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

by

Andrew Y. Ng

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 1998



Signature of Author .....
Department of Electrical Engineering and Computer Science
August 20, 1998

Certified by .....
Michael I. Jordan
Professor of Brain and Cognitive Sciences
Thesis Supervisor

Accepted by .....
A. C. Smith
Chairman, Departmental Committee on Graduate Students

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

by

Andrew Y. Ng

Revised version of a thesis submitted to the
Department of Electrical Engineering and Computer Science
on August 20, 1998, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

## Abstract

We consider feature selection for supervised machine learning 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.

# Acknowledgments

I give warm thanks to my advisor, Michael Jordan, for his guidance over the past year, and to both him 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, and I warmly thank him for all that and more. Finally, the author also gratefully acknowledges support from the National Science Foundation under Contract No. ASC-92-17041.

# Contents

# List of Figures

# Chapter 1

# Introduction

## 1.1 Feature Selection

In the common supervised machine learning problem, we are given a training set of input and output pairs, and are asked to learn a function of the inputs that predicts the outputs well, for example in the sense of low classification error if the outputs are binary. Each input typically consists of some list of "features" of which our learned mapping will be a function. But in many applications, it is often the case that only a small subset of these features are required to achieve low output error. For example, consider a medical diagnosis problem in which each input is a long list telling us about the presence or absence of possible symptoms in a patient, and where we are asked to predict from these symptoms whether or not the patient has a particular disease. Then it may well be the case that knowing about only a small subset of these symptoms will be sufficient to do an excellent job of predicting whether the patient has the disease. Now if we knew exactly which subset of these features were required to predict if the patient has the disease, then it is conceivable and in fact likely that if we restricted the inputs to having just this subset of features, then our learning algorithms would be able to learn a better classifier.

If we do indeed believe that the target concept may be well-represented using only a small number of features, then we may try to first identify these "relevant" features, and then do learning using only these features. This is where feature selection comes in. Informally, the goal of feature selection is commonly seen as that of, via examining the training data, selecting a subset of the features so as to eliminate the "irrelevant" features, with the hope that learning using only this smaller subset of features will result in a better estimator than if we had used the entire set of features. This is intuitively a very natural thing to do, and feature selection for classification and regression is a topic that has in recent years been enjoying increasing interest in the Machine Learning community; impressive performance gains have been reported by numerous authors, and numerous feature selection algorithms and feature subset search heuristics have also been proposed. (In Chapter 2, we will survey some of this work in more detail.)

## 1.2 Central Questions

In this section, we describe what we see as two central questions regarding feature selection, and which it will be the goal of this Thesis to answer.

An interesting observation is that many learning algorithms will, even without an explicit feature selection step, "automatically" learn to ignore "irrelevant" features. For example, in most parametric function approximation schemes such as linear regression, eliminating a feature is mathematically identical to setting the weights or coefficients associated with that feature to zero. If a particular feature is indeed unnecessary for our learning task, then even if we left the feature in and tried to estimate its associated coefficient, its estimated coefficient will typically be near zero anyway, so that the feature will "effectively" be ignored. While this may not yield as good an estimator as if we had known apriori to eliminate the feature, it may also not harm our performance too much to leave this "irrelevant" feature in, because its estimated coefficients will typically be near zero anyway.

Rather than leaving all the features in, feature selection schemes instead try to estimate, from the statistics of the training data, which features are "irrelevant," and set their coefficients to zero. Of course, this is done using possibly noisy training data, and naturally runs a risk of misidentifying the "irrelevant" features. Thus, despite the impressive empirical results from the feature selection literature, one central question is: *Why is it often superior (if indeed it is) 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)?* That is, why should be perform feature selection at all – is the noise introduced by risking misidentifying "irrelevant" features smaller than the noise introduced by including "irrelevant" features? The theoretical results in this thesis will address this question.

Furthermore, 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, and how can we make feature selection algorithms that scale well with the number of irrelevant features?* A closely related question is, how does the sample complexity (informally, meaning the number of training examples required to learn "well") of feature selection scale with the number of irrelevant features, or how can we perform feature selection so that sample complexity grows very slowly with the number of irrelevant features? These are questions whose answers have direct implications for how we design the features we give to our learning algorithms. For example, if performance degrades very slowly with the number of irrelevant features, then we may not need to spend too much human-expert effort towards selecting only the "relevant" features for inclusion, and we may perhaps choose to dedicate more effort towards improving our estimator in other ways (such as by obtaining more training examples).

Briefly summarizing the two central questions we attempt to answer, they are:

- Why perform feature selection at all, when it typically means risking misidentifying the "irrelevant" features, and most learning algorithms will automatically learn to ignore the "irrelevant" features anyway?

- With feature selection, how does performance (and sample complexity) scale with the number of "irrelevant" features, and how can we perform feature selection in a way that scales really well with the number of "irrelevant" features?

With respect to the answering the second of these questions, there are, outside of the feature selection literature, a number of algorithms (sometimes called "feature weighting" algorithms [3]) for learning certain restricted concepts and whose performance degrades particularly gracefully with the number of irrelevant features. We will review them in more detail in Chapter 2, but briefly, the Winnow algorithm of Littlestone for learning Boolean monomials from noiseless data enjoys worst-case loss logarithmic in the number of irrelevant features [21]. Likewise, Kivinen and Warmuth's 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 [16]. Finally, within the feature selection literature, when learning, from noiseless data, a representation of a boolean function (over boolean inputs), Almuallim and Dietterich have also shown that an algorithm that finds the smallest set of features consistent with the training data also enjoys loss logarithmic in the number of irrelevant features [1]. If it were true in general that feature selection can make 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, and seems a worthy goal to aim for. As a preview to a main result of this Thesis, we will actually 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.

## 1.3 Thesis Overview

The remainder of this thesis is structured as follows: In Chapter 2, we give an overview of the feature selection framework that we will be using, review some related work, and also begin to develop the notation that we will use to state our formal results. Chapter 3 then presents the main theoretical results of this Thesis, which will give answers to our two central questions raised in the previous section. In doing so, we also develop a new algorithm that, from an information-theoretic point of view, scales much better than previous algorithms with the number of irrelevant features, and has the logarithmic sample complexity we just mentioned. We propose this as a viable alternative to current algorithms, and in Chapter 4, spell out some of its implementational details, and also empirically compare it on an artificial problem to another feature selection algorithm, and to learning without feature selection. Chapter 5 then extends some of our results in Chapter 3, examining the finer details of our proofs and also considering a possible modification to our algorithm. Finally, Chapter 6 closes with conclusions and future work.

Also, as a guide to understanding the theoretical results in this thesis, we wished to present our proofs in a manner accessible to readers without a background in Computational Learning Theory, but also without making the entire thesis a tutorial on the subject. Thus, while our proof sketches within the thesis body are brief and intended only for readers familiar with VC-theory and related work, the Appendix to this Thesis also contains full and detailed proofs of our main results, written in a tutorial fashion and meant to be widely accessible. In reading this Thesis, readers without a background on Computational Learning Theory may therefore wish to refer not to the proofs in the Thesis body but instead to the proofs in the Appendix, which also gives a (hopefully) readable tutorial introduction to some of the very interesting methods used in the field.

Our analysis is largely inspired by Kearns [12], with our theoretical results heavily based on the techniques given there and those outlined in Kearns, Mansour, Ng and Ron [13]. We also rely heavily on tools from Vapnik [29], that give a very general framework for bounding the deviation of training error from generalization error.

# Chapter 2

# Background and Preliminaries

In the chapter, we first discuss what the goal of feature selection is, then briefly review previous work on feature selection paying particular attention to the "wrapper model," and set up the framework that we will use for our analysis.

## 2.1 The goal of feature selection?

The notion of "relevance" is closely related to feature selection. In the previous chapter, we said that one view of feature selection's goal is that of eliminating all but a small set of "relevant" features which will be given to a learning algorithm. And indeed, it is quite possible to set this up as feature selection's goal and to try to design algorithms to do exactly this, and obtain reasonable empirical results. For example, in text categorization, a simple heuristic estimating how dependent individual features and the target concept are (more formally, in terms of mutual information) so as to try to eliminate the "irrelevant" features seems to do well [32], and as we will describe in the next section, most of the other "filter" model approaches to feature selection take a similar view towards feature selection [8].

But as pointed out by Kohavi and John [18], there have been difficulties with a number of definitions of relevance. Paraphrasing them, the main problems is how to exclude "redundant" features – for example, if two of the input features give exactly identical information (say, height in centimeters and height in meters), then neither feature is "necessary" as we can always exclude it and use the other. But they may still be encoding a required attribute (height in our example); so despite neither of these features being necessary, nor would we want to declare them both "irrelevant" and exclude both of them.

Therefore, we take an alternative view of the goal of feature selection, which we feel also gets much more directly at what we are trying to achieve, 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 is very similar in flavor to the goals of some of the "feature weighting" schemes in [21] and [16], and also bears some resemblance to the framework used in [13]. This goal 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 Thesis, we will try to use the terms "relevant" and "irrelevant" only informally and when we expect an intuitive sense these words to suffice.

## 2.2 Related Work

### 2.2.1 Feature selection algorithms

Informally stated, a very general form of the feature selection problem (ideologically if perhaps not notationally consistent with our goal in feature selection that we just described) for supervised machine learning is the following: We are given a learning algorithm $L$ that, given any training set consisting of input/output pairs, tries to fit a function of the inputs, that will accurately predict the outputs. Suppose many of the input features are unnecessary ("irrelevant") to learning a good mapping from the inputs to the outputs; we then wish to identify only a small subset of ("relevant") features, and eliminate all but this set of features from $L$'s consideration, such that it will hopefully output a better hypothesis. All this will be made formal and concrete later in this Chapter, but let us now briefly review some previous work on feature selection.

Feature selection in various forms has been around for many years and, in certain instantiations (usually in the statistical literature,) is also sometimes called "model selection." An excellent survey of much of the earlier material, mostly from the statistics literature and dealing with regression, is in Miller [24]; but since it is classification and not regression that we will mostly be studying in this Thesis, we will not be focusing on the family of algorithms designed for feature selection in regression. Thus for now, let us just concentrate on the more recent work on algorithms for feature selection in classification, which comes mostly from the Machine Learning literature.

Blum and Langley [3] have an excellent review of recent work in feature selection methods, and our review is largely based on theirs. Also, we will describe only a small number of the many noteworthy algorithms in the field, with the goal being mainly to convey an idea of the issues addressed by researchers, and we refer the reader to Blum and Langley's review for a fuller overview of the field.

An early feature selection algorithm is Almuallim and Dietterich's FOCUS algorithm, intended for learning problems with boolean inputs and outputs [1]. They suggest trying to find the smallest set of input features such that there exists a mapping from just these features to the output that is exactly consistent with the training set, and then learning (via a decision tree $L$) using just these features. Their algorithm clearly works only on boolean or discrete domains, and would naturally be quite brittle with respect to noisy examples in the training set (for example, if training examples are occasionally mislabelled), so from a practical point of view, it is perhaps not a very widely applicable algorithm. But, it does nicely capture the

[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 thesis.

intuition of wanting to find a small set of "relevant" features that are sufficient to predict the output well. Under the (stringent) assumptions of noiseless training data and boolean inputs and outputs, they show a nice result that if the output is *exactly* a binary function of a small number of the inputs, then if FOCUS can find the smallest set of features consistent with the training set (an NP-hard search problem, unfortunately), the resulting learning algorithm will have sample complexity logarithmic in the number of irrelevant features.

Next, Kira and Rendell's Relief algorithm [15] allows general inputs/outputs this time, and does not rely on having perfectly noiseless training examples. They use a number of heuristics to try to assign to each feature a score for how useful it is in labeling the training data correctly, and then perform learning using only the features that had been assigned a score above some user-defined threshold. Again, this nicely captures the intuition of trying to assign scores to features according to how "relevant" they are to the learning problem, and then using only those features that are "sufficiently relevant." We know of no theoretical results regarding Relief (and it is actually not too difficult to construct examples where Relief's heuristics would utterly fail to find a good small feature subset; for example, Relief is known to be completely ineffective at removing *redundant* features). Nevertheless, Relief is appealing in terms of its being a relatively fast algorithm, and their experiments also showed it giving good performance on a few problems.

To try to capture the notion of a feature being "relevant" to the mapping we are trying to learn, Koller and Sahami [19] have a recent paper in which they appeal to the language of probability theory, and suggest trying to find a small feature subset such that the excluded features convey no further information about the output (formally, that the excluded features are conditionally independent of the output given the included features). Doing exactly this actually turns out to be quite infeasible, and so they propose a number of heuristics to try to approximate doing this. In terms of classification accuracy, their experiments did not demonstrate a significant improvement (though they do argue convincingly about giving significant computational savings).

A commonality between the algorithms described so far is that all of them try to find a good feature subset independently of the learning algorithm $L$. Using the terminology due to John et. al. [8], these are "filter" model feature selection algorithms, in that they rely on general characteristics of the training data to select some feature subset, doing so without reference to the learning algorithm. Also in John et. al.'s terminology, the alternative is the "wrapper" model feature selection algorithms, where one generates sets of candidate features, runs them through the learning algorithm, and uses the performance of the resulting hypothesis to evaluate the feature subset in what usually then becomes a search for a good feature subset. Since we will revisit the wrapper model shortly, we will not describe its workings in more detail here; but while the wrapper model tends to be computationally more expensive (since it generally requires a search problem and repeated applications of $L$,) it also unsurprisingly tends to find feature sets better suited to the inductive biases of our learning algorithm, and tends to give superior performance [20]. But because of its computational cost, a significant amount of work on wrapper algorithms has been to develop good search heuristics or fast evaluation methods to quickly allow us to find

a good feature subset. (Examples include [25, 5, 31]; see [3, 24] for many more.)

### 2.2.2 Learning with many irrelevant features

Interestingly, another approach to the "feature selection problem" actually does not select a particular feature subset, but has led to very strong theoretical results regarding algorithms that scale logarithmically with the number of "irrelevant" features. They tend to be for restricted domains, and the one algorithm in this family (Littlestone's Winnow, described below, and certain variants of it) that is directly applicable to feature selection for classification unfortunately works only under stringent assumptions, such as perfectly noiseless outputs and an exactly representable "true mapping" from a small set of functions. Nevertheless, this literature represents a very interesting and significant body of work that has answered a number of important questions, and we will very briefly describe some of it here (though again, the reader is referred to [3] for a larger overview).

The Winnow algorithm of Littlestone for learning Boolean monomials (for example, functions of the form $x_{i_1} \wedge x_{i_2} \wedge \ldots \wedge x_{i_r}$, where the $x_j$s are boolean inputs) from noiseless data enjoys worst-case loss that grows only logarithmically in the number of irrelevant features [21]. Through various transformations, this also allows learning certain other restricted concepts, such as $k$-DNF formulae for small $k$, and $r$-of-$k$ threshold functions (over boolean inputs; where the output is 1 whenever at least $r$ of the $k$ relevant inputs are 1). Likewise, Kivinen and Warmuth's remarkable 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 [16]. Aside from these, there is also a number of related algorithms that use similar techniques to solve various related and unrelated problems; we will not review them here, but as just one example, there is the Weighted Majority algorithm for predicting outputs nearly as well an the best classifier in a given pool of classifiers [22].

Both Winnow and EG, and all of their variants that we are aware of, are only for learning specific concepts; for example, Winnow can learn only a small class of functions, as described above. This is in contrast to feature selection algorithms that try to accommodate general learning algorithms $L$ that may output arbitrary target concepts. Moreover, most/all of their formal results seem to be based on using potential functions to prove worst-case loss bounds in the online learning case: That is, suppose that we are sequentially given input/output pairs and that on the $i$-th step, after being given the $i$-th input but before being given the $i$-th output, we are asked to predict the $i$-th output as a function of the input we were just given. Suppose further that after being shown each output, we are allowed to change our classifier. Then the online worst-case framework asks, after any $m$ such examples, can we bound the total number of errors we have made so far? It is a remarkable result that Winnow and EG enjoy worst-case online loss logarithmic in the number of irrelevant features. So, one might also ask, what of the perhaps more natural framework of average-case loss (where we train and test from the same distribution, and ask about expected test error)? It is usually possible to transform worst case loss bounds into average-case loss bounds, though since worst-case loss bounds must hold for even an adversarially

chosen set of $m$ examples, they often give significantly looser bounds than the style of bounds provable in average-case results; this is particularly true when there is training label noise or when the target mapping can only be approximately represented by our classifier[2]. In this work, we will actually focus on average-case error, and will be able to prove the tighter forms of the bounds that give better guarantees on generalization error. (See preceding footnote.)

## 2.3 Preliminaries

In this section, we lay out the notation and the formal framework and assumptions that we will be using for the remainder of this Thesis.

### 2.3.1 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 mapping inputs to outputs, $c : X \longmapsto \{0,1\}$. 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 tl e 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 $\eta$. 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 the natural way (with $h$ ignoring features not in $F$). Following standard PAC assumptions [28], we assume testing is done with respect to samples drawn from the same distribution $D_X$ as the training samples were, and we can thus, for any hypothesis $h$, 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). Also, we write the empirical error of $h$ on a set of data $S$ as $\hat{\varepsilon}_S(h) = \frac{1}{|S|}|\{(x,y) \in S | h(x) \neq y\}|$.

[2]For example, as a general flavor of these results, if the "best possible classifier" has error $\varepsilon$, then a worst-case loss style algorithm may be able to approach $k\varepsilon$ generalization error, where $k$ is some constant (possibly significantly) larger than 1. (See [16] for caveats, and a brief discussion on this.). In contrast, for average-case style algorithms, we may be able to approach $\varepsilon$ error, the very best error possible. Unless $\varepsilon$ is close to zero, this can be quite a significant difference.

### 2.3.2 The wrapper model

In this section, we describe a very natural way of performing feature selection, and further develop the notation and assumptions that we use in the remainder of the Thesis. The very natural feature selection algorithm we describe here will be one that we study as we try to answer the questions raised in the Introduction.

In the wrapper model of feature selection suggested by [8], 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 $\gamma m$, 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 [14], 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 [29], and is also used in [12] and [13] 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 [13] 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 too much in terms of theoretical results. Though see also Section 5.3 for when we relax this assumption.) 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 Thesis. 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 $S|_F$. Unless otherwise stated, we will, for the remainder of this Thesis, 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 [30, 29] of the hypothesis class $H_r$, which characterizes the "complexity" 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 in Chapter 3 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.[3] 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 [6], 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 [25, 5, 31], and [20, 24] include overviews.) In this development, we will, in the style of [12], 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 in Chapter 5) aspiring to do, with the bounds giving insight into how we might expect the algorithms to perform.

[3] As a minor technical point, we may wish here to be more careful about whether this is the minimum or the infimum error achievable using $r$ features; purely for subsequent notational convenience in our proofs, let us assume the former, though the latter case will also be briefly treated in the proofs in Chapter 3.

# Chapter 3

# Main Results

In this Chapter, we prove our main theoretical results regarding learning with and without feature selection, and also propose a new algorithm that has rather strong theoretical properties regarding how well it scales with the number of irrelevant features. Our analysis is largely inspired by [12], with our theoretical results heavily based on the techniques given there and those outlined in [13]. We also rely heavily on tools from [29], that give a very general framework for bounding the deviation of training error from generalization error. Also, the reader is reminded that while only proof sketches are given in this Chapter, the Appendix contains detailed (and significantly lengthier) versions of these proofs which are written in a tutorial fashion meant to be accessible to persons without a background in Computational Learning Theory, and which we hope will also serve some as a readable introduction to a couple of interesting results from the field.

The ensuing bounds on generalization error are all given to hold "with high probability," by which we mean each of the bounds hold with at least probability $1 - \delta$ for any $\delta > 0$, with constants that depend on $\delta$ (through an omitted $\log \frac{1}{\delta}$ term) hidden by the $O(\cdot)$ notation. Our handling of uncertainty and informal hiding of constants using the big-$O$ notation is quite standard, and the full, formal versions of these bounds are given in the Appendix.

## 3.1 Learning without feature selection

To start, let us consider learning without feature selection, and ask how such a learning algorithm might perform. The Universal Estimate Rate bound of Vapnik and Chervonenkis [30, 29] (discussed in more detail in the Appendix) 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:*

$$\varepsilon(\hat{h}) \leq \varepsilon_g(f) + O\left(\sqrt{\frac{f_{\mathrm{VC}}}{m}\left(\log\frac{m}{f_{\mathrm{VC}}} + 1\right)}\right) \tag{3.1}$$

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

## 3.2 Performance of STANDARD-WRAP

Next, we consider the STANDARD-WRAP algorithm. Applying largely the proof technique given in [12] (used to bound the error of hold-out) to feature selection, we obtain the following theorem:

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

$$\varepsilon(\hat{h}) \leq \min_{0 \leq r \leq f} \left\{ \varepsilon_g(r) + O\left( \sqrt{\frac{r_{VC}}{(1-\gamma)m} \left( \log \frac{m}{r_{VC}} + 1 \right)} \right) \right\} + O\left( \sqrt{\frac{f}{\gamma m}} \right) \tag{3.2}$$

**Proof (Sketch):** Conceptually, we may think of STANDARD-WRAP as enumerating all $2^f$ feature subsets, running $L$ on the $(1-\gamma)m$ training samples restricted to each of these features subsets, and putting the resulting $2^f$ hypotheses into a set $\mathcal{H}$, and then finally using the hold-out set to pick the one in the set $\mathcal{H}$ with the lowest hold-out error. Now fix any $r$, $0 \leq r \leq f$. Let $F_r$ be the feature subset so that $H_{F_r}$ contains a hypothesis $\hat{h}_r^*$ with $\varepsilon_g(r)$ generalization error.[1] Then, by the Universal Estimation rate bound again (essentially Theorem 1), we know that with high probability, upon generating the hypothesis $h = L(S'|_{F_r})$, the resulting hypothesis will have generalization error bounded by

$$\varepsilon(h) \leq \varepsilon_g(r) + O\left( \sqrt{\frac{r_{VC}}{(1-\gamma)m} \left( \log \frac{m}{r_{VC}} + 1 \right)} \right) \tag{3.3}$$

Thus, with high probability, the set $\mathcal{H}$ contains at least one hypothesis with error bounded by the above. Lastly, via a standard Chernoff bound and uniform convergence argument, when use an independent hold-out test set of size $\gamma m$ to estimate generalization errors on $2^f$ hypotheses, the errors of our estimates will, with high probability, be simultaneously bounded (for all $2^f$ hypotheses) by $O(\sqrt{\log(2^f)/\gamma m}) = O(\sqrt{f/\gamma m})$. Thus with high probability, when we pick the hypothesis in $\mathcal{H}$ with the lowest hold-out error, we pick a hypothesis that is at most $O(\sqrt{f/\gamma m})$ worse than the best hypothesis in the of $2^f$, giving an additional $O(\sqrt{f/\gamma m})$ term. Finally, the

[1] A comment only for readers familiar with real analysis and wondering if $\hat{h}_r^*$ may exist: As briefly mentioned in Section 2.3, one may wish to be more careful about whether $\varepsilon_g(r)$ is the minimum or infimum error achievable using $r$ features. The argument holds with a tiny modification for the latter case (by considering a sequence of $\hat{h}_r^*$s whose generalization errors approach $\varepsilon_g(r)$), so purely for notational convenience, let us assume such a hypothesis exists.

above argument holds for any fixed $r$; in particular, it must hold for the value of $r$ that minimizes the right size of Equation (3.3). (It may be a slightly subtle point that we can take this min-operation over $r$ without needing another uniform bound over all $r$; see the Appendix if the reader wishes further justification.) This then gives us our result. □

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 $\eta$ is the noise rate.

Recall that, in the Introduction, we said one of the questions we care about is how our feature selection algorithms scale with the number of irrelevant features. Notice that for STANDARD-WRAP, as the number of irrelevant features (or $f$) becomes large, the dominating term in our bound is the $O(\sqrt{f/\gamma m})$ term; so, let us spend a moment examining where it comes from.

It is a well-understood fact (described in detail in Appendix A.1) that when we use an independent hold-out set $S''$ of size $\gamma m$ to test a set of $N$ hypotheses and pick the one with the lowest hold-out error, then with high probability, the hypothesis we end up picking will be no more than $O(\sqrt{\log N/\gamma m})$ worse than the best hypothesis in the set of $N$. Here, we are conceptually testing a pool of $N = 2^f$ hypotheses corresponding to the $2^f$ feature subsets, so substituting $N = 2^f$ in, we have a a $O(\sqrt{f/\gamma m})$ bound. But, this is a worst-case bound for evaluating $2^f$ hypotheses on an the independent hold-out set size $\gamma m$. 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 $\gamma m$ hold-out samples. (In the context of feature selection, the issue of overfitting of hold-out data was also raised by Kohavi and Sommerfield [17]. Also, it is a subtle issue how such "overfitting" really occurs; see our earlier work [26] for a detailed discussion on such overfitting of hold-out data in hypothesis selection.) 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 fully formalize this, 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 $\alpha \in (0,1]$. In Section 5.2, we will actually examine this issue and the $O(\sqrt{f/\gamma m})$ term in significantly more detail, to try to understand how actual performance might behave; and our analysis there suggests that for a (perhaps surprisingly large) range of behaviors for how much hypotheses change when $F$ is changed, the number of pairwise-"significantly different" hypotheses does grow as $2^{O(f)}$, which suggests $\alpha = 1$ behavior. On the other hand, it is also possible to come up with (sometimes slightly unnatural) feature selection scenarios where a bound with $\alpha < 1$ may be proved to

hold, but we defer this discussion to Section 5.2.

## 3.3 Performance of a new algorithm

For STANDARD-WRAP, the dependence on $f$ of our bound on the error is $\sqrt{f/\gamma m}$ (or possibly $\sqrt{f^a/\gamma m}$), 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. So, towards performing feature selection in a way that gives good performance even as the number of irrelevant features becomes very large (that is, as $f$ becomes large), let us try to design a new algorithm that does not involve this process of testing $2^f$ hypotheses on a hold out set. Consider the following algorithm, which we call ORDERED-FS:

1. For each $0 \leq r \leq f$, 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 $\sqrt{(\log f)/\gamma m}$.

One succinct way of describing the difference between this and STANDARD-WRAP is what while STANDARD-WRAP uses the hold-out data to decide *which particular feature subset* to use, ORDERED-FS instead uses the hold-out data only to decide *how many features* to include. That is, STANDARD-WRAP may be viewed as using the hold-out set to evaluate all $2^f$ feature subsets, picking the one particular feature subset that gave the lowest hold-out error. In contrast, ORDERED-FS uses the hold-out set to test only $f+1$ hypotheses before picking the one with the lowest hold-out error, and its choice corresponds exactly to deciding whether to use 0 features, 1 feature, 2 features, ..., or $f$ features – that is, the hold-out set is essentially used to decide how many features to use.

As for ORDERED-FS's performance, we have the following theorem:

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

$$\varepsilon(\hat{h}) \leq \min_{0\leq r\leq f}\left\{\varepsilon_g(r) + O\left(\sqrt{\frac{r_{\mathrm{VC}}}{(1-\gamma)m}\left(\log\frac{m}{r_{\mathrm{VC}}}+1\right)}\right) + O\left(\sqrt{\frac{r}{(1-\gamma)m}\left(\log\frac{f}{r}+1\right)}\right)\right\} + O\left(\sqrt{\frac{\log f}{\gamma m}}\right) \tag{3.4}$$

**Proof (Sketch):** Let $\mathcal{H}$ denote the set of hypotheses $\{\hat{h}_r\}_{r=0}^{f}$. Now, fix any $r$, $0 \leq r \leq f$. Then for any one fixed feature subset $F$ of size $r$, with probability $1-\delta$, it holds true simultaneously for all $h \in H_F$ that $|\varepsilon(h)-\hat{\varepsilon}_{S'}(h)| \leq O\left(\sqrt{\frac{r_{VC}}{m}(1+\log\frac{m}{r_{VC}})+\frac{1}{m}\log(1/\delta)}\right)$ (where, for the sake of brevity within the proof sketch, we now hide $1-\gamma$ and $\gamma$ in the big-$O$ notation as well). Thus, by taking a union bound over all $\binom{f}{r}$ feature subsets of size exactly $r$, with probability at least $1-\delta$, it holds simultaneously for all $h \in H_F, |F| = r$ that $|\varepsilon(h) - \hat{\varepsilon}_{S'}(h)| \leq O\left(\sqrt{\frac{r_{VC}}{m}(1+\log\frac{m}{r_{VC}})+\frac{1}{m}\log(\binom{f}{r}/\delta)}\right)$. Let $\chi(r)$ denote this $O(\cdot)$ term, and $\hat{h}_r^*$ be the hypothesis that, using $r$ features, achieves generalization error $\varepsilon_g(r)$.[2] Then, when we choose the hypothesis $\hat{h}_r$ that minimizes training error over all hypotheses using exactly $r$ features, we have, with probability at least $1-\delta$:

$$\begin{aligned}
\varepsilon(\hat{h}_r) &\leq \hat{\varepsilon}_{S'}(\hat{h}_r) + \chi(r) && (3.5)\\
&\leq \hat{\varepsilon}_{S'}(h_r^*) + \chi(r) && (3.6)\\
&\leq \varepsilon(h_r^*) + 2\chi(r) && (3.7)\\
&= \varepsilon_g(r) + 2\chi(r) && (3.8)
\end{aligned}$$

where the first and third inequalities used $|\varepsilon(h) - \hat{\varepsilon}_{S'}(h)| \leq \chi(r)$, and the second used $\hat{\varepsilon}_{S'}(\hat{h}_r) \leq \hat{\varepsilon}_{S'}(h)\ \forall h \in H_F, |F| = r$ (since $\hat{h}_r$ was chosen to minimize training error). Thus with high probability, the set $\mathcal{H}$ of hypotheses contains at least one hypothesis with generalization error bounded by $\varepsilon_g(r) + 2\chi(r)$. Taking a Chernoff bound argument as we did in the proof for STANDARD-WRAP, when we use a hold-out set of size $\gamma m$ to try to pick the best of our $f+1$ hypotheses in $\mathcal{H}$, we will, with high probability, pick a hypothesis $\hat{h}$ no more than $O((\log f)/\gamma m)$ worse than this. So, with high probability, $\varepsilon(\hat{h}) \leq \varepsilon_g(r)+2\chi(r)+O(\sqrt{(\log f)/\gamma m})$. Again this was proved to hold for any fixed $r$; in particular, it holds for the value of $r$ that minimizes the right hand size (though again, it may be a subtle point that it is a correct operation to take a min over $r$ without taking another union bound over all $r$; see the Appendix if the reader wishes more detailed justification). Finally, substituting back the definition of $\chi(r)$ and manipulating it slightly (mostly to bound $\log\binom{f}{r}$, using a counting result from [4]; see the Appendix), we arrive at our theorem. □

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, and we describe in Chapter 4 a very natural way of doing this. 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 that we will make formal in Chapter 5, 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

[2] Again, similar comments as the last footnote regarding existence of $\hat{h}_r^*$ apply.

what a heuristic search implementation is trying to attain. (In considering heuristic search implementations, it is also worth pointing out 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. [24]) 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 $f$. As we will shortly see, its sample complexity is also logarithmic in $f$. So, 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. Of course, we still do need good search algorithms in the heuristic implementations of ORDERED-FS to make it tractable, but we believe that this result, 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.

## 3.4 Sample Complexities and comparisons

To close this section, it is informative to informally re-state our 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, suppose the target concept can be well-represented using $r^*$ features. Then how many training examples $m^*$ would each of these learning algorithms need to learn a hypothesis with generalization error that will be nearly as good as the best hypothesis that uses only $r^*$ 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 $\epsilon$ will again be hidden by the $O(\cdot)$ notation.) From the earlier theorems, it is not too 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 can have 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 our first central question raised in the Introduction, of why we might want to do feature selection at all. That is, the answer is that if we are using a hypothesis class whose complexity $r_{VC}$ grows superlinearly in $r$ (for example, if $r_{VC} = r^2$) then if it is indeed true that the target concept can be well-represented using only a small subset of the input features, then feature selection

what a heuristic search implementation is trying to attain. (In considering heuristic search implementations, it is also worth pointing out 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. [24]) 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 $f$. As we will shortly see, its sample complexity is also logarithmic in $f$. So, 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. Of course, we still do need good search algorithms in the heuristic implementations of ORDERED-FS to make it tractable, but we believe that this result, 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.

## 3.4 Sample Complexities and comparisons

To close this section, it is informative to informally re-state our 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, suppose the target concept can be well-represented using $r^*$ features. Then how many training examples $m^*$ would each of these learning algorithms need to learn a hypothesis with generalization error that will be nearly as good as the best hypothesis that uses only $r^*$ 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 $\epsilon$ will again be hidden by the $O(\cdot)$ notation.) From the earlier theorems, it is not too 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 can have 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 our first central question raised in the Introduction, of why we might want to do feature selection at all. That is, the answer is that if we are using a hypothesis class whose complexity $r_{VC}$ grows superlinearly in $r$ (for example, if $r_{VC} = r^2$) then if it is indeed true that the target concept can be well-represented using only a small subset of the input features, then feature selection

can dramatically reduce the number of training examples needed to learn the target concept well. From looking at the direct bounds on generalization error, a statement can also be made under similar conditions about feature selection giving much lower generalization error. Thus, if we have reason to believe many of the input features are unnecessary or "irrelevant," then these results do (in our opinion) give a very good reason for for performing feature selection.

As for the second central question raised in the Introduction, of how well our algorithms scale with the number of irrelevant features, our bounds also answer this question for the various algorithms. For example, ORDERED-FS has sample complexity (and generalization error, as pointed out earlier) that scales only *logarithmically* with $f$. Thus, from an information theoretic point of view, we may square the number of features, and expect to need only about twice as many training examples. Of course, from a practitioner's point of view, this also means that, particularly 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, and this suggests that it will probably give significantly better performance when indeed only a small fraction of the input features are needed to represent the target concept well.

# Chapter 4

# Experimental Results

Our theoretical results predicted ORDERED-FS to be much more tolerant to the presence of a large number of irrelevant features than STANDARD-WRAP. In this Chapter, we describe our implementations of heuristic search versions these two algorithms, and then our experiments empirically comparing them.

## 4.1 Heuristic search implementation

Our descriptions of STANDARD-WRAP and ORDERED-FS were both "idealized" in that they both required enumeration over all $2^f$ feature subsets which, even for moderate $f$, would be intractable. In practice, we will almost certainly require search heuristics to approximate these enumeration procedures. 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. Recall also that we had mentioned (in a sense to be made rigorous in Chapter 5,) that performance bounds on heuristic versions of these algorithms can be proved to reach our earlier bounds exactly to the extent that we solve these search problems; so in actual implementations of these algorithm, even our bounds suggest it may be worthwhile devoting significant effort to choosing a good search heuristic.

Also recall that ORDERED-FS was originally formulated as requiring $f+1$ separate search problems – one for each value of $r$ – and while it may certainly be implemented that way, there are also rather natural ways for all $f+1$ searches to be carried out "at the same time," as we will shortly describe.

In our experiments, we chose beam search/forward search, which starts out with the empty set of features, and incrementally adds features until we have the full set of features. Forward search is a popular choice that appears to usually do well [24], and beam search, with a beam width of 50 in our case, should be a strict improvement. For STANDARD-WRAP, this is an entirely straightforward application of forward/beam search: We start from the empty set of features, and incrementally add features until we have the full set of features, and pick the hypothesis that, of all the hypotheses we evaluated along the way, gave the lowest hold-out error. For ORDERED-FS, however, each "increment" of the number of features in the hypotheses we are considering actually gives us a new $\hat{h}_r$. This beam-search implementation of ORDERED-FS is given in detail in Figure 1. Thus, while ORDERED-FS was originally described as

$f + 1$ separate searches, it is probably most naturally implemented as carrying out all the searches "together" via a single search in the space of all $2^f$ feature subsets; and of course, our beam search implementation is just one example of an algorithm that does this.

Input: $L$, $S$, $\gamma$, beam width $b$.

1. Randomly split $S$ into training and hold-out test sets $S'$ and $S''$ according to hold-out fraction $\gamma$. Set $\mathcal{H} \leftarrow \emptyset$.
2. Set $r \leftarrow 0$ and $F \leftarrow \emptyset$ (the empty set of features), and set $\mathcal{F} \leftarrow \{F\}$. Add $\hat{h}_0 = L(S'|_F)$ to $\mathcal{H}$.
3. Increment $r$.
4. Set $\mathcal{F}' \leftarrow \emptyset$. Then for each feature subset $F \in \mathcal{F}$ and each feature $F_i$ not already in $F$, add $F \cup \{F_i\}$ to $\mathcal{F}'$.
5. Set $\mathcal{H}' \leftarrow \emptyset$. For each $F \in \mathcal{F}'$, add $L(S'|_F)$ to $\mathcal{H}'$.
6. Let $\hat{h}_r = \text{argmin}_{h \in \mathcal{H}'} \hat{\varepsilon}_{S'}(h)$. Add $\hat{h}_r$ to $\mathcal{H}$
7. Set $\mathcal{F}$ to contain (only) the $b$ feature subsets in $\mathcal{F}'$ that gave (in step 5) the $b$ hypotheses $h$ in $\mathcal{H}'$ with the lowest $\hat{\varepsilon}_{S'}(h)$. (If $|\mathcal{F}'| < b$, set $\mathcal{F} = \mathcal{F}'$.)
8. While $r < f$, loop back to Step 3.
9. Output as our final hypothesis $\text{argmin}_{h \in \mathcal{H}} \hat{\varepsilon}_{S''}(h)$.

Figure 1: Beam search implementation of ORDERED-FS

It is incidentally an interesting fact that if the "$\hat{\varepsilon}_{S'}(h)$" in steps 6 and 7 of our beam-search implementation were modified to "$\hat{\varepsilon}_{S''}(h)$" (which will make us search to minimize hold-out rather than training error), the resulting algorithm is actually equivalent to the beam-search implementation of STANDARD-WRAP. Finally, with the understanding that our heuristic search implementations of STANDARD-WRAP and ORDERED-FS for our experiments used beam search, we will, in the remainder of the Chapter, not distinguish between the "idealized" versions of these two algorithms and their approximate versions.

## 4.2 Experiments

The learning algorithm used was logistic regression [23], 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. Unlike many "real life" problems, all of our input features were independent, and so there were, for example, no complicated interactions between them that could complify the search procedure. All experimental results reported here are averages of 200 independent trials.

For both algorithms, the hold-out fraction $\gamma$ is a parameter that had to be chosen. The analysis of [12] 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 $\gamma$, we obtain Figure 2, 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 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 $\gamma m$ 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 $\gamma$ might alleviate this effect. Doing so, we obtain Figure 3, 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. (On real problems, one may also try $n$-fold cross-validation, and other more sophisticated testing schemes.)

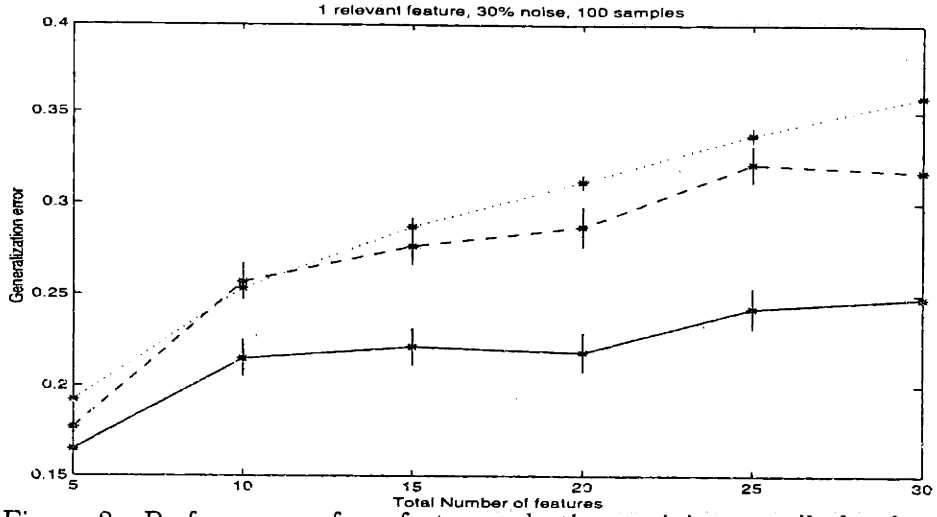


Figure 2: Performance of no feature selection training on all the data (dot), of STANDARD-WRAP (dash) with $\gamma = 0.3$ and ORDERED-FS (solid) with $\gamma = 0.3$. Vertical dashes are 1se.

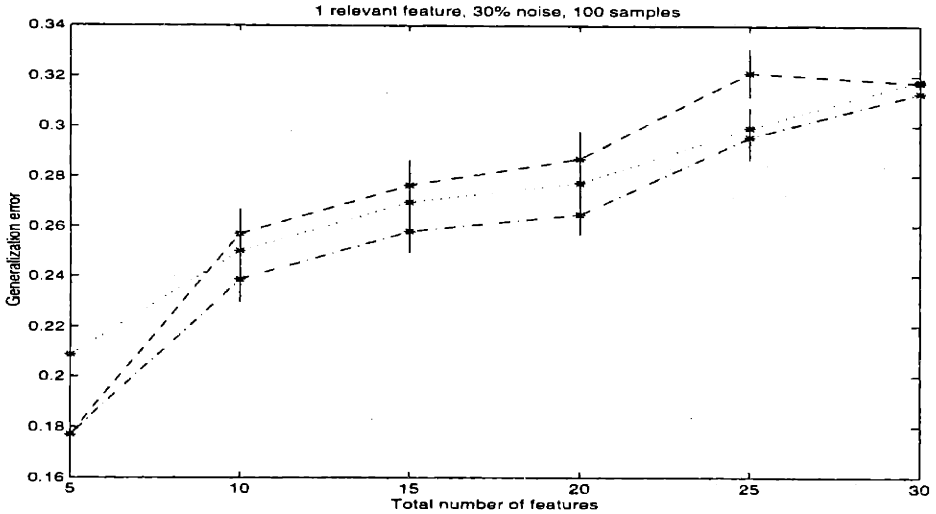


Figure 3: Performance of STANDARD-WRAP using $\gamma = 0.3$ (dash), $\gamma = 0.5$ (dot-dash) and $\gamma = 0.7$ (dot). Vertical dashes are 1se.

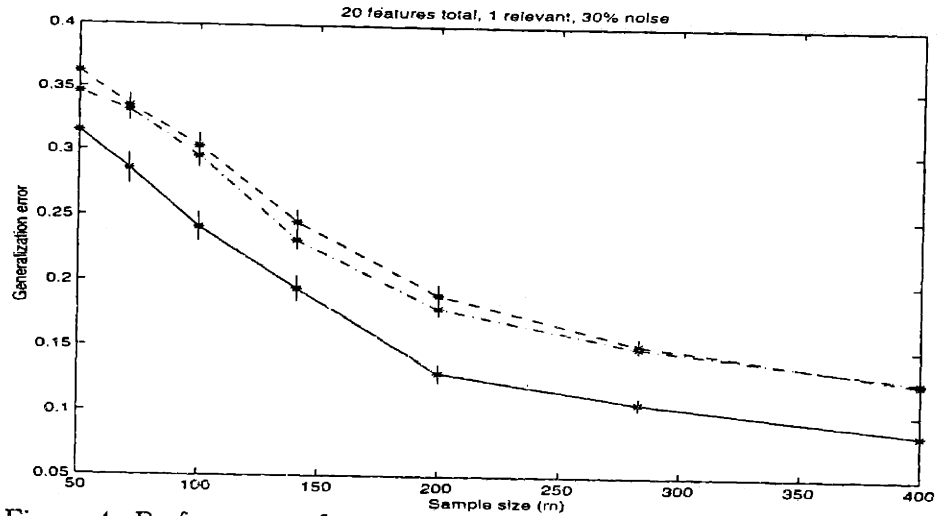


Figure 4: Performance of STANDARD-WRAP with $\gamma = 0.3$ (dash) and $\gamma = 0.5$ (dot-dash), and ORDERED-FS with $\gamma = 0.3$ (solid).

Next, as we vary $m$, the number of training examples, keeping the total number of features at 20, Figure 4 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 5 and 6, which both show ORDERED-FS performing significantly better.

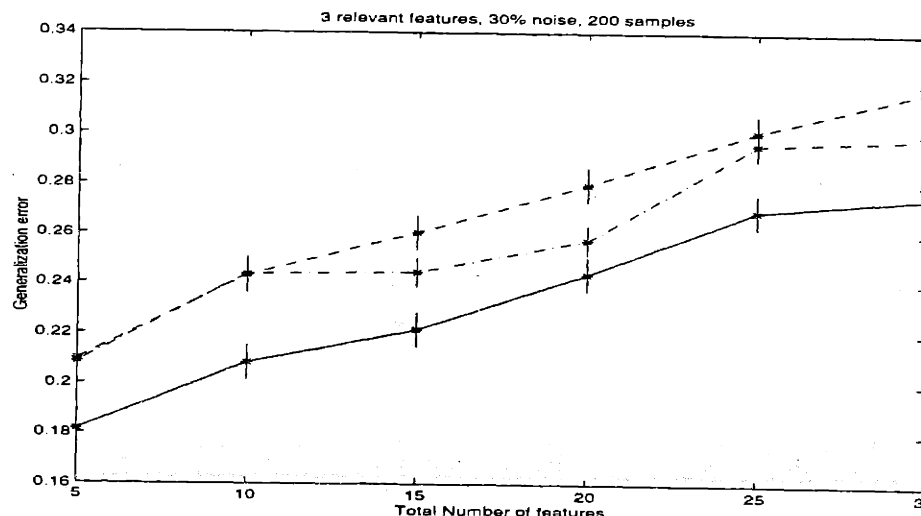


Figure 5: Performance of STANDARD-WRAP and ORDERED-FS. Target has **3** relevant features. (Same legend as Figure 4.)

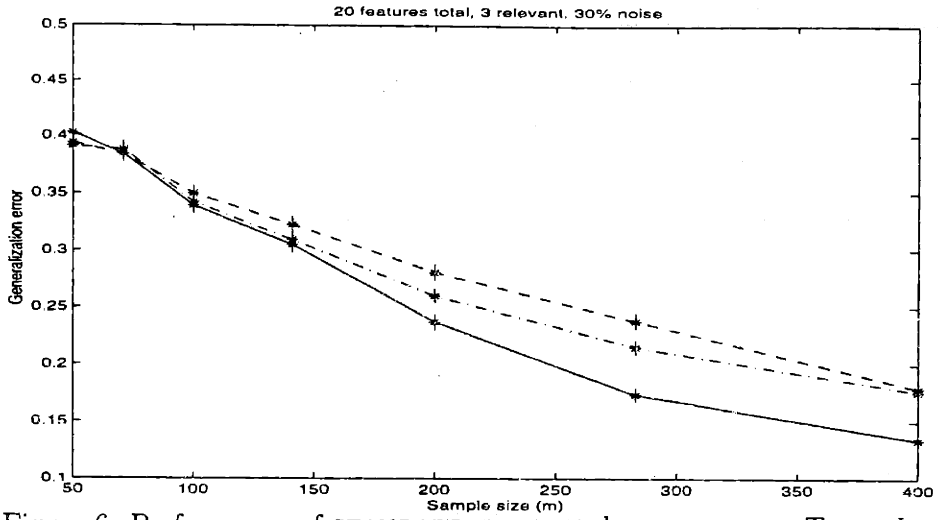

Figure 6: Performance of STANDARD-WRAP and ORDERED-FS. Target has 3 relevant features.

Our (admittedly limited) experimental results showed our heuristic-search version of ORDERED-FS generally beating that of STANDARD-WRAP. While this is encouraging, we of course also 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.) Finally, while more empirical evaluation comparing the two algorithms remains to be done, our experimental results are consistent with our theoretical results, and support the view that ORDERED-FS, as a practical feature selection algorithm, probably does have significantly higher tolerance to the presence of many irrelevant features than STANDARD-WRAP does.

# Chapter 5

# Discussion

In this Chapter, we discuss a number of issues closely related to our theoretical results, and also consider a possible modification to our ORDERED-FS algorithm. First, we have so far proved bounds for STANDARD-WRAP and ORDERED-FS assuming that the relevant search problems are exactly solved, claiming that performance of heuristic-search versions of these algorithms can be proved to reach our bounds exactly to the extent to which their relevant search problems can be solved. In Section 5.1, we make explicit the sense in which we mean this. Also, an interesting question raised in Chapter 3 was how tight the $O(\sqrt{f/\gamma m})$ term in the bound for STANDARD-WRAP is; this is actually a rather tricky issue, and in Section 5.2, we discuss when it may or may not be expected to be tight. Finally, in Section 5.3, we consider a modification to ORDERED-FS that, while maintaining the property of being able to handle exponentially many irrelevant features as training examples, would be more applicable to certain "problematic" feature selection problems, and perhaps also to use with learning algorithms $L$ that do not minimize training error.

## 5.1 Heuristic search versions of STANDARD-WRAP and ORDERED-FS

So far, we have proved bounds assuming the relevant search problems are exactly solved. For STANDARD-WRAP, we needed to find a feature subset out of the $2^f$ feature subsets that resulted in lowest holdout error. This of course becomes an intractable search problem as $f$ becomes even moderately large, and must thus be approximated using search heuristics. One may then ask, suppose we only approximate this minimization problem to some $\varepsilon_+$; that is, suppose our algorithm can only find a feature subset that comes within $\varepsilon_+$ of minimizing hold-out error. What performance bound can we give? One may hope that a small violation of the assumption of exact search would not result in a catastrophic weakening of the bounds, and this is indeed the case.

**Theorem 4** *Given $L, S, \gamma$, a hypothesis $\hat{h}$ output by an approximation to* STANDARD-WRAP *that minimizes hold-out error to within $\varepsilon_+$ will, with high probability, have generalization error bounded by*

$$\varepsilon(\hat{h}) \leq \min_{0 \leq r \leq f} \left\{ \varepsilon_g(r) + O\left( \sqrt{\frac{r_{\mathrm{VC}}}{(1-\gamma)m} \left( \log \frac{m}{r_{\mathrm{VC}}} + 1 \right)} \right) \right\} + O\left( \sqrt{\frac{f}{\gamma m}} \right) + \varepsilon_+ \quad (5.1)$$

The proof is given in the Appendix, but notice how this is exactly the same as our earlier bound in Theorem 2, but with an additional $\varepsilon_+$ term. As for ORDERED-FS, an analogous result can similarly be proved. Recall that, for each value of $r$, we needed to search for a hypothesis that, of all hypotheses using exactly $r$ features, minimizes training error. Now suppose we only approximate the $r$-th search problem to within $\varepsilon_+(r)$. That is, suppose that rather than finding the hypothesis that, of all hypotheses trained using exactly $r$ features, has the smallest training error, we instead find only a feature subset that gives training error only within $\varepsilon_+(r)$ of the minimum (which of course, also means minimizing to within $\varepsilon_+(r)$ training error over *all* hypothesis using exactly $r$ features). We then have the following result:

**Theorem 5** *Given $L, S, \gamma$, the hypothesis $\hat{h}$ output by an approximation to* ORDERED-FS *that, for each value of $r$ finds a $\hat{h}_r$ that minimizes training error to within $\varepsilon_+(r)$, will with high probability have generalization error bounded by*

$$\varepsilon(\hat{h}) \leq$$
$$\min_{0 \leq r \leq f} \left\{ \varepsilon_g(r) + O\left( \sqrt{\frac{r_{\mathrm{VC}}}{(1-\gamma)m} \left( \log \frac{m}{r_{\mathrm{VC}}} + 1 \right)} \right) + O\left( \sqrt{\frac{r}{(1-\gamma)m} \left( \log \frac{f}{r} + 1 \right)} \right) + \varepsilon_+(r) \right\}$$
$$+ O\left( \sqrt{\frac{\log f}{\gamma m}} \right) \quad (5.2)$$

This theorem is also proved in the Appendix. Also, notice again how similar this is to Theorem 3, our bound for when search is performed exactly: this is essentially the same bound, but with the additional $\varepsilon_+(r)$ added to the term we are minimizing over.

Formally, the goal of the search heuristics are to minimize $\varepsilon_+$ for STANDARD-WRAP, or, for the $r$-th search problem in ORDERED-FS, to minimize $\varepsilon_+(r)$. The two Theorems we just gave are formal bounds on generalization error for search heuristics that solve these search problems only to $\varepsilon_+$ or to $\varepsilon_+(r)$, and these bounds can be seen in a very clean way to reach our old (exact search) bounds exactly to the extent that the search heuristics succeed in driving $\varepsilon_+$ and $\varepsilon_+(r)$ to zero. Thus in this sense, search heuristics can be argued to be directly trying to reach the performance given in our earlier bounds, and succeeding exactly to the extent to which they succeed in solving the search problem.

(Nevertheless, 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 [26]. This suggests that in such somewhat-degenerate cases, using a weaker search heuristic may actually be helpful.

In our experiments, we did find parameter ranges that exhibited this effect; but, we do not know how prevalent this effect is in real problems, and would of course recommend using a good optimization criteria, like ORDERED-FS's, rather than using a less-sound criteria and then to ad hocly trying to patch it by doing a poor job in optimizing the less-sound criteria.)

Finally, we have so far skirted the issue of computational expense of (approximately) finding the best (in the training or hold-out error sense) set of features for STANDARD-WRAP and ORDERED-FS. While the results in this section are encouraging in that they give bounds for when we can only approximate the search problems, and in that also give us a nice view of what the search heuristics are trying to do, these results also do not tell us much about how to design good search heuristics, and still do not address what we believe is a rather deep issue of computational expense vs. performance. For example, given a certain amount of computation, what bound can we give on performance? Also, how can we characterize the set of hypotheses reachable with $n$ steps of a particular search algorithm? We are unable to give such issues a rigorous treatment at this time. Nevertheless, we believe, perhaps particularly in view of the results in this section, that much work remains to be done both on these computation vs. performance issues, and also on designing search algorithms for finding feature subsets to minimize training error such as ORDERED-FS requires. (On the latter point, we for example already have very efficient algorithms for performing forward and backward search to minimize training error in linear regression [24], but few generalizations or fast approximations thereof to other algorithms.)

## 5.2 The $O(\sqrt{f/\gamma m})$ term

After presenting our bound for the generalization error of STANDARD-WRAP, we mentioned that while our result is always a rigorous upper bound, the $O(\sqrt{f/\gamma m})$ may also not always be tight. Recall that, via a Chernoff bound argument (mentioned in Chapter 3 and given in detail in the Appendix), testing $N$ hypotheses on an independent hold-out set and picking the one with the lowest empirical error will, with high probability, result in a hypothesis with error no more than $O(\sqrt{(\log N)/\gamma m})$ worse than the best hypothesis in the set. Here, $N = 2^f$, which gave our $O(\sqrt{f/\gamma m})$ term in the upperbound.

How tight is this bound? It is certainly possible to come up with degenerate examples where this bound can be significantly tightened. For example, consider the (extremely degenerate) example of having real input features and where $H_r$ is the set of all linear classifiers that have non-zero coefficients on at most 1 of the $r$ input features. That is, even though a hypothesis $h \in H_r$ has $r$ input features, it ignores all but 1 of them. Then in the pool of $2^f$ hypotheses, there are really only $f+1$ "different hypotheses"[1]; and so, setting $N = f + 1$, the $O(\sqrt{f/\gamma m})$ can really be tightened to

[1]For such a statement to be formally true, we actually need make a number of additional trivial regularization assumptions, but they are neither relevant to our argument nor interesting, and we ignore them here.

$O(\sqrt{(\log f)/\gamma m})$ for this degenerate example. Thus, for a "sufficiently stupid" choice of hypothesis classes, this bound is not tight. Similarly, for a "sufficiently stupid" choice of input features, this bound can also be made loose: For example, using linear classifiers again, suppose that even as $f$ increases, all the input features are identical (that is, $x_1 = x_2 = \cdots$), then no matter how many features there are in total, there are really only 2 "interestingly different" hypotheses in the set of $2^f$ (one which uses 0 features, and one which uses 1 feature), and so the $O(\sqrt{f/\gamma m})$ term can really be tightened to $O(\sqrt{1/\gamma m})$ for this degenerate problem.

From our two examples, one with degenerate hypothesis classes, and one with degenerate features, it appears that any statement about the tightness of this bound must somehow take these two quantities and their interactions into account. For specific hypothesis classes and particular feature subsets, it may be possible to prove tighter bounds (for example, some interesting work has tried to characterize the complexity of hypotheses classes when linear classifiers are used with features that are polynomial terms: that is, given a single input $x_1$, we let the additional features to be $x_2 = x_1^2, x_3 = x_1^3, x_4 = x_1^4 \ldots$ [9]; though note also how restrictive this feature set is as *all* $f$ features jointly span only a 1-dimensional manifold.) But, because of the complicated way in which the tightness of these bounds must take into account and characterize the interaction between hypothesis classes and features, we believe a statement about provably tighter bounds would be difficult to make with any generality. Nevertheless, we do care how this term really behaves, as it is the dominating term in STANDARD-WRAP's bound as the number of irrelevant features becomes large, and we want to know how STANDARD-WRAP's performance scales with the number of irrelevant features. So rather than giving up, let us try to characterize how this term may behave under different additional assumptions, with the warning that our goal here is to gain some understanding of how this quantity really behaves, and that some of the assumptions we make here will include parts that might be slightly conjectural.

We will try to rephrase our problem so that, with a little help from coding theory, we may attempt to characterize the behavior of this term. Recalling that the $O(\sqrt{f/\gamma m})$ bound comes from testing $2^f$ hypotheses on a hold-out set of size $\gamma m$, and that (informally) if the number of "significantly different" hypotheses is much smaller[2], then actual performance may be better than the $O(\sqrt{f/\gamma m})$ term suggests. Again informally, let us try to characterize $N$, the number of "significantly different" hypotheses in the set of $2^f$, with the idea that a $O(\sqrt{(\log N)/\gamma m})$ bound may be the tightest bound we may hope to prove. Note that even if $N = 2^{O(f)}$ so that $N$ may be much smaller than $2^f$, it would still give us an $O(\sqrt{f/\gamma m})$ bound and suggest that our bound is reasonably tight.

Continuing in this informal vein, let us ask then ask how many "significantly different" hypotheses there are in the set of $2^f$, for a "general" training-error-minimizing learning algorithm with non-degenerate input features. Intuitively, we expect two hy-

[2]The notion of the number "significantly different" hypotheses may actually be formalized, for example using something called $\epsilon$-nets (see [10]), but that is not useful to us here, and we will leave it informal for now.

potheses to be "significantly different" if they use significantly different sets of input features, but perhaps not if their sets of input features differ only slightly. Thus, let us assume (with a small leap of faith) that two hypotheses in the set of $2^f$ are "significantly different" if the sets of inputs they use differ by at least some (to be determined) fraction $\beta$ of the total number of features. That is, if the symmetric set difference between the two sets of input features has size at least $\beta f$, then we believe the two hypotheses are "significantly different," where $\beta$ is to be determined. We believe that this will actually give a rather low estimate of the number of "significantly different" hypotheses – for example, the hypothesis using only features 1 and 2 and the hypothesis using only features 5 and 6 are probably "significantly different" as their input feature sets do not even overlap; but since their feature subsets differ only on 4 features, then unless $\beta f \leq 4$, these two hypotheses would not be judged significantly different by this criterion.

Keeping in mind that we may arrive at a lower-than-actual count of the number of significantly different hypotheses, let us now associate each of the $2^f$ hypotheses with an $f$-bit bitstring, that has a one in the $r$-th position if and only if that hypothesis is given the $r$-th feature as an input feature. Then the question of asking how many significantly different hypotheses there are becomes the question of asking how many $f$-bit bitstrings we are able to find, so that each pair of them differs on at least $d = \beta f$ bits. Fortunately, Coding Theory gives at least one lowerbound to this answer. We have:

**Theorem 6 (Gilbert-Varshamov Bound)** *For $f, d > 1$, there exists a set of $N$ length $f$ bitstrings, such that each pair of them has Hamming distance of at least $d$, and where $N$ is lower-bounded by:*

$$N \geq \frac{2^{f-1}}{\binom{f-1}{0} + \binom{f-1}{1} + \binom{f-1}{2} + \cdots + \binom{f-1}{d-2}} \tag{5.3}$$

In Coding Theory, this is actually a lower-bound on the size of a "linear code of length $f$ and distance $d$," and the proof, while not difficult, is also not of particular interest to us. (For details, see any comprehensive Coding Theory textbook, for example [7], which also gives a very readable introduction to the subject.)

Next, applying Proposition 15 (a counting result due to Blumer et. al. [4], given in the Appendix) to upperbound the denominator ($\leq (e(f-1)/(d-2))^{d-2}$) and then taking logs, we have, as $f$ becomes large:

$$\begin{aligned}
\ln N &\geq (f-1)\ln 2 - (d-2)\left(1 + \ln\frac{f-1}{d-2}\right) \\
&= (f-1)\ln 2 - (\beta f - 2)\left(1 + \ln\frac{f-1}{\beta f - 2}\right) \\
&\approx (f-1)\ln 2 - (\beta f - 2)\left(1 + \ln\frac{1}{\beta}\right) \\
&\geq f\left(\ln 2 - \beta\left(1 + \ln\frac{1}{\beta}\right)\right) - \ln 2
\end{aligned} \tag{5.4}$$

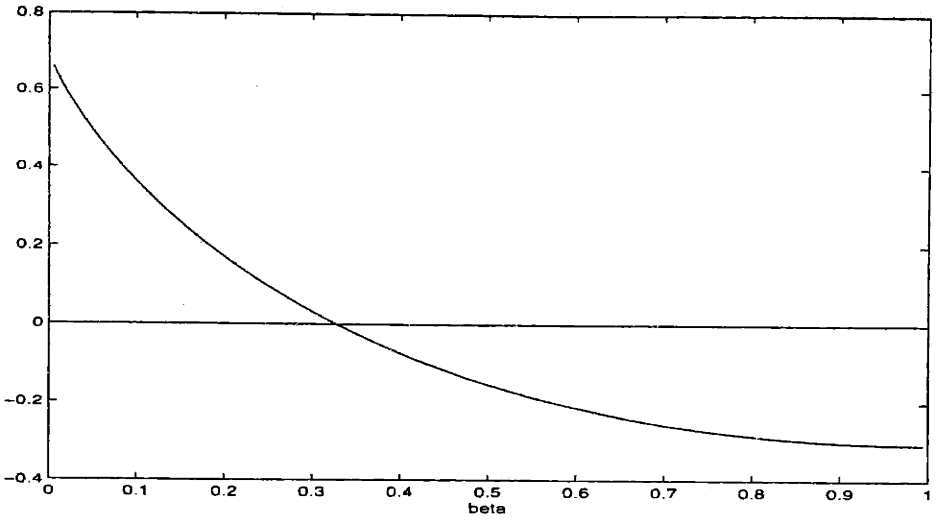


Figure 7: Plot of $\ln 2 - \beta\left(1 + \ln\frac{1}{\beta}\right)$ term vs. $\beta$.

Plotting, as a function of $\beta$, the $\ln 2 - \beta\left(1 + \ln\frac{1}{\beta}\right)$ term multiplying into $f$ (see Figure 7), we see that for all about $\beta \leq 0.32$ it is strictly positive. Thus, for all $\beta \leq 0.32$, $\ln N = O(f)$, or $N = 2^{O(f)}$, and so $O(\sqrt{(\log N)/\gamma m}) = O(\sqrt{f/\gamma m})$, meaning that our $O(\sqrt{f/\gamma m})$ bound may quite reasonably be expected to be tight (in the big $O$ sense).

To summarize, suppose we take any pair of hypotheses to be "significantly different" so long the symmetric set difference between the sets of features they use has size at least $0.3f$. Note that, since half of all of our hypotheses have $\leq 0.5f$ input features in total, 0.3 seems (in our opinion) a reasonably large fraction of the input features to require to be different in order for a pair of hypotheses to be considered "significantly different." (In fact, by negating all the bits in our $N$ bitstrings if necessary, it is also easy to show that if we take just the bitstrings with $\leq 0.5f$ ones, we can actually find a set of at least $N/2$ "significantly different" hypotheses, each of which uses at most $0.5f$ features.[3]) Then, within the set of $2^f$ hypothesis, we can find a subset of $2^{O(f)}$ hypotheses that are pairwise "significantly different," which suggests the $O(\sqrt{f/\gamma m})$ term would be a generally tight bound in such cases.

As a closing comment for this section, it is of course important to remember that all we have done here is characterize a set of behaviors under which we may expect the $O(\sqrt{f/\gamma m})$ term to be tight. While we believe these behaviors to cover a very wide range of learning scenarios, our assumptions were also justified only intuitively, including our assumption of $\beta < 0.32$ behavior. Future work may well come up with better characterizations of the tightness of this bound (and we had in fact conjectured earlier that $O(\sqrt{f^\alpha/\gamma m})$, $\alpha \in (0, 1]$ behavior may also be possible). But if we believe that the assumptions we made here are reasonable, then for now, it also seems reasonable to believe from this result that our bound will "often" be

[3]Though note also that, since we are using symmetric set differences, given two hypotheses using $0.5f$ features each, we require only 30% of each hypothesis' input features to be excluded from the other hypothesis' set, and not 60%, for them to be "significantly different."

tight (in the big-$O$ sense) for general feature selection problems with non-degenerate learning algorithms and features.

## 5.3 An algorithm for difficult feature subsets and non-error minimizing $L$s

There are certain (possibly non-error minimizing) learning algorithms $L$ and "difficult" feature subsets that, when used in combination with ORDERED-FS, give bad generalization error. Let us remain with error-minimizing $L$ for now, and consider, as an example, a medical diagnosis task where we are given features of a patient, and asked to predict if the patient has a disease.[4] Suppose all the features are binary (say, whether the patient exhibits each of a number of symptoms) except for one last feature, which is a unique patient identification number. Also, suppose that the learning algorithm finds the smallest decision tree over the training set so as to minimize training error. Then since each patient has a unique ID, so long as the ID is one of the features given to the learning algorithm, it would be able to (rather stupidly) choose a function over that patient ID, that gives zero training error. (Note this also implies that $r_{VC} = \infty \; \forall r \geq 1$, meaning our earlier bounds for the performance of ORDERED-FS, while still being rigorous bounds, hold vacuously.)

Of course, the problem here is our assumption in Section 2.3 that all features are treated "uniformly" by the learning algorithm, whereas the patient ID feature is somehow different in that it gives $L$ much more representation power. In this example, if the target concept can be well-represented using a "small" number of features other than the patient ID, then we would still like our feature selection algorithm to do well. Here, we do not want to assume $L$ "uniformly" treats features, so rather than letting $r_{VC}$ characterize the complexity of *any* hypothesis classes using $r$ features, let us make a finer gradation, and let $F_{VC}$ be the VC dimension of the hypothesis class $H_F$. (The reader may wish to refer to Section 2.3 to refresh his/her memory on some of these definitions.) Then, consider the following algorithm, which we simply call ORDERED-FS2:

1. Randomly split the training set $S$ into a training set $S'$ and 2 hold-out sets $S''$ and $S'''$, of sizes $\gamma_1 m, \gamma_2 m$, and $\gamma_3 m$ respectively, $\gamma_1, \gamma_2, \gamma_3 > 0, \gamma_1 + \gamma_2 + \gamma_3 = 1$.

2. For each value of $r$, find the feature subset $F$ of exactly size $r$ and hypothesis $\hat{h} = L(S'|_F)$ so that hold-out test error on $S''$, that is $\hat{\varepsilon}_{S''}(\hat{h})$, is minimized. Let $\hat{h}_r$ be this hypothesis.

3. Of the hypotheses in the set $\{\hat{h}_0, \hat{h}_1, \ldots, \hat{h}_f\}$, pick and output the one with the lowest empirical error on the second hold-out set $S'''$.

Then using proof techniques completely analogous to those used in Chapter 3 and the Appendix, and letting $\varepsilon_g(F)$ denote the least generalization error achievable by

[4] This example was suggested to the author by one of the anonymous reviewers of an earlier version of this work [27] submitted to the *Fifteenth International Conference on Machine Learning.*

any hypothesis using the feature subset $F$ (and in the hypothesis class $H_F$), one can easily show the following bound on this algorithm's generalization error:

**Theorem 7** *Given $S, \gamma_1, \gamma_2, \gamma_3$, and error-minimizing $L$, the hypothesis $\hat{h}$ output by* ORDERED-FS2 *will, with high probability, have generalization error bounded by:*

$$\varepsilon(\hat{h}) \leq \min_F \left\{ \varepsilon_g(F) + O\left(\sqrt{\frac{F_{\text{VC}}}{\gamma_1 m}\left(\log\frac{m}{F_{\text{VC}}}+1\right)}\right) + O\left(\sqrt{\frac{|F|}{\gamma_2 m}\left(\log\frac{f}{|F|}+1\right)}\right)\right\} + O\left(\sqrt{\frac{\log f}{\gamma_3 m}}\right) \tag{5.5}$$

Also recall that in Chapter 3, we had given bounds on sample complexity assuming the target concept was well-approximated using some $r^*$ features. To give an analogous result, we now need to assume instead that there is a small set of features $F^*$ with which the target concept can be well represented (and such that $F_{VC}$ is hopefully small – meaning, in our medical example, that $F$ should not contain the patient ID feature). Then, under very similar conditions as before, ORDERED-FS2 has sample complexity bounded by:

$$O\left(F^*_{VC} + |F^*| \log f\right) \tag{5.6}$$

(Compare this with the $O(r^*_{VC} + r^* \log f)$ bound we had for ORDERED-FS.) As for implementation, like ORDERED-FS, our ORDERED-FS2 algorithm would probably also need to be implemented via a search heuristic. Though from a practical point of view, a slightly unappealing aspect of it is that it requires saving aside two rather than just one hold-out set. This does not weaken the theoretical results much, but when actually performing feature selection, then if one believes the feature selection problem at hand to have features that really cannot be treated "uniformly" (meaning $H_F$ varies much more with $F$ than only through $|F|$), we would recommend first trying to recode the input features to ameliorate this problem. If that fails, then we suggest one may turn to ORDERED-FS2, but otherwise, we do find ORDERED-FS to be the ascetically more pleasing algorithm.

Before closing this section, one more comment is that ORDERED-FS2 will also handle non-error-minimizing algorithms to some extent (though again, unless $L$ is "significantly non-error-minimizing," we recommend that the practitioner try ORDERED-FS first). Rather than devoting too much space developing the notation to prove the result formally, we will simply state it informally here. Suppose there is a small feature subset so that $L$ applied to learning on just that feature subset does well. Then with high probability, ORDERED-FS2 will find a hypothesis that does "nearly as well" as $L$ applied to just this feature subset (using $\gamma_1 m$ training samples), where "nearly as well" means that we have "additional" generalization error bounded by $O\left(\sqrt{\frac{|F|}{\gamma_2 m}\left(\log\frac{f}{|F|}+1\right)}\right) + O\left(\sqrt{\frac{\log f}{\gamma_3 m}}\right)$. (Note there is no longer a mention of VC-dimension here, as $L$ is no longer an error-minimizing algorithm so that there is no longer a notion of minimizing training error over a hypothesis class of bounded complexity, and $L$ may now even do its own clever regularization/feature selection.)

# Chapter 6

# Conclusions

In the Introduction, we raised two questions that we believed to be central to the entire notion of performing feature selection: First, why should we perform feature selection at all, when learning algorithms are generally able to learn, by themselves, to ignore the "irrelevant" features? And second, since feature selection is about removing irrelevant features, how well do or can feature selection algorithms scale with the number of irrelevant features? To answer the first question, we showed in Chapter 3 that the very natural feature selection algorithm STANDARD-WRAP can indeed dramatically reduce the sample complexity needed to learn, when the target concept can be represented using only a small number of the input features. But, its sample complexity still scaled about linearly with the number of irrelevant features, and we wanted to do better. By looking at the form of the bound for STANDARD-WRAP, we developed a new feature selection algorithm, ORDERED-FS, that has sample complexity that scales only logarithmically with the number of irrelevant features. Similar to STANDARD-WRAP, it required for us to be able to approximately solve a large search problem in order to achieve such performance, and we also showed in Chapter 4 how there are some very natural ways to implement search heuristics for it, and presented encouraging experimental results on an artificial domain, that showed ORDERED-FS significantly beating STANDARD-WRAP as the number of irrelevant features became large. Finally, Chapter 5 also discussed some of the finer points of our bounds, including what happens when STANDARD-WRAP and ORDERED-FS are approximated with search heuristics.

As for results of this work, we have answered our two questions, the first answer of which gives us theoretical reassurance for performing feature selection and which we hope may encourage more practitioners to use feature selection when it is appropriate, and the second of which also gave us the ORDERED-FS algorithm, which has rather strong theoretical properties. While much empirical evaluation comparing ORDERED-FS and other feature selection algorithms remains to be done, it is also true that, from a practical stand-point, ORDERED-FS's information-theoretic property of having sample complexity logarithmic in the number of irrelevant features is a very strong one, previously shared by only a small number of sometimes-brittle algorithms for restricted hypothesis classes. Such logarithmic sample complexity means, for example, being able to square the total number of features and expect to need only about twice as many samples to learn. Unfortunately, it does require good search or optimization algorithms to realize this performance fully, and the development

of such search heuristics is still very much an open question. Nevertheless, careful human-design of features can often be an arduous and time-consuming task, and so we believe that if this logarithmic sample complexity is even only approximately realized by search algorithms, then it may for example have tremendous consequences for feature design in actual applications of supervised machine learning – indeed, it would imply that modulo computational expense, overly careful human design of features is often not necessary, so long as additional training data can be obtained reasonably cheaply.

# Appendix A

# Detailed Proofs of Theorems

In this Appendix, we give an overview of a small number of results from Computational Learning Theory that will required to understand and derive our earlier theorems, and give detailed versions of our proofs. While the proof sketches in the main body of the Thesis were intended for readers familiar with Computational Learning Theory, the material in this appendix is specifically written to be accessible to readers with little or no background in Computational Learning Theory (though while not necessary, some intuition about VC dimension would help). Thus, if the reader is already familiar with Chernoff bounds and with Vapnik's work on uniform convergence, he/she should be warned that this Appendix may seem somewhat longwinded to them, and that the proof sketches given in Chapter 3 may be more appropriate.

For an introduction to Computational Learning Theory outside of this thesis, we highly recommend the excellent textbook by Kearns and Vazirani [10], which gives a clear and succinct introduction to the topic. Another, older and slightly more narrow, but also very readable introduction is Anthony and Biggs [2]. Also, in this work, we rely heavily on the work of Vapnik and Chervonenkis, and some of the theorems we quote are proved in [29]. Finally, some of the proof techniques we use were derived from those used in Kearns [12] and Kearns, Mansour, Ng and Ron [13], which the reader may also find to be of interest.

## A.1 Chernoff bounds and error from hold-out testing

First, we state without proof a fundamental result generally referred to as the Chernoff bounds (this particular form of which is widely attributed to Hoeffding), that bounds the probability that the empirical mean of a sample of $i.i.d.$ Bernoulli random variables will deviate significantly from the true mean:

**Theorem 8 (Chernoff bounds)** *Suppose $x_1, x_2, \ldots, x_m$ are $m$ samples drawn $i.i.d.$ from a Bernoulli(p) distribution. Let $\overline{x} = \frac{1}{m}\sum x_i$ be their mean, and $\chi \geq 0$ be any non-negative constant. Then*

$$\Pr[\overline{x} > p + \chi] \leq e^{-2\chi^2 m} \tag{A.1}$$

$$\Pr[\overline{x} < p - \chi] \leq e^{-2\chi^2 m} \tag{A.2}$$

Putting Equations (A.1) and (A.2) together, the probability of deviation from the mean by $\chi$ in either the positive or negative directions is therefore at most $\Pr[|\overline{x}-p| > \chi] \leq 2\exp(-2\chi^2 m)$. Now, given a fixed hypothesis $h$, if we draw a training set $S$ of $m$ (uncorrupted) *i.i.d.* training examples, the Chernoff bound therefore immediately allows us to bound the probability that $\hat{\varepsilon}_S(h)$ would be far from $\varepsilon(h)$, by viewing the training sample as $m$ independent draws of the random variable with mean $\varepsilon(h)$, that is 1 whenever $h$ makes a mistake on a randomly drawn input. Applying the Chernoff bound, we therefore have, for any $\chi \geq 0$:

$$\Pr[|\hat{\varepsilon}_S(h) - \varepsilon(h)| > \chi] \leq 2e^{-2\chi^2 m} \tag{A.3}$$

In short, this is saying that if we are trying to evaluate any one fixed hypothesis, then when we try to estimate its generalization error using an independent test set of size $m$, the probability that our estimate has error greater than any fixed $\chi$ decreases exponentially in $m$.

Next, let us ask, suppose we have a set $H = \{h_1, h_2, \ldots, h_N\}$ of $N$ fixed hypotheses rather than just one, and again draw a set $S$ of $m$ training samples and evaluate their empirical errors $\hat{\varepsilon}_S(h_1), \ldots, \hat{\varepsilon}_S(h_N)$ using $S$. Can we then upper-bound the probability that any *one* (or more) of our estimates will deviate from the true generalization error by more than $\chi$? Since, for a fixed $i$, the probability that $|\hat{\varepsilon}_S(h_i) - \varepsilon(h_i)| > \chi$ is upper-bounded by $2\exp(-2\chi^2 m)$, the probability that any one of them will deviate by more than $\chi$ is upper-bounded by $N$ times this quantity[1]:

$$\Pr[|\hat{\varepsilon}_S(h_1) - \varepsilon(h_1)| > \chi \text{ or } \ldots \text{ or } |\hat{\varepsilon}_S(h_N) - \varepsilon(h_N)| > \chi] \leq 2Ne^{-2\chi^2 m} \tag{A.4}$$

Letting the left hand side (probability of deviation of any one of $\hat{\varepsilon}_S(h_i)$ from $\varepsilon(h_i)$ by more than $\chi$) be $\delta$ and solving for $\chi$, we have:

$$\chi \geq \sqrt{\frac{1}{2m} \ln \frac{2N}{\delta}} \tag{A.5}$$

That is, for any $\chi$ satisfying this inequality, we may assert with probability at least $1 - \delta$ that all $N$ of the $\hat{\varepsilon}_S(h_i)$s are within $\chi$ of the $\varepsilon(h_i)$s. Clearly, we get the tightest bound by setting the inequality in Equation (A.5) to an equality.

This will shortly give us our first lemma, also used in Kearns [12] and Kearns et. al [13] (and many other places), to bound the error introduced by selecting one hypothesis out of a set $H$ of $N$ hypotheses by picking the one with the lowest hold-out test error on an independently drawn hold-out set of size $m$. For notational convenience, let $h^* = \arg\min_{h \in H} \varepsilon(h)$ be the best hypothesis in the set $H$, and $\hat{h} = \arg\min_{h \in H} \hat{\varepsilon}_S(h)$ be the hypothesis chosen via our hold-out testing. We have already shown that, with high probability (meaning probability at least $1-\delta$ for any fixed $\delta > 0$), all $N$ estimates of empirical error will be close (within $\chi$) to the true generalization

[1]This result comes from a simple fact commonly called the Union Bound, and simply reflects the fact that $\Pr[A_1 \text{ or } A_2 \text{ or } \ldots \text{ or } A_N] \leq \Pr[A_1] + \Pr[A_2] + \cdots + \Pr[A_N]$ for any set of events $A_1, \ldots, A_N$. Here, $A_i$ is the event that $|\hat{\varepsilon}_S(h_i) - \varepsilon(h_i)| > \chi$.

errors – that is, that the $\hat{\varepsilon}_S(h_i)$s will uniformly converge to the $\varepsilon(h_i)$s. Using this, we can show that no hypothesis "much worse" than the $h^*$ could have lower empirical error than $h^*$. Letting $\chi = \sqrt{(1/m)\ln(2N/\delta)}$, we have shown that with probability at least $1-\delta$, we may assert $|\hat{\varepsilon}_S(h_i) - \varepsilon(h_i)| \leq \chi$ simultaneously for all $i = 1 \ldots N$. Thus, with probability at least $1-\delta$:

$$\begin{aligned} \varepsilon(\hat{h}) &\leq \hat{\varepsilon}_S(\hat{h}) + \chi & \text{(A.6)} \\ &\leq \hat{\varepsilon}_S(h^*) + \chi & \text{(A.7)} \\ &\leq \varepsilon(h^*) + 2\chi & \text{(A.8)} \end{aligned}$$

where the first and last inequalities used $|\hat{\varepsilon}_S(h) - \varepsilon(h)| \leq \chi$ holding for all $h \in H$ (in particular, for $h = \hat{h}$ and $h = h^*$) and the second inequality used the fact that $\hat{\varepsilon}_S(\hat{h}) \leq \hat{\varepsilon}_S(h)$ for all $h \in H$ (because $\hat{h}$ was specifically chosen to minimize $\hat{\varepsilon}_S(\cdot)$). Substituting back $\chi$, we therefore have the following lemma:

**Lemma 9** *Given a fixed set $H = \{h_1, h_2, \ldots, h_N\}$ of $N$ hypotheses, let us draw an independent hold-out test set $S$ of size $m$. Let $h^* = \arg\min_{h \in H} \varepsilon(h)$ be the hypothesis in $H$ with the lowest generalization error, and $\hat{h} = \arg\min_{h \in H} \hat{\varepsilon}_S(h)$ be hypothesis chosen by standard hold-out testing. Then with probability at least $1-\delta$,*

$$\varepsilon(\hat{h}) \leq \varepsilon(h^*) + 2\sqrt{\frac{1}{2m}\ln\frac{2N}{\delta}} \qquad \text{(A.9)}$$

Moreover, since $\varepsilon(h^*) \leq \varepsilon(h)$ for all $h \in H$, we also have the following Corollary (which, for any fixed $h$, is merely a weakening of our lemma):

**Corollary 10** *Under the conditions of Lemma 9, we may assert with probability $1-\delta$ that simultaneously for all $h \in H$,*

$$\varepsilon(\hat{h}) \leq \varepsilon(h) + 2\sqrt{\frac{1}{2m}\ln\frac{2N}{\delta}} \qquad \text{(A.10)}$$

We will use this (reasonably well-known) result to prove some of our results later, but before continuing, it is worth summarizing our argument as we will shortly use something very much like it again. We first showed that, with probability at least $1-\delta$, we may simultaneously assert, for all $h \in H$, that the empirical error $\hat{\varepsilon}_S(h)$ of $h$ is within $\chi(\delta, m)$ of its generalization error $\varepsilon(h)$. This was all that was needed in Equations (A.6) to (A.8) to show that, with probability at least $1-\delta$, the hypothesis $\hat{h}$ selected by hold-out testing on an independent set of size $m$ would have generalization error no more than $2\chi(\delta, m)$ worse than any other hypothesis $h$ in the set $H$. When we uniformly bound the deviation of training from generalization error, the same type of argument will give us a bound of the generalization error of the hypothesis chosen by minimizing training error.

## A.2 Learning without feature selection

The proof of the bound on generalization error when learning without feature selection is due to Vapnik and Chervonenkis. First, we state without proof the a result (proved in [29]) that will shortly lead to the bound.

**Theorem 11 (Vapnik and Chervonenkis, 1971)** *Suppose we are given a training set $S$ of size $m$ and a hypothesis class $H_F$ of hypotheses defined over the input space $X$ unrestricted to any feature subset. If $f_{\text{VC}} < m$, then for any fixed $\delta > 0$, we may with probability at least $1 - \delta$ assert simultaneously for all $h \in H_F$:*

$$|\varepsilon(h) - \hat{\varepsilon}_S(h)| < 2\sqrt{\frac{f_{\text{VC}}}{m}\left(\ln\frac{2m}{f_{\text{VC}}} + 1\right) + \frac{1}{m}\ln\frac{9}{\delta}} \tag{A.11}$$

Purely as a notational convenience, we want to drop the $f_{VC} < m$ restriction, and so letting $\ln_+ x$ denote $\max\{\ln x, 0\}$, we have:

**Corollary 12** *Suppose we are given a training set $S$ of size $m$ and a hypothesis class $H_F$ of hypotheses defined over the input space $X$ unrestricted to any feature subset. Then for any fixed $\delta > 0$, we may with probability at least $1 - \delta$ assert simultaneously for all $h \in H_F$:*

$$|\varepsilon(h) - \hat{\varepsilon}_S(h)| < 2\sqrt{\frac{f_{\text{VC}}}{m}\left(\ln_+\frac{2m}{f_{\text{VC}}} + 1\right) + \frac{1}{m}\ln\frac{9}{\delta}} \tag{A.12}$$

(because $|\varepsilon(h) - \hat{\varepsilon}_S(h)| \leq 1$ always since $\varepsilon(h)$ and $\hat{\varepsilon}_S(h)$ are in $[0, 1]$, and so the inequality hold vacuously whenever $f_{VC} \geq m$).

Hence, with high probability, the deviation between empirical error on $S$ and generalization error is again uniformly bounded by $\chi' = 2\sqrt{\frac{f_{VC}}{m}\left(\ln_+\frac{2m}{f_{VC}} + 1\right) + \frac{1}{m}\ln\frac{9}{\delta}}$. Thus, using exactly the same argument as that used earlier in Equations (A.6)–(A.8), we see that, when we use a training-error minimizing learning algorithm $L$ (that, by definition, picks $L(S) = \arg\min_{h \in H_F} \hat{\varepsilon}_S(h)$), we have the following bound, which is the full version of Theorem 1 that we had presented earlier.

**Theorem 13 (Vapnik and Chervonenkis, 1971)** *Given a training set $S$ of size $m$ and training-error minimizing $L$, the hypothesis $\hat{h} = L(S)$ output by $L$ will, with probability at least $1 - \delta$, have generalization error bounded by*

$$\varepsilon(\hat{h}) \leq \varepsilon_g(f) + 4\sqrt{\frac{f_{\text{VC}}}{m}\left(\ln_+\frac{2m}{f_{\text{VC}}} + 1\right) + \frac{1}{m}\ln\frac{9}{\delta}} \tag{A.13}$$

## A.3 Learning with STANDARD-WRAP feature selection

In this section, we give the full version of Theorem 2. and prove it by applying the proof technique given in [12] (used to bound the error of hold-out) to feature selection.

**Theorem 14** *Given $L, S, \gamma$, the hypothesis $\hat{h}$ output by* STANDARD-WRAP, *given by $\hat{h} = L(S'|_{\hat{F}})$ where $\hat{F} = \text{argmin}_F\ \hat{\varepsilon}_{S''}(L(S'|_F))$, will, with probability at least $1-\delta$, have generalization error bounded by*

$$\varepsilon(\hat{h}) \leq \min_{0\leq r\leq f} \left\{ \varepsilon_g(r) + 4\sqrt{\frac{r_{\text{VC}}}{(1-\gamma)m}\left(\ln_+ \frac{2(1-\gamma)m}{r_{\text{VC}}} + 1\right) + \frac{1}{(1-\gamma)m}\ln\frac{18}{\delta}} \right\}$$
$$+2\sqrt{\frac{1}{2\gamma m}\left((f+2)\ln 2 + \ln\frac{1}{\delta}\right)} \qquad \text{(A.14)}$$

**Proof:** Conceptually, we think of STANDARD-WRAP as doing the following: First, go through all $2^f$ possible feature subsets $F_1, \ldots, F_{2^f}$, running $L$ on $S'$ restricted to each of them, and using them to form a "pool" $\mathcal{H} = \{h_i | h_i = L(S'|_{F_i})\}$ of $2^f$ hypotheses. Next, use the holdout set to pick our final hypothesis $\hat{h} = \arg\min_{h\in\mathcal{H}} \hat{\varepsilon}_{S''}(h)$.

Now, fix any $r$, $0 \leq r \leq f$, and let $F_r^*$ be the feature subset that contains a hypothesis with generalization error $\varepsilon_g(r)$. (Recall $\varepsilon_g(r)$ is the best generalization error achievable by any hypothesis using exactly $r$ features; here, we let $F_r^*$ be a feature subset that realizes this.) Then in the pool of hypotheses $\mathcal{H}$, there is some hypothesis $\hat{h}_r^* = L(S'|_{F_r^*})$ that was trained using this feature subset. Applying Theorem 13 to using a training-error minimizing learning algorithm $L$ on a training set of size $(1-\gamma)m$ on the hypothesis class $H_r$ of VC-dimension $r_{VC}$, we have that, with probability at least $1-\delta/2$,

$$\varepsilon(\hat{h}_r^*) \leq \varepsilon_g(r) + 4\sqrt{\frac{r_{VC}}{(1-\gamma)m}\left(\ln_+ \frac{2(1-\gamma)m}{r_{VC}} + 1\right) + \frac{1}{(1-\gamma)m}\ln\frac{18}{\delta}} \qquad \text{(A.15)}$$

Hence, with probability at least $1-\delta/2$, the hypothesis $\hat{h}_r^*$, which is in the pool of hypotheses $\mathcal{H}$, has generalization error bounded by $\varepsilon_g(r)$ plus the square root term above. Next, we apply Corollary 10 to show that using the hold-out set $S''$ to choose one of the $2^f$ hypotheses will probably result in choosing a hypothesis not much worse than $\hat{h}_r^*$. Letting $h$ in the Corollary be $\hat{h}_r^*$, and $N = 2^f$, we have that, with probability at least $1-\delta/2$:

$$\begin{aligned}\varepsilon(\hat{h}) &\leq \varepsilon(\hat{h}_r^*) + 2\sqrt{\frac{1}{2\gamma m}\ln\frac{2(2^f)}{\delta/2}} \\ &= \varepsilon(\hat{h}_r^*) + 2\sqrt{\frac{1}{2\gamma m}\left((f+2)\ln 2 + \ln\frac{1}{\delta}\right)} \qquad \text{(A.16)}\end{aligned}$$

Now, each of Equation (A.15) and Equation (A.16) holds with probability at least $1 - \delta/2$. Thus, by the Union Bound ($\Pr[A_1 \text{ and } A_2] = 1 - \Pr[(\neg A_1) \text{ or } (\neg A_2)] \geq 1 - \Pr[\neg A_1] - \Pr[\neg A_2]$) we may assert with probability at least $1 - \delta$ that they simultaneously hold, in which case substituting Equation (A.15) into (A.16) gives

$$\begin{aligned} \varepsilon(\hat{h}) \leq \varepsilon_g(r) \quad &+ \quad 4\sqrt{\frac{r_{VC}}{(1-\gamma)m}\left(\ln_+ \frac{2(1-\gamma)m}{r_{VC}} + 1\right) + \frac{1}{(1-\gamma)m}\ln\frac{18}{\delta}} \\ &+ \quad 2\sqrt{\frac{1}{2\gamma m}\left((f+2)\ln 2 + \ln\frac{1}{\delta}\right)} \end{aligned} \tag{A.17}$$

Finally, we have proved this for any fixed $0 \leq r \leq f$. Since all the quantities on the right hand side of Equation (A.17) are deterministic, it must in particular hold with probability at least $1 - \delta$ for the value of $r$ that minimizes the right hand side. Thus, we finally take a min over $r$, which gives us the statement of the theorem. □

## A.4 Learning with ORDERED-FS feature selection

In this section, we give our bound on the generalization error under ORDERED-FS feature selection, which as shown earlier, depends only logarithmically in $f$. As a quick outline, we will first bound the generalization error of the hypothesis $\hat{h}_r$ as $\varepsilon_g(r)$ plus other terms that grow only logarithmically in $f$. Then, applying Corollary 10 again to bound the error from the hold-out testing, we arrive at our final bound. Before proceeding, we state without proof the following result due to Blumer et. al. [4]:

**Proposition 15 (Blumer et. al., 1989)** *For all $f \geq r \geq 1$:*

$$1 + \binom{f}{1} + \binom{f}{2} + \cdots + \binom{f}{r} \leq \left(\frac{ef}{r}\right)^r \tag{A.18}$$

The proof is not of particular interest to us (based mainly on a proof by induction), and it is not a vital piece of our proof (in essence, what this does is allow us to have a slightly tighter $O\left(\sqrt{\frac{r}{(1-\gamma)m}(\ln\frac{f}{r}+1)}\right)$ term in our bound, rather than the $O(\sqrt{\frac{r\ln f}{(1-\gamma)m}})$ term we would get if we were to use the looser and (but trivial to show) $\binom{f}{r} \leq f^r$,) and it is in fact the following weakening of it (dropping all but the last term on the left hand side, then taking logs) that we will use:

**Corollary 16** *For all $f \geq r \geq 0$:*

$$\ln\binom{f}{r} \leq r\left(\ln\frac{f}{r} + 1\right) \tag{A.19}$$

(Note however that we now require only $r \geq 0$ rather than $r \geq 1$. Also, we adopt the usual convention, based on taking limits, that "$0\ln\frac{1}{0} = 0$" for when $r = 0$.) We are now ready to state and prove our bound for learning with ORDERED-FS.

**Theorem 17** *Given $L, S, \gamma$, the hypothesis $\hat{h}$ output by* ORDERED-FS *will, with probability at leat $1-\delta$, have generalization error bounded by*

$$\varepsilon(\hat{h}) \leq \min_{0 \leq r \leq f} \left\{ \varepsilon_g(r) + 4\sqrt{\frac{r_{VC}}{(1-\gamma)m}\left(\ln_+ \frac{2(1-\gamma)m}{r_{VC}} + 1\right) + \frac{r\left(\ln \frac{f}{r} + 1\right) + \ln \frac{18}{\delta}}{(1-\gamma)m}} \right\}$$
$$+2\sqrt{\frac{1}{2\gamma m} \ln \frac{4(f+1)}{\delta}} \qquad \text{(A.20)}$$

**Proof:** Conceptually, what ORDERED-FS does is first, for each each $0 \leq r \leq f$, find the hypothesis $\hat{h}_r$ that, of all the hypotheses using exactly $r$ features, minimizes error on the training set $S'$, and puts them into a pool of hypotheses $\mathcal{H} = \{\hat{h}_0, \ldots, \hat{h}_f\}$. Then, it evaluates all $f+1$ hypotheses in $\mathcal{H}$ on the hold-out set $S''$, and picks the one with the smallest hold-out error.

Again, fix any $0 \leq r \leq f$. Now, for any *fixed* feature subset $F$ of size $r$, applying Theorem 11, we may assert with probability at least $1 - \delta/2\binom{f}{r}$ that simultaneously for all hypotheses $h \in H_F$ (that is, all hypotheses using exactly these $r$ features in $F$),

$$|\varepsilon(h) - \hat{\varepsilon}_{S'}(h)| < 2\sqrt{\frac{r_{VC}}{(1-\gamma)m}\left(\ln_+ \frac{2(1-\gamma)m}{r_{VC}} + 1\right) + \frac{1}{(1-\gamma)m} \ln \frac{18\binom{f}{r}}{\delta}} \qquad \text{(A.21)}$$

Note this holds with probability at least $1 - \delta/2\binom{f}{r}$ for any one fixed feature subset among the $\binom{f}{r}$ feature subsets of size exactly $r$. Thus, taking a Union bound again, it must simultaneously hold for all $\binom{f}{r}$ feature subsets of size $r$ with probability at least $1 - \delta/2$. Thus, with probability at least $1 - \delta/2$, we have uniformly bounded the deviation of training error (on $S'$) from generalization error of all hypotheses using exactly $r$ features. Now, $\hat{h}_r$ is the hypothesis that, of all hypotheses using exactly $r$ features, minimizes training error on $S'$. Thus, letting $\chi''$ be the term on the right hand size of Equation A.21 and using the same argument as given in Equations A.6 to A.8 before (where $\hat{h}$ there would be replaced with $\hat{h}_r$, and $\varepsilon(h^*)$ would be replaced with $\varepsilon_g(r)$), we have the result that, with probability at least $1 - \delta/2$, we may assert

$$\varepsilon(\hat{h}_r) \leq \varepsilon_g(r) + 4\sqrt{\frac{r_{VC}}{(1-\gamma)m}\left(\ln_+ \frac{2(1-\gamma)m}{r_{VC}} + 1\right) + \frac{1}{(1-\gamma)m} \ln \frac{18\binom{f}{r}}{\delta}} \qquad \text{(A.22)}$$

Next, we are using the hold-out set $S''$ of size $\gamma m$ to test $f+1$ hypotheses, and pick our final hypothesis as $\hat{h} = \arg\min_{h \in \mathcal{H}} \hat{\varepsilon}_{S''}(h)$. Thus, applying Corollary 10, we may assert with probability at least $1 - \delta/2$

$$\begin{aligned}\varepsilon(\hat{h}) &\leq \varepsilon(\hat{h}_r) + 2\sqrt{\frac{1}{2\gamma m}\ln\frac{2(f+1)}{\delta/2}} \\ &= \varepsilon(\hat{h}_r) + 2\sqrt{\frac{1}{2\gamma m}\ln\frac{4(f+1)}{\delta}} \qquad \text{(A.23)}\end{aligned}$$

Each of Equations (A.22) and (A.23) holds with probability at least $1-\delta/2$. Taking a Union bound again, with probability at least $1-\delta$, they must hold simultaneously, in which case substituting Equation (A.22) into (A.23) gives

$$\begin{aligned}\varepsilon(\hat{h}) \leq \varepsilon_g(r) &+ 4\sqrt{\frac{r_{VC}}{(1-\gamma)m}\left(\ln_+\frac{2(1-\gamma)m}{r_{VC}}+1\right)+\frac{1}{(1-\gamma)m}\ln\frac{18\binom{f}{r}}{\delta}} \\ &+ 2\sqrt{\frac{1}{2\gamma m}\ln\frac{4(f+1)}{\delta}} \qquad \text{(A.24)}\end{aligned}$$

Finally, we have again proved this for any fixed $0 \leq r \leq f$, and since all the quantities on the right hand side of Equation (A.20) are deterministic, it must in particular hold with probability at least $1-\delta$ for the value of $r$ that minimizes the right hand side. We finally therefore take a min over $r$, which, combined with Corollary 16 to bound the $\binom{f}{r}$ term, gives the statement of the theorem. □
(This bound still looks slightly different from the one given in big $O(\cdot)$ notation in Chapter 3, but the latter is obtained simply by observing that, for non-negative $a, b$, $O(\sqrt{a+b}) = O(\sqrt{a}+\sqrt{b})$, so we may break up the first square-root term to obtain Theorem 3.)

## A.5 Extensions: Noisy training examples, and approximate searches.

The theorems proved so far in this appendix were all for learning from noiseless examples, and when the relevant search problem is performed exactly. Regarding the assumption of noiseless examples, the Computational Learning Theory literature has come up with many interesting learning algorithms that can formally be proved to learn quickly, but whose performance proofs critically depend on the assumptions of having noiseless training examples, and of being able to *exactly* represent the target concept using the hypothesis class. (Examples abound, from learning rectangles to decision lists to DNF formulae; see [10] for example.) Such algorithms may fail catastrophically when just a tiny amount of training label noise is present, or when the target concept cannot be exactly represented by our hypothesis class. There has of course also been much work on building learning algorithms that are robust to noise, for example in the Statistical Query learning model [11], and work on proving worst-case bounds. Nevertheless, a lack of robustness to noisy examples does tend to

make a number of learning algorithms, while of significant theoretical interest, still unusable for most real-life problems. Thus, in designing our algorithms, we would like to verify they are indeed robust to training label noise. In this section, what we consider to be of interest is mostly a verification that our algorithms are robust to noise rather than the exact form of the bounds, and so we will only very briefly demonstrate how bounds can be proved when the training labels we are provided are noisy.

Also, for STANDARD-WRAP and ORDERED-FS, we have so far been considering the idealization where the relevant search problem is performed exactly. In this section, we also show robustness in the sense that if the search problems can only be approximated by a search heuristic, then similar results still hold, and a bound can be proved on generalization error, that reaches our earlier bounds exactly to the extent that the search problem can be solved. As discussed in more detail in Chapter 5, this means that, in a formal sense (modulo some other issues also discussed in Chapter 5,) the search heuristics can be interpreted as directly trying to reach the performance given in our earlier bounds.

### A.5.1 Learning with noisy examples

In this section, we consider the case of learning from noisy examples, and show, via reasonably standard techniques (also used in [12] and elsewhere), that very similar bounds as those proved earlier hold. As a quick outline, we will first make a few definitions related to the notion of "generalization error with respect to noisy samples." Then, while we had previously uniformly bounded the deviation of training error from generalization error to prove our bounds in the noiseless case, we now uniformly bound the deviation of training error (which is of course now with respect to noisy samples) from generalization error with respect to noisy samples; this will then immediately allow us to derive our formal bounds for the noisy case.

To start, rather than assuming uncorrupted training examples, let us suppose that the training labels have been independently corrupted at some 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 $\eta$, where $c(\cdot)$ is, as before, the target concept we are trying to learn. (See Section 2.3.) Evaluation is still with respect to uncorrupted data. Also, let $c_\eta(x)$ be a stochastic function of $x$, so that $c_\eta(x) = c(x)$ with probability $1 - \eta$, and $c_\eta(x) = 1 - c(x)$ with probability $\eta$. Since $c_\eta(\cdot)$ is essentially the concept-class function but with noisy labels, we shall call it the noisy concept class. Next, we define *noisy generalization error*, or generalization error with respect to the noisy concept class, as the probability that $h$ apparently misclassifies a noisy training example:

$$\varepsilon_\eta(h) = \Pr_{x \in D_X}[h(x) \neq c_\eta(x)]. \tag{A.25}$$

Clearly, noisy generalization error $\varepsilon_\eta(h)$ satisfies the following relationship with our old notion of generalization error $\varepsilon(h)$ (with respect to uncorrupted data):

$$\varepsilon_\eta(h) = (1 - \eta)\varepsilon(h) + \eta(1 - \varepsilon(h)) \tag{A.26}$$

which comes simply from observing that $h$ will misclassify a noisy sample ($h(x) \neq c_\eta(x)$) under two cases: First, with probability $1-\eta$, the label output by $c_\eta$ is correct, so $h$ apparently misclassifies $x$ exactly when it truly misclassifies it ($h(x) \neq c(x)$), which happens with probability $\varepsilon(h)$. Second, with probability $\eta$, the label output by $c_\eta$ is incorrect, which means $h$ apparently misclassifies $x$ exactly when it truly classifies it correctly ($h(x) = c(x)$). Expanding terms and rearranging, we also have

$$\varepsilon_\eta(h) = (1-2\eta)\varepsilon(h) + \eta \tag{A.27}$$

We are now ready to repeat our argument of Section A.1 for the case of using noisy examples, and our argument will show how the proofs in Sections A.2–A.4 can similarly be generalized. As before, suppose we have a fixed hypothesis $h$, and want to estimate $\varepsilon_\eta(h)$, by drawing a set $S$ of $m$ noisy examples, and taking our estimate of $\varepsilon_\eta(h)$ to be the empirical error $\hat{\varepsilon}_S(h)$ of $h$ on the set of $m$ examples. Then, noticing that "whether or not $h(x) = c_\eta(x)$" is a Bernoulli random variable, applying the Chernoff bound gives us, for any $\chi \geq 0$:

$$\Pr[|\hat{\varepsilon}_S(h) - \varepsilon_\eta(h)| > \chi] \leq 2e^{-2\chi^2 m} \tag{A.28}$$

Note how similar this is to Equation A.3. Similarly, repeating the argument from Equation A.3 up to Equation A.10 but with $\varepsilon(h)$ everywhere replaced by $\varepsilon_\eta(h)$ (that is, since we now have noisy training examples, we now want to uniformly bound the deviation of empirical error from noisy generalization error,) we then have the following result, informally stated for now: Given a fixed set of $N$ hypotheses, picking the one $\hat{h}$ with the lowest empirical error on a noisy hold-out set of size $m$ results in a hypothesis $\hat{h}$ that, with at least probability $1-\delta$, satisfies

$$\varepsilon_\eta(\hat{h}) \leq \varepsilon_\eta(h^*) + 2\sqrt{\frac{1}{2m}\ln\frac{2N}{\delta}} \tag{A.29}$$

where, also as before, $h^*$ is the hypothesis in the set of $N$ that has the smallest generalization error. Note the similarity between this and Equation 10. Substituting in $\varepsilon_\eta(\hat{h}) = (1-2\eta)\varepsilon(\hat{h}) + \eta$ and $\varepsilon_\eta(h^*) = (1-2\eta)\varepsilon(h^*) + \eta$ from Equation (A.27), we have, with probability at least $1-\delta$:

$$(1-2\eta)\varepsilon(\hat{h}) + \eta \leq (1-2\eta)\varepsilon(h^*) + \eta + 2\sqrt{\frac{1}{2m}\ln\frac{2N}{\delta}} \tag{A.30}$$

which, by cancelling $\eta$s rearranging terms, easily gives us the following:

**Lemma 18** *Given a fixed set $H = \{h_1, h_2, \ldots, h_N\}$ of $N$ hypotheses, let us draw an independent hold-out test set $S$ of size $m$ that has labels independently corrupted at noise rate $\eta \in [0, 0.5)$. Let $h^* = \arg\min_{h\in H} \varepsilon(h)$ be the hypothesis $H$ with the lowest generalization error, and $\hat{h} = \arg\min_{h\in H} \hat{\varepsilon}_S(h)$ be the hypothesis chosen by standard hold-out testing. Then with probability at least $1-\delta$,*

$$\varepsilon(\hat{h}) \leq \varepsilon(h^*) + 2\sqrt{\frac{1}{2(1-\eta)^2 m}\ln\frac{2N}{\delta}} \tag{A.31}$$

This Lemma, of course, generalizes Lemma 9. Recapitulating, in the noiseless case, we had proved our bounds by uniformly bounding the deviation of empirical error from generalization error by some factor $\chi$, and then, via Equations (A.6)-(A.8), given a bound for generalization error of $\hat{h}$ as being no more than $2\chi$ worse than the best possible. Here in the case of noisy data, we instead uniformly bound the deviation of empirical error from *noisy* generalization error, which then allows us to bound generalization error of $\hat{h}$ as being no more than $2\chi/(1-\eta)$ worse than the best possible. Very similar arguments can be used generalize the rest of our bounds.

Applying Theorem 11 similarly, we also have its the following slightly revised version of the theorem, that given a training set $S$ of size $m$ with labels independently corrupted at noise rate $\eta \in [0, 0.5)$, and a hypothesis class $H_F$, then with probability at least $1-\delta$, we may assert simultaneously for all $h \in H_F$:

$$|\hat{\varepsilon}_S(h) - \varepsilon_\eta(h)| < 2\sqrt{\frac{f_{VC}}{m}\left(\ln_+ \frac{2m}{f_{VC}} + 1\right) + \frac{1}{m}\ln\frac{9}{\delta}} \tag{A.32}$$

Having thus uniformly bounded the deviation of training from noisy generalization error, we may again, via our earlier argument, bound the error of the output hypothesis when learning without feature selection:

**Theorem 19** *Given a training set $S$ of size $m$ with labels independently corrupted at noise rate $\eta \in [0, 0.5)$, and training-error minimizing $L$, the hypothesis $\hat{h} = L(S)$ output by $L$ will, with probability at least $1-\delta$, have generalization error bounded by*

$$\varepsilon(\hat{h}) \leq \varepsilon_g(f) + \frac{4}{1-2\eta}\sqrt{\frac{f_{VC}}{m}\left(\ln_+ \frac{2m}{f_{VC}} + 1\right) + \frac{1}{m}\ln\frac{9}{\delta}} \tag{A.33}$$

Finally the proofs for STANDARD-WRAP and ORDERED-FS learning from noisy data are completely analogous, and the reader should be able to derive them quite easily via the method we have used here. Rather than stating and proving the bounds formally, we simply give them without proof. Generalizing Theorem 14 for STANDARD-WRAP, we have, with probability at least $1-\delta$:

$$\varepsilon(\hat{h}) \leq \min_{0\leq r\leq f} \left\{\varepsilon_g(r) + \frac{4}{1-2\eta}\sqrt{\frac{r_{VC}}{(1-\gamma)m}\left(\ln_+ \frac{2(1-\gamma)m}{r_{VC}} + 1\right) + \frac{1}{(1-\gamma)m}\ln\frac{18}{\delta}}\right\}$$
$$+\frac{2}{1-2\eta}\sqrt{\frac{1}{2\gamma m}\left((f+2)\ln 2 + \ln\frac{1}{\delta}\right)} \tag{A.34}$$

And generalizing Theorem 17 for ORDERED-FS, we have, with probability at least $1-\delta$:

$$\varepsilon(\hat{h}) \leq \min_{0\leq r\leq f} \left\{\varepsilon_g(r) + \frac{4}{1-2\eta}\sqrt{\frac{r_{VC}}{(1-\gamma)m}\left(\ln_+ \frac{2(1-\gamma)m}{r_{VC}} + 1\right) + \frac{r\left(\ln\frac{f}{r}+1\right) + \ln\frac{18}{\delta}}{(1-\gamma)m}}\right\}$$

$$+\frac{2}{1-2\eta}\sqrt{\frac{1}{2\gamma m}\ln\frac{4(f+1)}{\delta}} \tag{A.35}$$

### A.5.2 Approximating search

Briefly, we will show in this section that a search heuristic implementation of STANDARD-WRAP and ORDERED-FS that can only approximate the relevant search problem has performance that can be bounded by a bound quite similar to our earlier ones, and which reaches our earlier bounds exactly to the extent that the search problems can be solved.

For STANDARD-WRAP, a search heuristic is used to try to find a feature subset so that training using that feature subset gives low error on the hold-out test set. We will be using notation defined in Section A.3 (in particular, the definition of $\mathcal{H}$,) the reader may wish to briefly peruse that section if he/she has not already done so. Now, suppose the search heuristic can approximate this search problem to $\varepsilon_+$; that is, rather than finding a hypothesis $\hat{h}$ that satisfies $\hat{\varepsilon}_{S''}(\hat{h}) \leq \hat{\varepsilon}_{S''}(h)$ for all hypotheses $h \in \mathcal{H}$, only satisfies $\hat{\varepsilon}_{S''}(\hat{h}) \leq \hat{\varepsilon}_{S''}(h) + \varepsilon_+$. Now as shown in Section A.3, with probability at least $1-\delta/2$, it is simultaneously true for all $h \in \mathcal{H}$ that

$$|\varepsilon(\hat{h}) - \hat{\varepsilon}_{S''}(\hat{h})| \leq \sqrt{\frac{1}{2\gamma m}\left((f+2)\ln 2 + \ln\frac{1}{\delta}\right)} \tag{A.36}$$

Letting $\chi$ denote the right side of the above inequality, and letting $\hat{h}^*_r$ be as defined in Section A.3 (basically the hypothesis trained using the "best" feature subset of size exactly $r$), then for any fixed $r$, we have, with probability at least $1-\delta/2$:

$$\begin{aligned}
\varepsilon(\hat{h}) &\leq \hat{\varepsilon}_{S''}(\hat{h}) + \chi && \text{(A.37)}\\
&\leq \hat{\varepsilon}_{S''}(\hat{h}^*_r) + \varepsilon_+ + \chi && \text{(A.38)}\\
&\leq \varepsilon(\hat{h}^*_r) + \varepsilon_+ + 2\chi && \text{(A.39)}
\end{aligned}$$

where the first and last inequalities used $|\varepsilon(h) - \hat{\varepsilon}_{S''}(h)| \leq \chi$, and the second inequality used $\hat{\varepsilon}_{S''}(\hat{h}) \leq \hat{\varepsilon}_{S''}(\hat{h}^*_r) + \varepsilon_+$. Note the similarity between these and Equations (A.6)-(A.8). Combining this with Equation A.15 to bound $\varepsilon(\hat{h}^*_r)$, substituting back in the definition of $\chi$, and taking a min over $r$ as before then gives us the following theorem:

**Theorem 20** *Given $L, S, \gamma$, a hypothesis $\hat{h}$ output by an approximation to* STANDARD-WRAP *that minimizes hold-out error to within $\varepsilon_+$, will with probability at least $1-\delta$, have generalization error bounded by*

$$\begin{aligned}
\varepsilon(\hat{h}) \leq \min_{0\leq r\leq f} \quad & \left\{\varepsilon_g(r) + 4\sqrt{\frac{r_{\mathrm{VC}}}{(1-\gamma)m}\left(\ln_+\frac{2(1-\gamma)m}{r_{\mathrm{VC}}} + 1\right) + \frac{1}{(1-\gamma)m}\ln\frac{18}{\delta}}\right\}\\
& + 2\sqrt{\frac{1}{2\gamma m}\left((f+2)\ln 2 + \ln\frac{1}{\delta}\right)} + \varepsilon_+ \qquad \text{(A.40)}
\end{aligned}$$

For ORDERED-FS, a very similar result can be proved using very similar techniques. Recall that, for each value of $r$, we needed to find a feature subset that, of all hypotheses using exactly $r$ features, minimizes training error. For each value of $r$, this involves a search over $\binom{f}{r}$ feature subsets. Suppose we can only approximate this, and that we can only approximate the $r$-th search problem to within $\varepsilon_+(r)$. That is, rather than finding the hypothesis that, of all hypotheses trained using exactly $r$ features, has the smallest training error, suppose we find only a feature subset that gives training error only within $\varepsilon_+(r)$ of the minimum (which of course, also means minimizing training error over *all* hypothesis using exactly $r$ features, to within $\varepsilon_+(r)$). Then, from Section A.4 Equation (A.21) for any fixed $r$, it holds true with probability at least $1-\delta/2$, that simultaneously for all hypotheses $h$ using exactly $r$ features:

$$|\varepsilon(h) - \hat{\varepsilon}_{S'}(h)| < 2\sqrt{\frac{r_{VC}}{(1-\gamma)m}\left(\ln_+ \frac{2(1-\gamma)m}{r_{VC}} + 1\right) + \frac{1}{(1-\gamma)m}\ln\frac{18\binom{f}{r}}{\delta}} \quad \text{(A.41)}$$

Letting $\chi'$ be the right side of the above and again applying an argument like that in Equations (A.37)-(A.39), we have, for any fixed value of $r$, that with probability at least $1-\delta$:

$$\varepsilon(\hat{h}_r) \leq \varepsilon_g(r) + 2\chi' + \varepsilon_+(r) \quad \text{(A.42)}$$

Note the similarity between this and Equation (A.22). Thus, substituting back the definition of $\chi'$ and, as before, applying the usual bound on the error from using $\gamma m$ hold-out test samples (Equation (A.23)), and finally taking a min over $r$, we have the following theorem:

**Theorem 21** *Given $L, S, \gamma$, the hypothesis $\hat{h}$ output by an approximation to* ORDERED-FS *that, for each value of $r$ finds a $\hat{h}_r$ that minimizes training error to within $\varepsilon_+(r)$, will with probability at least $1-\delta$ have generalization error bounded by*

$$\varepsilon(\hat{h}) \leq$$

$$\min_{0\leq r\leq f}\left\{\varepsilon_g(r) + 4\sqrt{\frac{r_{VC}}{(1-\gamma)m}\left(\ln_+\frac{2(1-\gamma)m}{r_{VC}}+1\right) + \frac{r\left(\ln\frac{f}{r}+1\right)+\ln\frac{18}{\delta}}{(1-\gamma)m}} + \varepsilon_+(r)\right\}$$

$$+2\sqrt{\frac{1}{2\gamma m}\ln\frac{4(f+1)}{\delta}} \quad \text{(A.43)}$$

# Bibliography


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

[2] Martin Anthony and Norman Biggs. *Computational Learning Theory.* Cambridge University Press, 1992.

[3] Avrim Blum and Pat Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97:245–271, 1997.

[4] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth. Learnability and the vapnik-chervonenkis dimension. *Journal of the ACM*, 36(4):929–965, 1989.

[5] Richard Caruana and Dayne Frietag. Greedy attribute selection. In *Proceedings of the Eleventh International Conference on Machine Learning.* Morgan Kaufmann, 1994.

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

[7] D. G. Hoffman, D. A. Leonard, C. C. Linder, K. T. Phelps, C. A. Rodger, and J. R. Wall. *Coding Theory: The Essentials.* Monographs and textbooks in pure and applied mathematics. Marcel Dekker Inc., 1991.

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

[9] Marek Karpinski and Thorsten Werther. VC dimension and uniform learnability of sparse polynomials and rational functions. *SIAM Journal on Computing*, 22(6):1276–1285, 1993.

[10] Michael Kearns and Umesh V. Vazirani. *An Introduction to Computational Learning Theory.* MIT Press, 1994.

[11] Michael J. Kearns. Efficient noise-tolerant learning from statistical queries. In *Proceedings of the 25th ACM Symposium on the Theory of Computing*, pages 392–401. ACM Press, 1993.

[12] Michael J. Kearns. 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, 1996.

[13] Michael J. Kearns, Yishay Mansour, Andrew Y. Ng, and Dana Ron. An experimental and theoretical comparison of model selection methods. *Machine Learning Journal*, 27(1):7–50, 1997.

[14] Michael J. Kearns and Dana Ron. 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, 1997.

[15] K. Kira and L. Rendell. A practical approach to feature selection. In *Proceedings of the Ninth International Conference on Machine Learning*. Morgan Kaufmann, 1992.

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

[17] R. Kohavi and D. Sommerfield. 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*, 1995.

[18] Ron Kohavi and George H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97:273–324, 1997.

[19] Daphne Koller and Mehran Sahami. Towards optimal feature selection. In *Proceedings of the Thirteenth International Conference on Machine Learning*. Morgan Kaufmann, 1996.

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

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

[22] Nick Littlestone and Manfred Warmuth. The weighted majority algorithm. *Information and Computation*, 108:212–261, 1994.

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

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

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

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

[27] Andrew Y. Ng. On Feature Selection: Learning with exponentially many irrelevant features as training examples. In *Proceedings of the Fifteenth International Conference on Machine Learning (to appear)*, pages 404–412. Morgan Kaufmann, 1998.

[28] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 25(11):1134–1142, 1984.

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

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

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

[32] Yiming Yang and Jan O. Pederson. A comparative study on feature selection in text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning.* Morgan Kaufmann, 1997.