch): The first square-root term is simply the universal estimation rate bound as before, that says that with high probability, the hypothesis obtained by applying L to $S'|_F$ for any fixed F with $|F| = r$ will give additional error no more than $O(\sqrt{\frac{r_{VC}}{(1-\gamma)m}(\log\frac{m}{r_{VC}}+1)})$. Following this, using a holdout-test set of size γm to test 2^f hypotheses will, by a standard Chernoff-bound argument, result with high probability in picking a hypothesis with generalization error no more than $O(\sqrt{\log(2^f)/\gamma m}) = O(\sqrt{f/\gamma m})$ higher. □

Again, this bound holds only when learning from noiseless data. Similar to Theorem 1, a generalization to learning from noisy data can be obtained by replacing all occurrences of m in any denominator term in the bound by $(1-2\eta)^2 m$, where η is the noise rate.

One important remark here is that the $O(\sqrt{f/\gamma m})$ term is a worst-case bound for evaluating 2^f hypotheses on the independent hold-out set S'' of size γm. Its increase with f reflects the fact that we are testing a set of hypotheses of size exponential in f, and that there is potential for "overfitting" the γm hold-out samples. (In the context of feature selection, the issue of overfitting of hold-out data was also raised by [Kohavi and Sommerfield, 1995]; see also [Ng, 1997] for a detailed discussion of overfitting of hold-out data in hypothesis selection.) But since this is a worst-case bound, it holds in particular for the "bad case" where all 2^f hypotheses are "very different" from each other. This is unlikely as they were trained on the same dataset S' and using only f distinct features. For at least some pathological hypothesis classes (that may, for example, include a set of hash-like basis function so that changing one feature's range dramatically changes the output hypotheses,) this is certainly possible; but for more "sensible" hypothesis classes, we might expect it to be possible to significantly tighten this bound. We have not managed to formalize this yet, but conjecture, based on the behavior of power-law decay learning curves, that the asymptotic behavior for "many" learning algorithms will be better modeled by replacing this last term in the bound by $\sqrt{f^\alpha/\gamma m}$ for some $\alpha \in (0,1]$. (A preliminary analysis suggests that under a (perhaps surprisingly large) range of formal modeling assumptions regarding how much hypotheses change when F is changed, the number of "significantly different" hypotheses does grow as $2^{O(f)}$, which would suggest $\alpha = 1$ behavior. On the other hand, there are certainly also some reasonable assumptions that would lead to $\alpha < 1$; and we defer a detailed discussion of this to the full paper.)

Bound for performance of new algorithm

For STANDARD-WRAP, the dependence on f of our bound on the error is $\sqrt{f/\gamma m}$ (or possibly $\sqrt{f^\alpha/\gamma m}$), and it comes from testing 2^f hypothesis on holdout-data. If $f \gg r_{VC}$ where r is the number of features needed to approximate the target concept well, this $\sqrt{f/\gamma m}$ will be the dominant term. Consider instead the following algorithm, which we call ORDERED-FS:

1. For each $0 \le r \le f$, find the hypothesis $\hat{h}_r$ that, of all the hypotheses using exactly r features, minimizes error on the training set S'. (This involves a search over all sets of r features.)
2. Evaluate all $f+1$ hypotheses $\{\hat{h}_r\}_{r=0}^{f}$ on the hold-out set S'', and pick the one with the smallest hold-out error.

Note that we are now testing only $O(f)$ hypotheses on the hold-out data, so the previous $\sqrt{f/\gamma m}$ term now becomes $\sqrt{(\log f)/\gamma m}$.

Theorem 3 *Given L, S, γ, the hypothesis $\hat{h}$ output by* ORDERED-FS *will, with high probability, have generalization error bounded by*

$$\varepsilon(\hat{h}) \le \min_{0\le r\le f}\left\{\varepsilon_g(r) + O\left(\sqrt{\frac{r_{\rm VC}}{(1-\gamma)m}\left(\log\frac{m}{r_{\rm VC}}+1\right)}\right) + O\left(\sqrt{\frac{r\log f}{(1-\gamma)m}}\right)\right\} + O\left(\sqrt{\frac{\log f}{\gamma m}}\right) \qquad (3)$$

Proof (Sketch): The first square-root term is simply the universal estimation rate bound as before, used to bound the additional error when training on any fixed feature set. For this to hold with probability $1-\delta$, there is also an additive $(1/m)\log(1/\delta)$ within the square-root. Now, for any fixed r, we want to uniformly bound the deviation of training error from generalization error for all $\binom{f}{r}$ hypotheses that use exactly r features. Taking a standard union bound (see [Vapnik, 1982]), we replace $(1/m)\log(1/\delta)$ with $(1/m)\log\left(\binom{f}{r}/\delta\right)$, which (noting $\log\binom{f}{r} \le r\log f$) gives the second term. Lastly, the third term comes, using a standard Chernoff-bound argument as before, from testing $O(f)$ hypotheses on the hold-out set of size γm. □

Notice that, similar to STANDARD-WRAP, we have not explicitly addressed the NP-hard search problem for the optimal (here in the minimum training error sense) set of r features, and actual implementations of ORDERED-FS will generally have to rely on heuristic search. But for now, let us beg this computational issue and treat it similarly to how we had treated STANDARD-WRAP, appealing to the same approximations/idealizations as before, and also mentioning that, in a rigorous sense to be discussed later, the extent to which an approximation algorithm can solve the optimization is exactly the extent to which its error bound will reach the bound we give here, which means that our bound can as before be interpreted, in a formal sense, as being exactly what a heuristic search implementation is trying to attain. (In considering heuristic search implementations, it is also worth mentioning that searching to minimize training error is probably often somewhat easier than searching to minimize hold-out error, which STANDARD-WRAP requires; for example, in linear regression, we have fast algorithms for simultaneously evaluating training error for all single-feature changes to a feature subset.) This bound is also easily generalized to learning from noisy examples (again by replacing all occurrences of m in any denominator term with $(1-2\eta)^2 m$).

In any case, the key point of this bound is then the following: The dependence of our bound on f is only *logarithmic* in f. It is also easy to see from the bound that the sample complexity m is also logarithmic in f. As discussed in the Introduction, this means that, from an information-theoretic point of view, one may *square* the number of features (for example by adding all cross-terms between all features), and expect to need only *twice* as much training data. We believe that this, if even only approximately realizable by search algorithms, may have tremendous consequences for feature design – that modulo computational expense, overly careful human design of features would often be unnecessary, so long as additional training data can be obtained reasonably cheaply.

To close this section, we informally restate our theoretical results in terms of upper bounds on sample complexity, if the target concept is well represented by some small number r^* of features. That is, we want the number m^* of examples required so that generalization error will be close to that of the optimal hypothesis that uses r^* features. (Slightly more formally, we want, for any fixed $\epsilon > 0$, that $\varepsilon(\hat{h}) < \varepsilon_g(r^*) + \epsilon$ with high probability, and where dependence of m^* on ϵ will again be hidden by the $O(\cdot)$ notation.) From the earlier theorems, it is not difficult to derive the following (upper bounds on) sample complexity:

algorithm	m^*
No feature selection	$O(f_{VC})$
STANDARD-WRAP	$O(r^*_{VC} + f^{\alpha})$, $\alpha \leq 1$
ORDERED-FS	$O(r^*_{VC} + r^* \log f)$

Particularly if r_{VC} grows superlinearly in r, we easily see STANDARD-WRAP has a significantly smaller sample complexity than not performing feature selection if $r^* \ll f$. This appears to us to be rather strong theoretical justification for performing feature selection, thereby answering the question of "why feature selection" raised in the Introduction. Also, when $r^* \ll f$, ORDERED-FS, which has sample complexity logarithmic in f, is likely to learn with many fewer training examples than STANDARD-WRAP.

4 Experimental Results

Our theoretical results predicted ORDERED-FS to be much more tolerant to having a large number of irrelevant features than STANDARD-WRAP. To test this hypothesis, we ran both algorithms on a small, artificial feature selection problem.

The learning algorithm used was logistic regression [McCullagh and Nelder, 1989], used to fit a linear discriminant function, and which, while not minimizing training error, approximates that reasonably. The input space was $X = \mathcal{R}^f$, and the first target concept c we used had only one relevant feature:

$$c(x) = \begin{cases} 1 & \text{if } x_1 + 0.2 > 0 \\ 0 & \text{otherwise} \end{cases}$$

Training examples were corrupted at a noise rate $\eta = 0.3$, and all input features were *i.i.d.* zero-mean unit variance normally distributed random variables. The search heuristic was beam search/forward search (starting out with the empty set of features, and incrementally adding features until we have the full set of features). Forward search is a popular choice that appears to usually do well [Miller, 1990], and beam search, with a beam width of 50 in our case, should be a strict improvement. (Notice also that, while ORDERED-FS was originally formulated as consisting of $f+1$ separate searches, it is probably most naturally implemented as carrying out all the searches "together"; our beam search implementation, which starts from zero features and incrementally considers higher numbers of features, is one example of such.) Unlike many "real life" problems, all of our input features are independent, and so there were, for example, no complicated interactions between them that could complify the search procedure. For STANDARD-WRAP, we are searching for a feature set F so that training on $S'|_F$ would give low hold-out error. For ORDERED-FS, we are searching, for each r, for a feature set F of size r so that training on $S'|_F$ gives low training error. In the rest of this section, we will not distinguish between the "idealized" versions of these two algorithms and the approximate versions of the algorithms. All experimental results reported here are averages of 200 independent trials.

For both algorithms, the hold-out fraction γ is a parameter that had to be chosen. The analysis of [Kearns, 1996] suggests that, for a wide range of hold-out testing applications, $\gamma \approx 0.3$ is a good choice (though it is unclear STANDARD-WRAP would fall into his framework). Using this as an initial choice for γ, we obtain Figure 1, as we vary the total number of features. We see from the graph that ORDERED-FS is performing significantly better on this domain. For reference, the performance of learning without feature selection, using all the features and not saving any data for hold-out testing, has also been plotted; for this problem, this is not really a competitive algorithm (and it is only very slightly competitive on the other target concept we test), and we omit it from the rest of our graphs.

Earlier, our bound had predicted that as f increases, the dominating factor for the error of STANDARD-WRAP comes from testing 2^f hypotheses on γm hold-out samples, thereby possibly "overfitting" the hold-out data. For STANDARD-WRAP, it is therefore natural to see if increasing the hold-out fraction γ might alleviate this effect. Doing so, we obtain Figure 2, which shows results for STANDARD-WRAP using $\gamma = 0.3$, 0.5, and 0.7. While still inferior to ORDERED-FS, the choice of $\gamma = 0.5$ does appear to give better performance for large f, and for the remainder of our experimental results, we report results using STANDARD-WRAP with $\gamma = 0.3$ and 0.5.

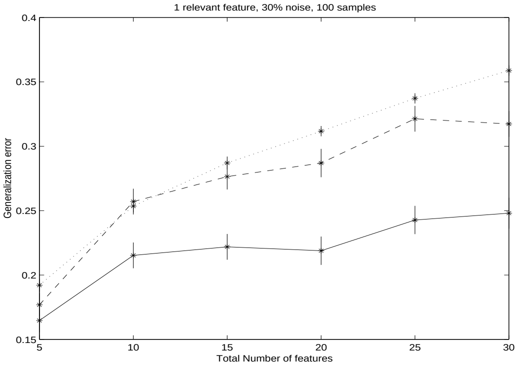

Figure 1: performance of no feature selection training on all the data (dot), of STANDARD-WRAP (dash) with $\gamma = 0.3$ and ORDERED-FS (solid) with $\gamma = 0.3$. Vertical dashes are 1se.

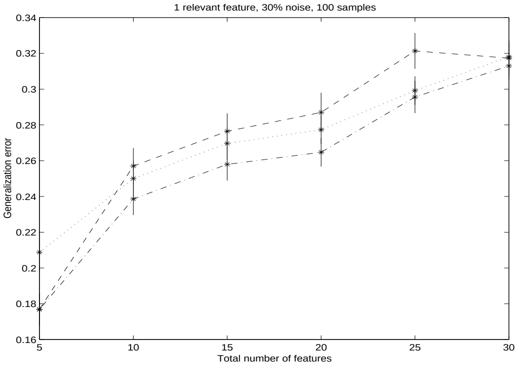

Figure 2: performance of STANDARD-WRAP using $\gamma = 0.3$ (dash), $\gamma = 0.5$ (dot-dash) and $\gamma = 0.7$ (dot). Vertical dashes are 1se.

Next, as we vary m, keeping the total number of features at 20, Figure 3 shows ORDERED-FS still consistently beating STANDARD-WRAP. Lastly, performing similar experiments with a new target function, this time with 3 relevant features

$$c(x) = \begin{cases} 1 & \text{if } x_1 + x_2 + x_3 > 0 \\ 0 & \text{otherwise} \end{cases}$$

we obtain Figures 4 and 5, which both show ORDERED-FS performing significantly better.

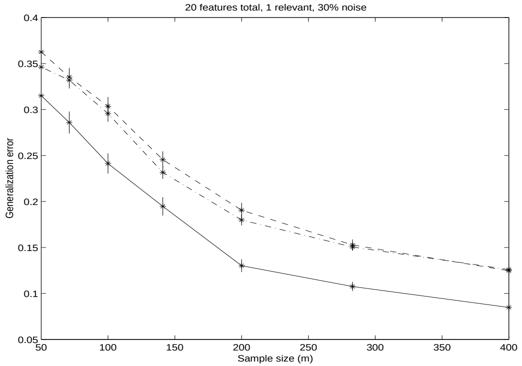

Figure 3: performance of STANDARD-WRAP with $\gamma = 0.3$ (dash) and $\gamma = 0.5$ (dot-dash), and ORDERED-FS with $\gamma = 0.3$ (solid).

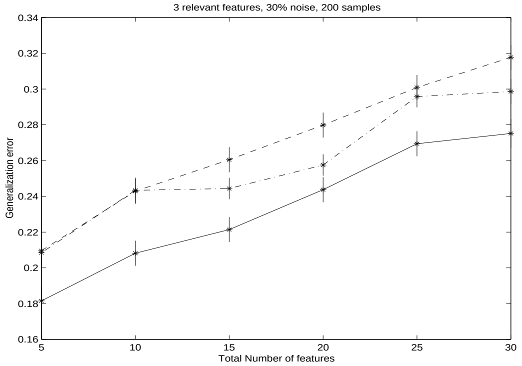

Figure 4: performance of STANDARD-WRAP and ORDERED-FS. Target has 3 relevant features. (Same legend as Figure 3.)

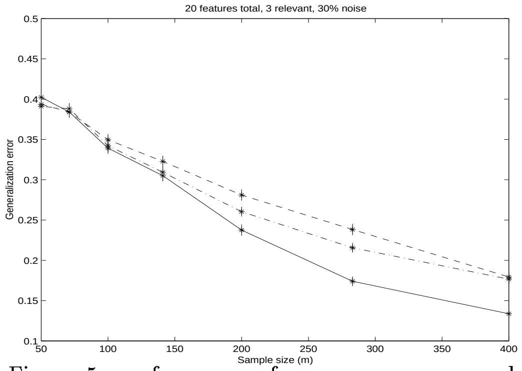

Figure 5: performance of STANDARD-WRAP and ORDERED-FS. Target has 3 relevant features.

5 Discussion and Conclusions

Our experimental results showed our heuristic-search version of ORDERED-FS generally beating that of STANDARD-WRAP. Of course, we do not claim that this will always be the case; indeed, a more detailed analysis than we had given suggests STANDARD-WRAP might do slightly better than ORDERED-FS when the number of relevant features is large, for example if $r \approx 0.5f$. (But then, this is often also the case when feature selection is less useful, compared to learning on the entire set of features.)

Throughout the paper, we skirted the issue of computational expense in (approximately) finding the best (in the training or hold-out error sense) set of features. Indeed, we believe that much work remains to be done on this field, perhaps particularly in designing algorithms for finding feature subsets that minimize training error such as ORDERED-FS requires; for example, we have very efficient algorithms for performing forward and backward search for linear regression [Miller, 1990], but few generalizations or fast approximations thereof to other algorithms. Moreover, for our bounds to predict actual performance well on real problems, we have to rely on these heuristics to perform well, though rigorous bounds for performance using search heuristics can also be given if we can bound how well the heuristic performs the required search/optimization. In particular, if heuristic approximation to STANDARD-WRAP finds only a feature subset that comes within only ε_+ of minimizing hold-out error, then a rigorous bound for its generalization error is the same as for STANDARD-WRAP with an additional ε_+ term. For ORDERED-FS, if for each value of r, we succeed in finding only a feature subset that comes within $\varepsilon_+(r)$ of minimizing training error over all feature subsets of size r, then a rigorous bound for generalization error is the same as for ORDERED-FS but with an additional $\varepsilon_+(r)$ term in the $\{\}$ curly brackets. (We defer proofs and a more detailed discussion of implications to the full paper.) Nevertheless, search heuristics are then immediately seen to be trying to drive ε_+ or $\varepsilon_+(r)$ to zero, and can therefore be argued to be trying to reach the performance suggested by our bounds. (However, one other surprising effect not modeled by our bounds and which deserves mention is that when STANDARD-WRAP is "badly" overfitting the hold-out data, then our earlier work suggests that even randomly throwing some subset of the 2^f hypotheses away may improve performance [Ng, 1997]. This suggests that in such somewhat-degenerate cases, using a weaker search heuristic may actually be helpful. In our experiments, we did manage to find parameter ranges that seemed to exhibit this effect; but, we do not know how prevalent this effect is in practice, and would of course recommend using a good optimization criteria, like ORDERED-FS's, rather than using a less-sound criteria and then to trying to do a poor job in optimizing it.)

Finally, using techniques similar to those used in this paper, it is possible to derive other algorithms or modified versions of our algorithm that, like ORDERED-FS, have strong theoretical properties regarding tolerance to the presence of many irrelevant features, and which may have slightly different strengths and weaknesses than ORDERED-FS; and we discuss a number of them in detail in the full paper. But for now, a significant result of this work is that with appropriate feature selection, sample complexity becomes *logarithmic* in the number of irrelevant features, so that we can handle *exponentially* many irrelevant features as training examples. Of course, we still have rely on search heuristics to help us reach these bounds, and while much empirical work remains to be done evaluating ORDERED-FS and comparing it to STANDARD-WRAP and possible interpolations between the two algorithms, we also believe that being able to give these bounds is very encouraging, because it means that if they are even only approximately realizable by search algorithms, they may have tremendous consequences for feature design – that modulo computational expense, overly careful human design of features may often be unnecessary, so long as additional training data can be obtained reasonably cheaply.

Acknowledgments

I give warm thanks to Michael Jordan and Dana Ron for interesting and helpful conversations. Also, this work would not have been possible if not for numerous greatly edifying early conversations I had with Michael Kearns about VC theory in general and related work. The author was supported by the National Science Foundation under Contract No. ASC-92-17041.

References

[Almuallim and Dietterich, 1994] Almuallim, H. and Dietterich, T. (1994). Learning boolean concepts in the presence of many irrelevant features. *Artificial Intelligence*, 69(1-2):279–305.

[Caruana and Frietag, 1994] Caruana, R. and Frietag, D. (1994). Greedy attribute selection. In *Proceed-*

ings of the Eleventh International Conference on Machine Learning. Morgan Kaufmann.

[Garey and Johnson, 1979] Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman.

[John et al., 1994] John, G., Kohavi, R., and Pfleger, K. (1994). Irrelevant features and the subset selection problem. In *Proceedings of the Eleventh International Conference on Machine Learning.* Morgan Kaufmann.

[Kearns, 1996] Kearns, M. J. (1996). A bound on the error of Cross Validation using the approximation and estimation rates, with consequences for the training-test split. In *Advances in Neural Information Processing Systems 8*, pages 183–189. Morgan Kaufmann.

[Kearns et al., 1997] Kearns, M. J., Mansour, Y., Ng, A. Y., and Ron, D. (1997). An experimental and theoretical comparison of model selection methods. *Machine Learning Journal*, 27(1):7–50.

[Kearns and Ron, 1997] Kearns, M. J. and Ron, D. (1997). Algorithmic stability and sanity-check bounds for leave-one-out cross-validation. In *Proceedings of the Tenth Annual Conference on Computational Learning Theory.* Morgan Kaufmann.

[Kivinen and Warmuth, 1994] Kivinen, J. and Warmuth, M. K. (1994). Exponentiated gradient versus gradient descent for linear predictors. Technical Report UCSC-CRL-94-16, Univ. of California Santa Cruz, Computer Research Laboratory.

[Kohavi and John, 1997] Kohavi, R. and John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, 97:273–324.

[Kohavi and Sommerfield, 1995] Kohavi, R. and Sommerfield, D. (1995). Feature subset selection using the wrapper model: Overfitting and dynamic search space topology. In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining.*

[Langley, 1994] Langley, P. (1994). Selection of relevant features in machine learning. In *Proceedings of the AAAI Fall Symposium on Relevance.* AAAI Press.

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

[McCullagh and Nelder, 1989] McCullagh, P. and Nelder, J. A. (1989). *Generalized Linear Models (second edition).* Chapman and Hall.

[Miller, 1990] Miller, A. J. (1990). *Subset Selection in Regression.* Chapman and Hall.

[Moore and Lee, 1994] Moore, A. W. and Lee, M. S. (1994). Efficient algorithms for minimizing cross validation error. In *Proceedings of the 11th International Conference on Machine Learning.*

[Ng, 1997] Ng, A. Y. (1997). Preventing "overfitting" of Cross-Validation data. In *Proceedings of the Fourteenth International Conference on Machine Learning.* Morgan Kaufmann.

[Vapnik, 1982] Vapnik, V. N. (1982). *Estimation of dependencies based on empirical data.* Springer Verlag.

[Vapnik and Chervonenkis, 1971] Vapnik, V. N. and Chervonenkis, A. Y. (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264–280.

[Yang and Hoavar, 1997] Yang, J. and Hoavar, V. (1997). Feature subset selection using a genetic algorithm. In *IEEE Expert (Special Issue on Feature Transformation and Subset Selection).* In press.