Multiple Comparisons in Induction Algorithms

David Jensen and Paul R. Cohen
Experimental Knowledge Systems Laboratory
Department of Computer Science
Box 34610 LGRC
University of Massachusetts
Amherst, MA 01003-4610
413-545-3613
{jensen|cohen}@cs.umass.edu

Abstract

A single mechanism is responsible for three pathologies of induction algorithms: attribute selection errors, overfitting, and oversearching. In each pathology, induction algorithms compare multiple items based on scores from an evaluation function and select the item with the maximum score. We call this a *multiple comparison procedure* (*MCP*). We analyze the statistical properties of *MCPs* and show how failure to adjust for these properties leads to the pathologies. We also discuss approaches that can control pathological behavior, including Bonferroni adjustment, randomization testing, and cross-validation.

Keywords
Inductive learning, overfitting, oversearching, attribute selection, hypothesis testing, parameter estimation

Running Head
Multiple Comparisons

1 Introduction

This paper defines and analyzes *multiple comparison procedures* ($MCPs$). $MCPs$ are ubiquitous in induction algorithms as well as other AI algorithms. $MCPs$ have important statistical properties, and failure to adjust for these properties produces three pathologies of induction algorithms — attribute selection errors, overfitting, and oversearching.

The contribution of this work is to identify a single statistical mechanism underlying these pathologies. All induction algorithms implicitly or explicitly make statistical inferences, but nearly all make them incorrectly. Understanding why these inferences are incorrect explains the pathologies themselves, identifies potential solutions, and explains why previously proposed solutions have succeeded and failed.

2 An Example

Before discussing $MCPs$ in induction algorithms, let's begin with an analogy:

Suppose you are deciding whether to hire an investment advisor. This person's job will be to predict whether the stock market will close up or down on any given day. You hope to avoid hiring a charlatan — someone whose predictions are no better than chance. To evaluate a candidate, you devise a test: the candidate will make predictions for the next 14 days, and if 11 or more predictions are correct, you will conclude that the candidate is not a charlatan. The threshold of 11 is chosen because, if there is a 0.50 probability of a charlatan predicting correctly on any one day, there is only a 0.0287 probability that he or she will predict correctly on 11 of the next 14 days. Therefore, you reason, if a candidate passes the eleven-or-more test, he probably is not charlatan, and the chances of making a mistake by hiring him are no more than 0.0287.

Applied to only a single candidate, your logic is impeccable. However, what if you gather ten candidates, record each of their predictions for 14 days, select the candidate with largest number of correct predictions, and then apply the test to that candidate? A test on just one candidate has a 0.0287 chance of producing an error, but the overall probability an error depends on the number of candidates, n, and is 0.0287 only if $n = 1$. When $n > 1$, *each* charlatan has a 0.0287 probability of passing the test and, in general, the probability of selecting a charlatan is no greater than $1 - (1 - .0287)^n$. If $n = 10$, the probability is no greater than 0.253. By not adjusting for the number of candidates, you underestimate by roughly an order of magnitude the probability that *at least one of them* (or alternatively, *the best of them*) will pass the eleven-or-more test. Given a sufficiently large pool of charlatans, you can practically guarantee that at least one of them will achieve *any* performance level, but this doesn't mean the candidate in question is performing better than chance.

3 Multiple Comparison Procedures and Statistical Inferences

Many induction algorithms make inferences that are directly analogous to deciding whether to hire an investment advisor. We discuss three instances of such inferences in section 4, but to understand the analogy, let's analyze the investment advisor example in more detail.

The decision to hire an investment advisor can be divided into two parts: selecting the top-scoring candidate and inferring whether that candidate is performing better than chance. Selecting the top-scoring candidate uses a multiple comparison procedure (MCP):

Multiple comparison procedure (MCP)

1. *Generate n items* — Find n candidates.
2. *Calculate a score x for each item* using an evaluation function f and data sample $\mathcal{S}$ — Calculate a score for each candidate where f is the number of correct predictions and $\mathcal{S}$ is the past fourteen days of stock market activity. That is, $x_i = f(candidate_i, \mathcal{S})$.
3. *Select the item with the maximum score x_{max}* — Select the candidate with the largest number of correct predictions.

Any score x_i is inherently statistical because it is based on a particular data sample $\mathcal{S}$, and different samples will produce different scores. In statistical terms, x_i is a specific value of a random variable X_i. X_i is defined by the evaluation function f, the item being evaluated, the size of the sample, and the population from which data samples are drawn. For a given f and item, the values x_i for all possible samples of size $|\mathcal{S}|$ from a given population define the *sampling distribution* of X_i. Similarly, x_{max} is a specific value of a random variable, X_{max}, but X_{max} is defined by *all* the n items examined, not just a single item. The sampling distribution of X_{max} depends on n, the number of items examined.

This difference between X_i and X_{max} is critical to making two types of inferences based on the score x_{max}. The example illustrates the first type: using x_{max} to infer whether the top-scoring candidate is a charlatan. To make this inference, we compare x_{max} to a sampling distribution generated under the assumption that a single candidate is performing at a chance level, that is, we compare x_{max} to the sampling distribution for X_i. If x_{max} is very unlikely to have been drawn from that sampling distribution, we can conclude that the advisor is probably not a charlatan. As indicated in the example, using the sampling distribution of X_i will generally underestimate the probability of selecting a charlatan. The correct sampling distribution is for X_{max}, and that distribution depends on n.

The second type of inference can be illustrated by supposing that you and a friend are both selecting investment advisors. You evaluate the performance of 10 candidates, and your friend evaluates 30 candidates. Can you compare the score of your best candidate with the score of your friend's best candidate?

Suppose that all the candidates are charlatans, and thus no advisor is better than another. What is the probability that each top-scoring candidate will predict correctly for 11 or more of the 14 days? In your case, the probability is no greater than 0.253, but in your friend's case, the probability is more than twice that: $1 - (1 - .0287)^{30} = 0.583$. Merely by examining more candidates, your friend is more likely to find one with a given score for the past 14 days, even though all the candidates perform at a chance level. In general, if the number of candidates you evaluate (n_1) differs from the number of candidates your friend evaluates (n_2), the performance of the top-scoring candidates (x_{max_1} and x_{max_2}, respectively) are not directly comparable because they are drawn from different sampling distributions.

This problem particularly acute if we use x_{max} as an estimate of the true, long-run score for the candidate. This long-run score is called the *population* score, and x_{max} is generally a poor estimate of it. Suppose, as is quite likely, that your friend's top-scoring candidate passed our test and predicted correctly on 11 of the 14 days. Based on this sample performance, we might infer that, on the population, he will predict correctly more than three-quarters of the time ($11/14 = 0.786$). We would be mistaken, however, because your friend's top-scoring candidate is a charlatan, just like all the others, and his actual probability of a correct prediction is only 0.50.

Both types of inferences are inherently statistical. The first is a problem of statistical hypothesis testing. We wish to answer a yes-no question about a candidate ("Are a candidate's predictions better than chance?") based on a sample score. The second is a problem of parameter estimation. We wish to estimate the value of a population (i.e., long-run) score based on a sample score so we can accurately compare candidates ("What proportion of the time will a candidate predict correctly?"). In both cases, the scores are calculated from a data sample $\mathcal{S}$ so they are inherently statistical, regardless of whether statistical techniques are explicitly used. In both cases, using the score x_{max} introduces special problems of statistical inference.

4 Induction Algorithms and Pathologies

The example of the investment advisor is directly relevant to induction algorithms. Many algorithms use $MCPs$ and then make implicit or explicit statistical inferences based on the score x_{max}. Rather than examining advisors and their stock predictions for a given two-week period, induction algorithms examine models and their predictions for a given training set. In nearly all cases, induction algorithms do not adjust for the number of items n when making

inferences.[1]

For example, induction algorithms use $MCPs$ to decide which of several variables to use in a model component (e.g., which variable to use at a node in a decision tree), to decide whether to add a component to an existing model (e.g., whether to add a term to a linear regression equation), and to select among several different models. In each of these contexts, empirical studies have revealed an associated pathology — *attribute selection error*, *overfitting*, and *oversearching*, respectively. Each pathology occurs because of incorrect statistical inferences given the score x_{max}. In one case — overfitting — the inferences can be viewed as statistical hypothesis tests. In the two other cases — attribute selection errors and oversearching — the inferences can be viewed as parameter estimates.

Below, we formally describe these pathologies and highlight their essential similarities; overfitting first, then attribute selection errors and oversearching. Formal proofs of the effects described in this section are provided in section 5 and in several appendices.

4.1 Overfitting: Errors in Hypothesis Tests

Errors in adding components to a model, usually called *overfitting*, are probably the best known pathology of induction algorithms (Einhorn 1972; Quinlan 1987; Quinlan & Rivest 1989; Mingers 1989a; Weiss & Kulikowski 1989; White & Liu 1995; Oates & Jensen 1997). In empirical studies, induction algorithms often add spurious components to models. These components do not improve accuracy, and even reduce it, when models are tested on new data samples.[2]

Overfitting is harmful for several reasons. First, overfitted models are incorrect; they indicate that some variables are related when they are not. Some applications use induced models to support additional reasoning (e.g., Brodley & Rissland 1993), so correctness can be a central issue. Second, overfitted models require more space to store, and more computational resources to use, than models that do not contain unnecessary components. Third, the presence of irrelevant components in an overfitted model requires the collection of unnecessary data, increasing the cost and complexity of making predictions. Fourth, overfitted models are more difficult to understand. The unnecessary components complicate attempts to integrate induced models with existing knowledge derived from other sources. Overfitting avoidance has been justified solely on the grounds of producing comprehensible models (Quinlan 1987). Finally, overfitted models can have lower accuracy on new data than models that are not

[1] This problem is by no means limited to induction algorithms. Any algorithm that uses an MCP must consider n when making statistical inferences given x_{max}.

[2] The term "overfitting" is used in several ways in the literature on induction algorithms. In this paper, it refers to producing models with components that reduce population accuracy *or are unnecessary*. Other uses are more constraining, requiring that the added components always harm accuracy.

overfitted. This effect has been demonstrated with a variety of domains and systems (e.g., Quinlan 1987; Jensen 1992).

Overfitting occurs when a multiple comparison procedure is applied to model components. An algorithm generates a set of n components $\mathcal{C} = \{c_1, c_2, \ldots, c_n\}$, calculates a score x_i for each component, and selects the component c_{max} with the maximum score x_{max}. Algorithms decide whether adding c_{max} to an existing model m would improve the model's predictive accuracy.

Induction algorithms vary widely in how they generate and evaluate components, but all algorithms that decide whether to add c_{max} to a model make implicit or explicit statistical hypothesis tests.[3] One common form of the test asks: "Under the null hypothesis that a component c will not improve the predictive power of the model m, what is the probability of a score at least as large as x?" When this probability is very small, algorithms reject the null hypothesis and infer that adding c will improve the predictive power of m. This form of the test is usually *incorrectly* applied to the component c_{max} and its associated score x_{max}.

The test is incorrect because it does not adjust for n, the number of components examined. The test should ask: "Under the null hypothesis that *none* of the components in $\mathcal{C}$ will improve the predictive power of the model m, what is the probability of a maximum score at least as large as x_{max}?" Overfitting occurs because the wrong form of the test is used. The algorithm makes an incorrect inference and adds c_{max} even though it does not improve the predictive power of m.[4]

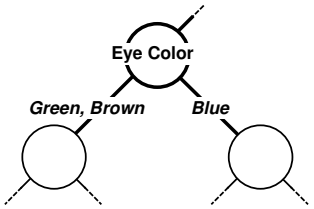
4.2 Attribute Selection Errors: Errors in Parameter Estimates

Some induction algorithms suffer from another pathology: a systematic, unwarranted preference for certain types of variables. For example, some decision tree algorithms are far more likely to construct models that use discrete variables with many values (e.g., home town) rather than discrete variables with relatively few values (e.g., gender). This behavior occurs even though models that use the latter variables have consistently higher scores when tested on new data samples. This pathology is sometimes called *attribute selection error*.[5] Attribute selection errors, particularly in tree-building systems, have been reported for more than a decade (Quinlan 1986; 1988; 1996; Mingers 1989b; Fayyad & Irani 1992; Liu & White 1994). Such errors are harmful because the resulting models have

[3] Some algorithms delay decisions about whether c_{max} will appear in the final model until a pruning phase, but they still make implicit or explicit hypothesis tests at that time.

[4] Incorrect inferences can occur even when statistical hypotheses are tested correctly. However, the probability of such errors can be made arbitrarily small.

[5] The term "attribute" in the pathology's name is derived from tree-building algorithms, where variables are sometimes called attributes.



Variable Value	Node Branch
Green	Left
Brown	Left
Blue	Right

(a)

If: (Height > 5") and
(Gender := Male) and
(Eye Color ∈ {Green, Brown})
Then: Class = +

Variable Value	Truth Value
Green	TRUE
Brown	TRUE
Blue	FALSE

(b)

Figure 1: Settings map between a variable's values and a component's output

consistently lower accuracy on new data than other models considered and rejected by an algorithm.

Attribute selection errors result from how induction algorithms construct model components. Examples of model components include nodes in decision trees, clauses in rules, nodes in connectionist networks, and terms in regression equations. In general, a component consists of a variable v and a setting t. The variable v is either drawn directly from the data sample or constructed from a combination of other variables. A setting t defines a mapping from v's values to a component's output.

In decision trees, a setting maps a variable's values to particular branches of a subtree. For example, figure 1a shows a node in a decision tree. The setting of the node $(\{Green, Brown\} \mid \{Blue\})$ maps values of the variable *eye color* to either the left or right branches of the node. Similarly, a setting in a rule maps a variable's values to a clause's truth value. Figure 1b shows a clause within a rule. The setting $(\{Green, Brown\})$ of the clause in bold maps values of *eye color* to either TRUE or FALSE.

Many algorithms select the setting of a component by using an MCP to find the best setting for each variable in a sample. For simplicity, we will examine the two-variable case, and later generalize to k variables. For two variables in a data sample $\mathcal{S}$, an algorithm generates n_1 settings $\mathcal{T} = \{t_1, t_2, \ldots, t_{n_1}\}$ for the first variable and n_2 settings $\mathcal{T} = \{t_1, t_2, \ldots, t_{n_2}\}$ for the second variable. For each variable, the algorithm then calculates a score for each setting, and selects the setting t_{max} with the maximum score x_{max}. This produces two settings t_{max_1} and t_{max_2} with scores x_{max_1} and x_{max_2}, respectively.

Ideally, we would like the two maximum scores x_{max_1} and x_{max_2} to be a good estimates of their respective population scores ψ_{*_1} and ψ_{*_2}. We denote the population score of item selected by an MCP as ψ_* rather than ψ_{max} because the

latter implies $\psi_{max} = max(\psi_1, \psi_2, \ldots, \psi_n)$, an incorrect interpretation. ψ_* is the population score of the item with the maximum sample score, not necessarily the maximum population score. If x_{max_1} and x_{max_2} are good estimates of the two population scores ψ_{*_1} and ψ_{*_2}, then we could determine which of the two variables produces the best overall component. In the terms of classical statistical inference, we wish to produce accurate estimates of two parameters — the population scores ψ_{*_1} and ψ_{*_2} of the settings selected by the two $MCPs$.

Unfortunately, the most obvious estimates, x_{max_1} and x_{max_2}, are biased and, if $n_1 \neq n_2$, they are not directly comparable. To place the scores on an equal footing, each score should be adjusted for its respective n, the number of settings. Otherwise, scores resulting from variables with large n will be incorrectly favored over scores resulting from variables with small n.[6] This effect generalizes to k variables, where in general $n_1 \neq n_2 \neq n_3 \ldots \neq n_k$.

This is directly analogous to the second part of the investment advisor example. Recall that you examined the performance of only 10 advisors while your friend examined the performance of 30 advisors. All advisors perform at a chance level, but your friend was far more likely to find a high-scoring advisor merely because he examined more advisors. Similarly, an induction algorithm is more likely to construct a high-scoring component when the number of settings n is large. Induction algorithms that directly compare $x_{max_1}, x_{max_2}, \ldots, x_{max_k}$ are making the same mistake as we would if we directly compared your top-scoring advisor with your friend's top-scorer.

4.3 Oversearching: Errors in Parameter Estimates

A third pathology was recently revealed by several studies (Murthy & Salzberg 1995; Quinlan & Cameron-Jones 1995) examining the behavior of induction algorithms that efficiently search extremely large spaces of models. Paradoxically, these algorithms produce models that are often less accurate on new data than models produced by algorithms that search only a fraction of the same space. This pathology, termed *oversearching*, is harmful because the resulting models have lower accuracy, and because constructing such models uses more computational resources.

Algorithms that suffer from oversearching examine progressively larger spaces of models. Initially, an algorithm examines a small space of models $\mathcal{M}_1 = \{m_1, m_2, \ldots, m_{n_1}\}$ and selects the model with the maximum score. Then, it expands the search to a larger space of models $\mathcal{M}_2 = \{m_1, m_2, \ldots, m_{n_1}, \ldots, m_{n_2}\}$, and selects the model with the maximum score. Expansion continues until a fixed resource bound is reached or until some predefined class of models has been searched exhaustively.

[6] Some early treatments of attribute selection error (e.g., Quinlan 1988) identify a secondary cause of the pathology — an evaluation function inherently biased toward attributes with larger numbers of possible values. This source of error has long been corrected in most induction algorithms yet the pathology remains (Quinlan 1996).

Searching progressively larger spaces of models involves several applications of a multiple comparison procedure. As in attribute selection errors, the relevant inference is which of k $MCPs$ produces the item with the best population score given the sample scores $x_{max_1}, x_{max_2}, \ldots, x_{max_k}$. Because $n_1 < n_2 \ldots < n_k$ the scores $x_{max_1}, x_{max_2}, \ldots, x_{max_k}$ are not directly comparable. Each score should be adjusted for the number of models examined by each MCP. Otherwise, scores resulting from $MCPs$ with large n will be incorrectly favored over scores resulting from $MCPs$ with small n.

5 Individual and Maximum Scores

The validity of both types of statistical inferences made by induction algorithms — hypothesis tests and parameter estimates — depend on using the correct sampling distribution. The investment advisor example sketched why the sampling distribution of X_{max} depends on n, the number of items examined by an MCP. In this section, we provide more general proofs of the effect of n on the sampling distribution of X_{max}, and how that distribution compares to the sampling distribution of an individual score X_i.

5.1 The Sampling Distribution of the Maximum

Statistical hypothesis tests use sampling distributions directly. By comparing a score x to the sampling distribution of X derived under the null hypothesis H_0, an algorithm can estimate $Pr(X \geq x|H_0)$. Alternatively, an algorithm can use the sampling distribution to derive a *critical value* x_c such that $Pr(X \geq x_c|H_0) \leq \alpha$, where α is a given probability of incorrectly rejecting the null hypothesis.

Even when induction algorithms do not explicitly test statistical hypotheses (and most do not), they do so implicitly. Nearly all algorithms require that a component's score exceed a given threshold before the algorithm will include the component in the final model. A threshold serves the same function as a critical value, and just like a critical value, the threshold should be set based on a sampling distribution. If it is not, the probabilistic interpretation of exceeding a threshold is unknown.

The sampling distribution of X_{max} (or, alternatively, the correct threshold value) depends on n, the number of items examined by an MCP. For simplicity and concreteness, assume the scores X_1 and X_2 have specific values x_1 and x_2 drawn from independent uniform distributions of integers $(0 \ldots 6)$. The distribution of X_{max} is shown in Table 1. Each entry in the table represents a joint event with the resulting maximum score; for example, $(X_1 = 3 \ \wedge \ X_2 = 4)$ has the result, $max(x_1, x_2) = 4$. Because X_1 and X_2 are independent and uniform, every joint event has the same probability, 1/49, but the probability of a given maximum score is generally higher; for example, $Pr(max(x_1, x_2) = 6) = 13/49$.

		X_1						
		0	*1*	*2*	*3*	*4*	*5*	*6*
	0	0	1	2	3	4	5	6
	1	1	1	2	3	4	5	6
	2	2	2	2	3	4	5	6
X_2	*3*	3	3	3	3	4	5	6
	4	4	4	4	4	4	5	6
	5	5	5	5	5	5	5	6
	6	6	6	6	6	6	6	6

Table 1: The joint distribution of the maximum of two scores, each of which takes integer values (0...6).

For independent and identically distributed (i.i.d.) scores $X_1, X_2, \ldots, X_n$, it is easy to specify the relationship between cumulative probabilities of individual scores and cumulative probabilities of maximum scores:

$$\text{If } Pr(X_i < x) = q, \text{ then } Pr(X_{max} < x) = q^n. \tag{1}$$

For example, in Table 1, $Pr(X_1 < 4) = 4/7$ (and $Pr(X_2 < 4)$ is identical, because X_1 and X_2 are i.i.d.), but $Pr(max(x_1, x_2) < 4) = (4/7)^2 = 16/49$. It is also useful to look at the upper tail of the distribution of the maximum:

$$\text{If } Pr(X_i \geq x) = p, \text{ then } Pr(X_{max} \geq x) = 1 - (1-p)^n. \tag{2}$$

These expressions and the distribution in Table 1 make clear that the distribution of any individual score X_i from i.i.d. scores $X_1, X_2, \ldots, X_n$ underestimates the distribution of X_{max}. $Pr(X_i \geq x)$ underestimates $Pr(X_{max} \geq x)$ for all values x if the distributions are continuous. Said differently, the distribution of X_{max} has a heavier upper tail than the distribution of X_i.

This disparity increases with n, the number of scores. Consider three scores distributed in the same way as the two in Table 1. Then,

$$\begin{aligned} Pr(X_i \geq 4) &= 3/7 = 0.43 \\ Pr(max(x_1, x_2, x_3) \geq 4) &= 1 - (1 - 3/7)^3 = 0.81. \end{aligned}$$

$Pr(X_i \geq 4)$ underestimates $Pr(X_{max} \geq 4)$ by almost half its value.

This effect can be demonstrated empirically. We draw 30,000 data samples of 250 instances from a population with a single binary classification variable and 30 binary attribute variables. All variables are independent and uniformly distributed. For each attribute, we calculate a score indicating how well it predicts the classification, using a chi-square statistic as an evaluation function.

This produces values of the scores $X_1, X_2, \ldots, X_{30}$ where each X_i is distributed as chi-square.

For each of the 30,000 samples, we find x_{max}. The maximum score is found for the first ten scores (e.g., $x_{max} = max(x_1, x_2, \ldots, x_{10})$) as well as all thirty. The distributions of these 30,000 maximum scores approximate the sampling distributions for X_{max} when $n = 10$, and $n = 30$.

Figure 2 shows how the distribution of a single score $(n = 1)$ compares to the distributions of the maximum scores for $n = 10$ and 30. For $n > 1$, the sampling distribution of X_{max} diverges from the sampling distribution of X_i $(n = 1)$. The degree of divergence increases with n. In practice, induction algorithms regularly use $MCPs$ for which $n > 100$ or even $n > 1000$. The number of items n considered by an MCP strongly affects the sampling distribution for X_{max}. Hypothesis tests will be inaccurate if they compare sample scores x_{max} to the sampling distribution for X_i rather than X_{max}.

5.2 The Maximum Score and Biased Estimators

Poor parameter estimates are responsible for the pathologies of attribute selection error and oversearching. Many induction algorithms use the sample score x_{max} to estimate ψ_*, the population score of the item with the maximum sample score. One way to examine how well x_{max} estimates ψ_* is to compare the expected value of X_{max}, $E(X_{max})$, to ψ_*. In statistical terms, an estimator X of a population parameter ψ is said to be unbiased if $E(X) = \psi$. Below, we establish that $E(X_i) < E(X_{max})$ for both discrete and continuous random variables. Then, we use this relationship to show that X_{max} is a biased estimator of ψ_*.

Proposition For discrete random variables $X_1, X_2, \ldots, X_n$ and $x_{max} = max(x_1, x_2, \ldots, x_n)$,

$$E(X_i) \leq E(X_{max})$$

Proof The expected value of the discrete random variable X is usually defined as the sum, over all possible values x, of the value x multiplied by its probability $p(x)$:

$$E(X) = \sum_x x p(x).$$

For scores, each possible value x is derived from one or more samples $\mathcal{S}$. Each sample produces only a single value x, although many samples may produce the same value x. Because of this many-to-one mapping from samples $\mathcal{S}$ to values x, the expected value of a discrete random variable can equivalently be defined over all possible samples $\mathcal{S}$

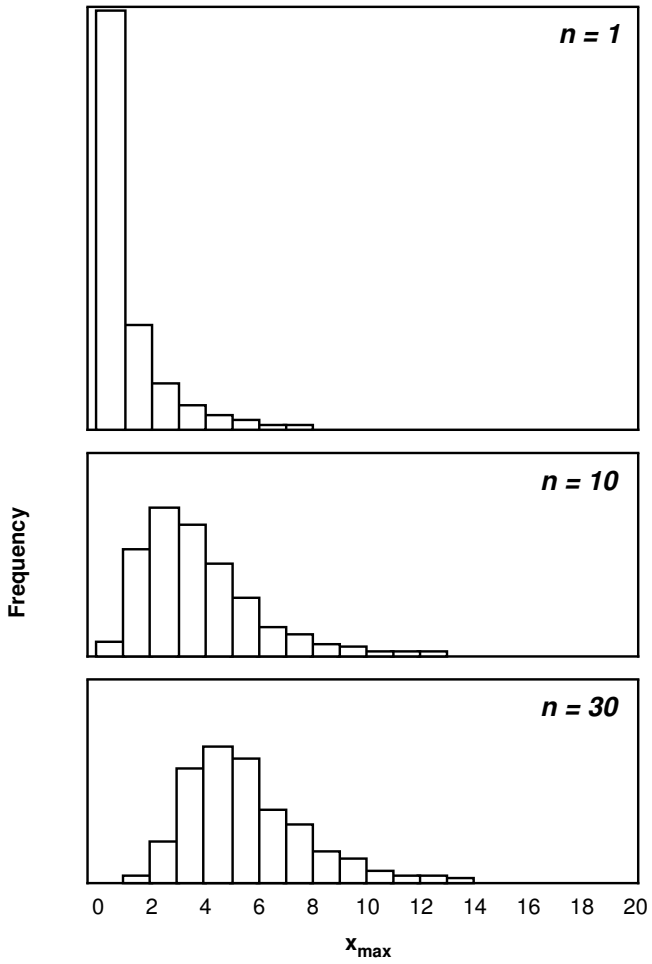


Figure 2: Distributions of X_{max} for $n = 1, 10$, and 30

$$E(X) = \sum_{\mathcal{S}} x(\mathcal{S})p(\mathcal{S}).$$

where $x(\mathcal{S})$ is the value of x for a given sample $\mathcal{S}$.

Given that the function max selects among the values $x_1, x_2, \ldots, x_n$, for any score x_i, $x_i \leq max(x_1, x_2, \ldots, x_n)$, where $1 \leq i \leq n$. More succinctly, $x_i \leq x_{max}$. For a given population, x_i and x_{max} are summed across the same samples, and those samples have identical probability distributions. Therefore,

$$E(X_i) \leq E(X_{max}).$$

If for one or more samples, $x_i < x_{max}$, then

$$E(X_i) < E(X_{max}).$$

This can also be proven for continuous random variables:

Proposition For continuous random variables $X_1, X_2, \ldots, X_n$ and $x_{max} = max(x_1, x_2, \ldots, x_n)$,

$$E(X_i) \leq E(X_{max}).$$

Proof For all non-negative values x and $x_{max} = max(x_1, x_2, \ldots x_n)$

$$Pr(X_i > x) \leq Pr(X_{max} > x).$$

Integrating both sides

$$\int_0^\infty Pr(X_1 > x)dx \leq \int_0^\infty Pr(X_{max} > x)dx. \tag{3}$$

A well-known theorem of probability states that $\int_0^\infty Pr(X > x)dx = E(X)$ (Ross 1984). So,

$$E(X_i) \leq E(X_{max}).$$

If, for one or more samples, $x_i < x_{max}$, then

$$E(X_i) < E(X_{max}).$$

As before, this effect can be demonstrated empirically. Based on the distributions shown in figure 2, we can calculate the expected value for each set of 30,000 scores. Table 2 shows how the expected value of the maximum score varies with n.

Given what we now know about the expected value of X_{max}, we can prove that X_{max} is a biased estimator of ψ_*.

n	1	10	30
$E(X_{max})$	0.983	3.728	5.501

Table 2: Expected value of chi-square

Proposition Given ψ_*, the population score of the item with the maximum sample score,

$$\psi_* \leq E(X_{max})$$

That is, X_{max} is a biased estimator of the population score ψ_*.

Proof If every X_i is an unbiased estimator of the population score ψ_i, then

$$\psi_i = E(X_i).$$

As previously proven, $E(X_i) \leq E(X_{max})$. Thus, for all ψ_i

$$\psi_i \leq E(X_{max}).$$

If, for one or more samples, $x_i < x_{max}$, then

$$\psi_i < E(X_{max}).$$

That is, X_{max} is a positively biased estimator of any ψ_i, including the population score ψ_* of the item with the maximum sample score, so

$$\psi_* < E(X_{max}).$$

In words, X_{max} is a biased estimator of ψ_*.

5.3 The Effects of n on Bias

We have shown that X_{max} is a biased estimator of ψ_*. However, the descriptions of attribute selection errors and oversearching in section 4 made an additional claim: that the degree of bias increases with n, making the scores X_{max_a} and X_{max_b} incommensurable if $n_a \neq n_b$.

Proposition

$$E(X_{max_a}) < E(X_{max_b}) \text{ for } n_a < n_b.$$

Proofs for two different cases are provided in appendix A.

To summarize this entire section, the sampling distribution of X_{max} differs from that of X_i such that for all x, $Pr(X_{max} \geq x) > Pr(X_i \geq x)$. In addition, X_{max} is a biased estimator of ψ_*, the population score of the item with the maximum sample score. The degree of bias increases with n, the number of items examined by an MCP.

6 Influences on the Maximum Score

Several factors influence the degree to which the sampling distribution of X_{max} diverges from the sampling distribution of X_i. For convenience, we define $\mathcal{E} = Pr(X_{max} \geq x) - Pr(X_i \geq x)$. Informally, $\mathcal{E}$ indicates the probability of error if one assumes the distributions of X_i and X_{max} are equal. Increasing $\mathcal{E}$ increases the probability of error. We have already shown that, if all other things are equal, $\mathcal{E}$ increases with n. In this section, we examine three other factors. $\mathcal{E}$ increases as: 1) $X_1, X_2, \ldots, X_n$ approach independence; 2) sample size $|\mathcal{S}|$ decreases; and 3) $E(X_1), E(X_2), \ldots, E(X_n)$ approach equality.

6.1 Independence

Two random variables, X and Y, are independent if knowing the value of one variable tells you nothing about the distribution of the other (Ross 1984). Discrete random variables are independent if $Pr(x, y) = Pr(x)Pr(y)$. Continuous random variables are independent if $Pr(X < x, Y < y) = Pr(X < x)Pr(Y < y)$.

In practice, *MCPs* often examine items whose scores are not independent. For example, decision tree algorithms examine multiple partitions of a continuous variable (e.g., the partitions $B < 1$, $B < 2$, $B < 3$, and $B < 4$). These partitions are certain to have dependent scores because they define related partitions. In addition, model components can have dependent scores when they use variables that are intrinsically dependent (e.g., height and weight).

We will prove that one form of dependence — positive correlation between scores — decreases $\mathcal{E}$. To understand the effect informally, consider the effect of positive correlation shown in figure 3. The figure shows three possible joint distributions of X_1 and X_2. Each point in a graph represents a joint event (x_1, x_2). The score x is marked on each variable's axis. The points in the shaded region of each figure indicate the events where $X_{max} \geq x$.

In figure 3a, X_1 and X_2 are independent. Because of the location of x, $Pr(X_i \geq x) = 0.50$. As indicated by the points in the shaded region, $Pr(X_{max} \geq x) = 0.75$, making $\mathcal{E} = 0.25$. Figure 3b shows the effect of strong positive correlation between X_1 and X_2. $Pr(X_{max} \geq x)$ is only slightly larger than 0.50, and therefore $\mathcal{E}$ is nearer to zero. In figure 3c, the positive correlation of the scores is perfect. The distribution of X_{max} is identical to the distribution of X_i, $Pr(X_{max} \geq x) = Pr(X_i \geq x)$ and thus $\mathcal{E} = 0$.

Appendix B contains a formal proof that, for continuous random variables $X_1, X_2, X_3,$ and X_4,

$$\mathcal{E}_a > \mathcal{E}_b.$$

for all values x where $\mathcal{E} = Pr(X_{max} \geq x) - Pr(X_i \geq x)$, $x_{max_a} = max(x_1, x_2)$, $x_{max_b} = max(x_3, x_4)$, $X_1, X_2,$ and X_3 are i.i.d., $X_1, X_2,$ and X_4 are i.i.d., but X_3 and X_4 are positively correlated.

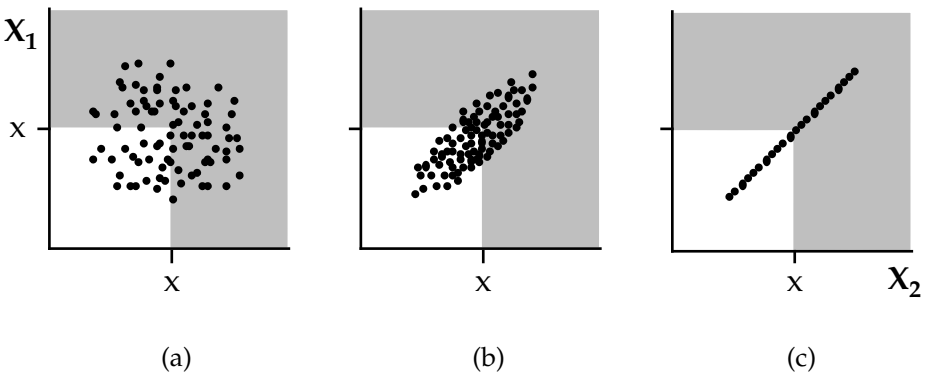


Figure 3: Positive Correlation Affects $Pr(X_{max} \geq x)$

6.2 Sample Size

The size of the sample $\mathcal{S}$ is another determinant of $\mathcal{E}$. Decreasing sample size increases the standard deviation of X_i, increasing the probability of values far from $E(X_i)$, thus increasing $Pr(X_{max} \geq x)$, and thus increasing $\mathcal{E}$. X_i is a sampling distribution of the score x_i, and thus the standard deviation of X_i is known as the *standard error* of the score x_i, denoted σ_{x_i}. As the size of $\mathcal{S}$ approaches the size of the entire population, σ_{x_i} approaches zero.

In practice, induction algorithms often calculate scores based on small samples. For example, tree-building algorithms systematically decrease sample size by repeatedly splitting the original data sample. Starting with a sample size of 1000, a tree with a branching factor of three produces leaves with fewer than 15 instances after only four levels. Lower levels of decision trees will thus have much larger $\mathcal{E}$ than higher levels.

We will show that increasing the σ_{x_i} increases $\mathcal{E}$, for all x such that $Pr(X_i \geq x) \neq 0.50$. This latter restriction on x holds true for nearly all situations of interest — we are nearly always interested in cases where $Pr(X_i \geq x)$ is very small, not where this probability is near 0.5.

Consider the graphical example in figure 4. The standard errors σ_{x_1} and σ_{x_2} are largest in figure 4a where $Pr(X_i \geq x) \approx 0.50$, $Pr(X_{max} \geq x) \approx 0.75$, and $\mathcal{E} \approx 0.25$. However, as the standard errors decrease (e.g., figure 4) these values all tend toward zero.

Appendix C gives a formal proof that:

$$\mathcal{E}_a > \mathcal{E}_b$$

where $\mathcal{E} = Pr(X_{max} \geq x) - Pr(X_i \geq x)$, $x_{max_a} = max(x_1, x_2)$, $x_{max_b} = max(x_3, x_4)$, $\sigma_{x_1} = \sigma_{x_2} > \sigma_{x_3} = \sigma_{x_4}$, $X_1 \ldots X_4$ are otherwise identically and independently distributed.

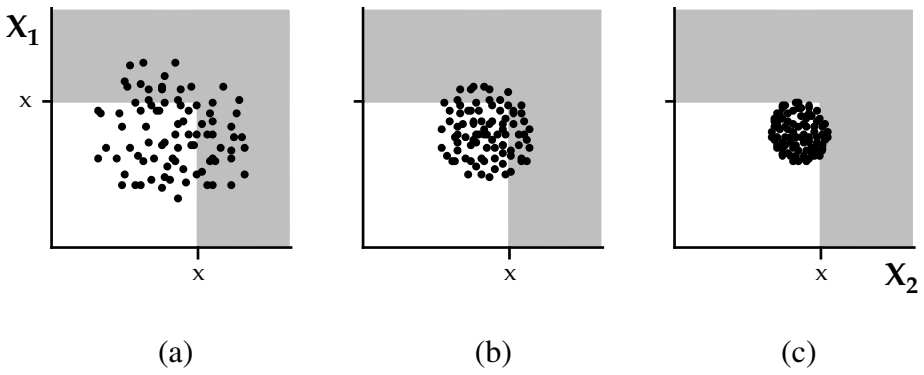


Figure 4: Standard Error Affects $Pr(X_{max} \geq x)$

6.3 Difference in Expected Value

Previous sections assumed that the expected values of individual scores $X_1, X_2, \ldots, X_n$ were equal, an assumption that is often incorrect. For example, if we were constructing model components in the domain of medical diagnosis, expected values would be equal only if all diagnostic tests and symptoms were equally useful in predicting disease. In reality, the utility of diagnostic signs varies greatly, and a similar situation prevails in most induction problems — the scores for different models, components, and settings rarely have identical expected values.

For convenience, we define $\delta = E(X_1) - E(X_2)$ as the difference between the expected values of two scores X_1 and X_2. Below, we prove that $\mathcal{E}$ varies inversely with δ. Figure 5 shows this effect graphically. In Figure 5a, $E(X_1) = E(X_2)$, $P(X_1 \geq x) = 0.50$ and $P(X_{max} \geq x) = 0.75$ (the shaded portion of the figure), making $\mathcal{E} = 0.25$. In Figure 5c, $E(X_1) \gg E(X_2)$ making $P(X_1 \geq x) \approx P(X_{max} \geq x) \approx 1.0$ and $\mathcal{E} \approx 0$.

In appendix D, we formally prove that:

$$\mathcal{E}_a > \mathcal{E}_b$$

where $\mathcal{E} = Pr(X_{max} \geq x) - Pr(X_1 \geq x)$, $x_{max_a} = max(x_1, x_2)$, $x_{max_b} = max(x_3, x_4)$, $E(X_1) = E(X_2) = E(X_3) < E(X_4)$, $X_1 \ldots X_4$ are otherwise identically and independently distributed.

7 Solutions

Several methods can compensate for the effects of $MCPs$ and allow valid statistical inferences about the score x_{max}. Four are covered below: 1) using a new data sample to derive scores for the item with the maximum sample score; 2) using cross-validation to derive scores; 3) constructing a reference distribution

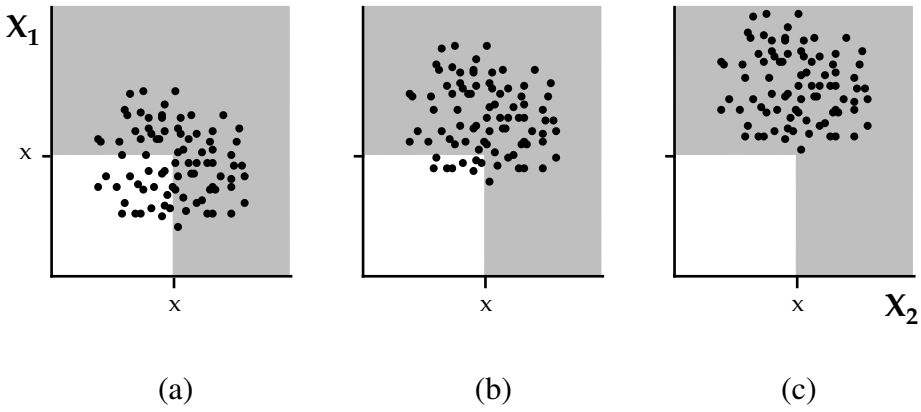


Figure 5: Expected Value Affects $\mathcal{E}$

for x_{max} by randomization; or 4) modifying the results of using a standard reference distribution by a Bonferroni adjustment. The first two methods calculate a score that can be treated as an individual score X_i rather than a maximum score X_{max}. The last two methods create a sampling distribution appropriate to X_{max}.

7.1 New Data Sample

The most obvious method to adjust for the effects of an MCP is to evaluate items on a new data sample $\mathcal{S}_{new}$ disjoint from the original sample $\mathcal{S}$. Suppose an MCP selects the component $c_3 = c_{max}$ using the data sample $\mathcal{S}$. Valid statistical inferences about c_3 that use $\mathcal{S}$ must adjust for n. However, inferences about c_3 that are based on a new data sample $\mathcal{S}_{new}$ need not consider how c_3 was selected using $\mathcal{S}$, as long as $\mathcal{S}_{new}$ shares no instances with $\mathcal{S}$. In the case of the investment advisor analogy, one could test the best candidate on 14 additional days — a new sample. If that candidate passes the eleven-or-more test based on the new sample, then the probability of incorrectly rejecting the hypothesis that he or she is a charlatan is not greater than 0.0287.

Several induction algorithms (e.g., Quinlan 1987; Jensen 1992) use new data to compensate for the effects of $MCPs$. They partition the training sample into two samples, use one sample for $MCPs$, and use the other for hypothesis tests and parameter estimates for the resulting items.

7.2 Cross-Validation

Cross-validation is a more sophisticated method for obtaining scores based on disjoint data samples (Kohavi 1995; Cohen 1995; Weiss & Kulikowski 1989).

Cross-validation divides a sample $\mathcal{S}$, with N instances, into k disjoint sets, $\mathcal{S}_i$, each of which contains N/k instances. Then, for $1 \leq i \leq k$, an MCP selects maximum-scoring items based on the sample $\mathcal{S} - \mathcal{S}_i$ and those items are evaluated on the sample $\mathcal{S}_i$. This produces k different estimates of accuracy that are combined to produce a single estimate (e.g., by averaging).

Cross-validation compensates for the effects of $MCPs$ and partially avoids the highly variable results obtained by using only a single partition of the data. However, the method is computationally-intensive (typically, $k = 10$) and its results can still be highly variable (Kohavi 1995).

7.3 Randomization

Randomization (Cohen 1995; Edgington 1995; Jensen 1992; Noreen 1989) can be used to construct an empirical sampling distribution. Each iteration of randomization creates a sample $\mathcal{S}_i^*$ that is consistent with the null hypothesis. The MCP used to obtain the actual score x_{max} is repeated on $\mathcal{S}_i^*$, producing a value $x^*_{max_i}$ from the sampling distribution of x under the null hypothesis. A large number of iterations produces an approximation to the complete sampling distribution of X_{max}.

For example, consider the problem of finding whether any of ten binary variables $A_1, A_2, \ldots, A_{10}$ is predictive of another binary variable A_0. The most predictive variable is the one most highly correlated with A_0 based on a sample $\mathcal{S}$. Call its correlation x_{max}. An hypothesis test requires the sampling distribution of x_{max} under the null hypothesis that A_0 is uncorrelated with any of the ten variables. Randomization can produce an approximate sampling distribution by generating 1000 randomized samples and finding the correlation of the most predictive variable in each. Each randomized sample reproduces the values of $A_1, A_2, \ldots, A_n$ but randomly reassigns the values of A_0 with respect to the values of the other variables, thus enforcing the null hypothesis. If x_{max} exceeds a significant fraction of the correlations from the randomized samples (e.g., 95%), we infer it is predictive of A_0.

Randomization tests have several desirable features. They produce reference distributions appropriate for X_{max} rather than only X_i. They do not require that the individual scores examined by an MCP be independent and identically distributed (requirements of another technique, Bonferroni adjustment, discussed below). Finally, randomization tests can create a reference distribution for any evaluation function f, not just those for which reference distributions have been analytically derived.

Unfortunately, randomization tests are computationally expensive, requiring evaluation of k randomized samples. Values of k are typically greater than 100, and the resolution of a randomization test depends on k. If $k < 100$, it is certainly impossible to make distinctions among probability values that differ by less than 1%, and $k \gg 100$ would be necessary before such fine distinctions could be made reliably.

7.4 Bonferroni Adjustment

Bonferroni adjustment converts probability values for a single score X_i into probability values for X_{max}. One basic form of the Bonferroni adjustment was given in equation 2. For scores X_i that are i.i.d.:

$$\text{If } Pr(X_i \geq x) = p, \text{ then } Pr(X_{max} \geq x) = 1 - (1-p)^n. \tag{4}$$

If we set x equal to an actual maximum score calculated for a particular sample, and determine p based on the sampling distribution for a single score X_i, then equation 4 can be used to determine $Pr(X_{max} \geq x)$ under the null hypothesis. Consider an algorithm that generates 50 models, evaluates each, and selects the model with the maximum score. If the evaluation function is the G statistic and the maximum value is 7.88, then $Pr(X_i \geq 7.88) = 0.005$ using a chi-square distribution with 1 degree of freedom. The algorithm can use the Bonferroni adjustment to compensate for evaluating 50 models and conclude that $Pr(X_{max} \geq 7.88) = 1 - (1 - 0.005)^{50} = 0.222$.

Bonferroni adjustment imposes almost no additional computational burden to adjust for the effects of $MCPs$, but equation 4 only holds if the scores X_i are mutually independent and identically distributed. Figure 6 illustrates how dependence among scores affects Bonferroni adjustment, randomization, and cross-validation. The experiment is similar to that which produced figure 2. We create random data samples, each with a binary classification variable and 20 attribute variables and with varying levels of dependence among the attributes (measured by median pairwise correlation). We conduct 500 trials for each level of dependence among the attributes. Each trial uses four methods to infer whether the correlation between the classification and the best attribute is significant at the 10% level — a significance test using the distribution of the single score X_i, cross-validation, randomization, and a Bonferroni-adjusted test. The y-axis indicates the percentage of trials in which a method inferred a significant relationship. Ideally, this empirical probability should be 0.10 across all values of median pairwise correlation. Using the distribution of a single score clearly fails except when the attributes exhibit complete dependence. Cross-validation and randomization both accurately adjust for the number of comparisons n over the entire range of attribute dependence. The Bonferroni adjusted estimate is correct for low values of attribute dependence, but not for high values. Only cross-validation and randomization tests are accurate across all levels of attribute dependence.

8 Previous Work

Several previous theories and empirical findings in machine learning and statistics implicate the statistical properties of multiple comparison procedures as the cause of pathologies in induction algorithms. Our work provides explicit proof

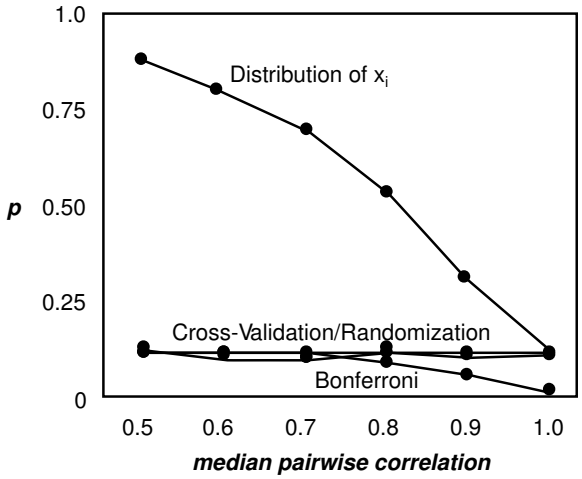


Figure 6: How different methods compensate for dependence among scores

of some prior qualitative explanations. For example, overfitting, oversearching, and attribute selection errors have often been attributed to "fluke" relationships. The statistical properties of $MCPs$ explain the frequency of those flukes and indicate effective solutions. In other cases, previous work lends support to the notion that $MCPs$ have an important influence on the credibility of induced models. For example, the Vapnik-Chervonenkis dimension and minimum description length principle point toward the number of comparisons n as an important factor in overfitting. Finally, our explanation of the mechanism behind overfitting, oversearching, and attribute selection errors is enhanced by looking at two related concepts: overfitting avoidance as bias and the bias-variance tradeoff. Each of these points is elaborated below.

8.1 Multiple Comparisons

A large statistical literature examines the effects of multiple comparisons (Miller 1981). Much of this literature is concerned with experimental design, rather than the design of induction algorithms. Some work in machine learning (Gascuel & Caraux 1992; Feelders & Verkooijen 1995; Salzberg 1997) also pursues this former course, correctly noting the effect of multiple comparisons on empirical evaluation of learning algorithms.

Only a few induction algorithms explicitly compensate for multiple comparisons. CHAID (Kass 1980) and TBA (Jensen & Schmill 1997) use Bonferroni adjustment to compensate for multiple comparisons during tree construction. INDUCE (Gaines 1989) uses a Bonferroni adjustment to compensate for comparing multiple rules. IRT (Jensen 1991; 1992) uses randomization tests to

compensate for comparing multiple classification rules.

The effects of multiple comparisons has led some researchers to reject statistical hypothesis tests entirely. For example, some early tree-building algorithms such as AID completely dispense with significance tests. According to the program's authors (Morgan & Andrews 1973; Sonquist, Baker, & Morgan 1971), AID's multiple comparisons render statistical significance tests useless. Similarly, Quinlan (Quinlan 1987) rejects conventional significance tests on empirical grounds in favor of error-based pruning, the current approach used in C4.5.

Despite this infrequent use of statistical tests and the lack of attention to multiple comparisons, the qualitative explanations for pathologies of induction algorithms often have statistical overtones. Explanations of overfitting (e.g., Mingers 1989a) frequently cite the problem of fitting models to "noise" or random variation. As noted above, explanations of oversearching (Murthy & Salzberg 1995; Quinlan & Cameron-Jones 1995) often cite "fluke" models that are more likely to be discovered with extensive search. Many explanations of attribute selection errors reference the increased likelihood of finding spuriously high scores when components use variables with many possible discrete values (e.g., Mingers 1989b). Few of these explanations are more than qualitative, and even fewer include theoretical proofs.

8.2 Model Complexity and Credibility

Some of the work that attempts to provide a theoretical basis for avoiding pathologies, particular overfitting, focuses on tradeoffs between the complexity and the accuracy of a model. For example, some algorithms explicitly consider both complexity and accuracy when evaluating model components (Iba, Wogulis, & Langley 1988). Cost-complexity pruning, a technique employed in the CART algorithm (Breiman *et al.* 1984), attempts to find a near-optimal complexity for a given tree through cross-validation.

Several more formal treatments consider model complexity as a way to avoid overfitting. One such treatment, the Minimum Description Length (MDL) principle, formally balances accuracy and complexity (Quinlan & Rivest 1989). MDL characterizes datasets and models by the number of bits required to encode them. The total information in a dataset $\mathcal{S}$ is described as the sum of the information necessary to encode a model and to encode any exceptions to the model remaining in $\mathcal{S}$. The best model results in the smallest total "description length" for the data, that is, the smallest sum of model description and description of the remaining data. MDL has been applied to avoid overfitting (Quinlan & Rivest 1989) and attribute selection errors (Quinlan 1996) in decision trees.

The Vapnik-Chervonenkis (VC) dimension also links complexity and overfitting. It characterizes a relationship between an hypothesis space H and an instance space X (Blumer *et al.* 1989). If at least one member of H can distinguish between any possible dichotomy of X, then X is said to be "shattered" by

H. The VC dimension of H is equal to the largest number of instances in X that can be shattered by H. Thus, if an induction algorithm can select any member of H as its final model, and the training sample $\mathcal{S}$ is smaller than the VC dimension, then it is possible to achieve perfect classification even if there is no relationship between the (binary) classification variable and the other variables. In theory, at least, the VC dimension compensates for multiple comparisons, but provides little guidance about how to construct realistic learning algorithms.

Despite this substantial body of research on complexity, there exists little theory for why complexity and overfitting should be related. A notable exception is Pearl's 1978 paper "On the complexity and credibility of inferred models" (Pearl 1978). Pearl explains why complexity should be related to accuracy — the complexity of the final model is often related to the number of intermediate models (or components) that have been compared during its construction. Comparing many models, in turn, makes overfitting more likely. Pearl's analysis shows persuasively that complexity is merely a surrogate for multiple comparisons.

Like Pearl, it is probable that some researchers understand that complexity is a mere surrogate for multiple comparisons, but it is easy to confuse the two. Complexity is often a poor indicator of the number of comparisons. First, algorithms can search different proportions of the space of possible components. Some algorithms might search exhaustively, while others employ strong *a priori* search biases. Both could construct models of the same complexity, but with vast differences in the number of comparisons. Work in oversearching demonstrates precisely this effect. In many cases, extensive search produces models that are less accurate and equally complex as models produced by less extensive search. Second, the relationship between complexity and number of comparisons depends on the number of variables in the dataset $\mathcal{S}$. If $\mathcal{S}$ contains many variables, an algorithm might evaluate thousands of components in order to construct a relatively simple final model. If $\mathcal{S}$ contains only a few variables, the same algorithm would have to evaluate far fewer components to construct a final model of the same complexity. The final models constructed in the two cases would be of the same complexity, but would have resulted from radically different numbers of comparisons.

Intriguingly, while the VC dimension and MDL are usually cast as defining model complexity, both are more closely related to the number of comparisons made by an induction algorithm. Thus, Pearl's insights, the VC dimension, and the MDL principle all point toward multiple comparisons as an important factor in overfitting.

8.3 Overfitting Avoidance as Bias

Schaffer (Schaffer 1993) characterizes overfitting avoidance as a learning bias — that is, a method of preferring one model over another whose appropriateness is domain specific. This view has been extended to more extreme forms, referred

to as a "law of generalization performance" or a "no free lunch (NFL) theorem" (Schaffer 1994; Wolpert 1992; 1994). This work holds that any gain in accuracy obtained by avoiding overfitting (or by any other bias) in one domain will necessarily be offset by reduced accuracy in other domains. Thus, over the course of many induction problems, no overfitting avoidance technique will produce a net gain in accuracy. These theories are still highly controversial, and they rest on two unrealistic assumptions: 1) that estimates of true accuracy should exclude all instances in the sample $\mathcal{S}$; and 2) that all possible assignments of class labels are equally likely, effectively making generalization impossible (Rao, Gordon, & Spears 1995).

Regardless of the larger claims about generalization accuracy, the work on overfitting avoidance as bias (particularly Schaffer 1993) reveals an important fact: avoiding overfitting will not universally improve accuracy. Attempts to avoid overfitting will decrease accuracy on new data in some situations. However, the work of Schaffer and others does little to identify the conditions that lead to such situations. In contrast, understanding the statistical properties of $MCPs$ identifies when overfitting, attribute selection errors, and oversearching will be most serious, complementing the work of Schaffer and others.

8.4 Bias-Variance Analysis

Several recent analyses of induction algorithms (Geman, Bienenstock, & Doursat 1992; Kohavi & Wolpert 1996) have used a characterization of prediction errors that appeared originally in the statistics literature. In the context of linear regression, total error is defined as the sum of intrinsic measurement error and errors due to two other factors: bias and variance. *Bias errors* stem from systematic errors made by the model. In regression, these typically arise from incorrectly specified models — models that contain incorrect components or that contain those components in an incorrect functional form. *Variance errors* stem from random errors made by the model. In regression, these typically arise from errors in parameter estimation — incorrect estimates of the coefficients for variables in the regression equation.

$MCPs$ can produce both bias and variance errors. Bias errors can increase because of attribute selection errors and oversearching. For example, if some components of a decision tree are systematically favored (e.g., because the attributes used by the node has a very large number of discrete values), then suboptimal components will be added to the model. Models with suboptimal components are more likely to be incorrectly specified, thus introducing bias errors. Variance errors can also increase because of overfitting. For example, decision trees that are overly complex can reduce the number of instances available at a leaf to estimate the correct label. This will increase the variance of parameter estimates, thus introducing variance errors. Bias-variance analysis complements our analysis of $MCPs$, by characterizing the errors introduced by attribute selection errors, overfitting, and oversearching.

9 Implications

The statistical properties of multiple comparison procedures depend strongly on n, the number of items compared. These statistical properties affect the inferences of every induction algorithm that generates and tests models or model components. Unless they adjust for n, algorithms will add useless components to models, and they will systematically prefer suboptimal models and model components.

While the effects of multiple comparisons on statistical experiments are well known, their effects on induction algorithms have not been well explored. We have tried to address this gap through theoretical proofs and empirical demonstrations that relate multiple comparisons to common procedures in inductive learning. We have also surveyed three approaches to adjusting for multiple comparisons: Bonferroni adjustment, cross-validation, and randomization testing.

In addition to the practical implications, however, the properties of multiple comparisons provide a single causal explanation for three phenomena that have been widely observed in induction algorithms: overfitting, attribute selection errors, and oversearching. Prior research documents situations where these pathologies occur, we provide a quantitative and causal explanation of why they occur.

10 Acknowledgments

The authors wish to thank Tim Oates, Paul Utgoff, and Gunnar Blix for comments on an early draft of this paper. This research is supported by DARPA/Rome Laboratory under contract No. #F30602-93-C-0100. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements either expressed or implied, of the Defense Advanced Research Projects Agency, Rome Laboratory or the U.S. Government.

A The Effects of n on Bias

Proposition

$$E(X_{max_a}) < E(X_{max_b}) \text{ for } n_a < n_b.$$

Proof

Case 1: max_a considers a subset of the items considered by max_b. In the simplest case,

$$\begin{aligned} x_{max_a} &= max(x_1, x_2, \ldots, x_n) \\ x_{max_b} &= max(x_1, x_2, \ldots, x_n, x_{n+1}). \end{aligned}$$

For all scores x_{n+1},

$$x_{max_a} \leq x_{max_b}.$$

Because x_{max_a} and x_{max_b} are summed over the same samples,

$$E(X_{max_a}) \leq E(x_{max_b}). \tag{5}$$

If, for one or more samples, $x_{max_a} < x_{n+1}$, then

$$E(X_{max_a}) < E(x_{max_b})$$

Case 2: max_a and max_b consider disjoint sets of items.
Consider two disjoint sets of n random variables, such that

$$\begin{aligned} x_{max_a} &= max(x_1, x_2, \ldots, x_n) \\ x_{max_b} &= max(x_{n+1}, x_{n+2}, \ldots, x_{2n}, x_{2n+1}) \end{aligned}$$

and a third set such that

$$x_{max_c} = max(x_{n+1}, x_{n+2}, \ldots, x_{2n})$$

If all variables are i.i.d., they have the same domains and probability distributions. Therefore,

$$E(X_{max_a}) = E(X_{max_c})$$

We know from equation 5 that

$$E(X_{max_a}) \leq E(X_{max_b})$$

If, for some sample, $x_{max_c} < x_{2n+1}$, then

$$E(X_{max_a}) < E(X_{max_b})$$

B Influence of Independence on the Maximum Score

Proposition For continuous random variables $X_1, X_2, X_3,$ and X_4,

$$\mathcal{E}_a > \mathcal{E}_b.$$

for all values x where $\mathcal{E} = Pr(X_{max} \geq x) - Pr(X_i \geq x)$, $x_{max_a} = max(x_1, x_2)$, $x_{max_b} = max(x_3, x_4)$, $X_1, X_2,$ and X_3 are i.i.d., $X_1, X_2,$ and X_4 are i.i.d., but X_3 and X_4 are positively correlated.

Proof Given that X_3 and X_4 are positively correlated,

$$Pr(X_3 < x) < Pr(X_3 < x | X_4 < x).$$

X_1 and X_3 are identically distributed, so $Pr(X_1 < x) = Pr(X_3 < x)$ and

$$Pr(X_1 < x) < Pr(X_3 < x | X_4 < x).$$

X_1 and X_2 are independent, so $Pr(X_1 < x) = Pr(X_1 < x | X_2 < x)$ and

$$Pr(X_1 < x | X_2 < x) < Pr(X_3 < x | X_4 < x).$$

X_2 and X_4 are identically distributed, so $Pr(X_2 < x) = Pr(X_4 < x)$ and

$$Pr(X_1 < x | X_2 < x) Pr(X_2 < x) < Pr(X_3 < x | X_4 < x) Pr(X_4 < x).$$

By simple axioms of probability and inequality,

$$\begin{aligned} Pr(X_1 < x, X_2 < x) &< Pr(X_3 < x, X_4 < x) \\ -Pr(X_1 < x, X_2 < x) &> -Pr(X_3 < x, X_4 < x) \\ 1 - Pr(X_1 < x, X_2 < x) &> 1 - Pr(X_3 < x, X_4 < x) \\ Pr(X_{max_a} \geq x) &> Pr(X_{max_b} \geq x). \end{aligned}$$

X_1, X_2 are i.i.d. with X_3, X_4 thus,

$$\begin{aligned} Pr(X_{max_a} \geq x) - Pr(X_{i_a} \geq x) &> Pr(X_{max_b} \geq x) - Pr(X_{i_b} \geq x) \\ \mathcal{E}_a &> \mathcal{E}_b. \end{aligned}$$

C Influence of Standard Error on the Maximum Score

Proposition

$$\mathcal{E}_a > \mathcal{E}_b$$

where $\mathcal{E} = Pr(X_{max} \geq x) - Pr(X_i \geq x)$, $x_{max_a} = max(x_1, x_2)$, $x_{max_b} = max(x_3, x_4)$, $\sigma_{x_1} = \sigma_{x_2} > \sigma_{x_3} = \sigma_{x_4}$, $X_1 \ldots X_4$ are otherwise identically and independently distributed (see figure 7).

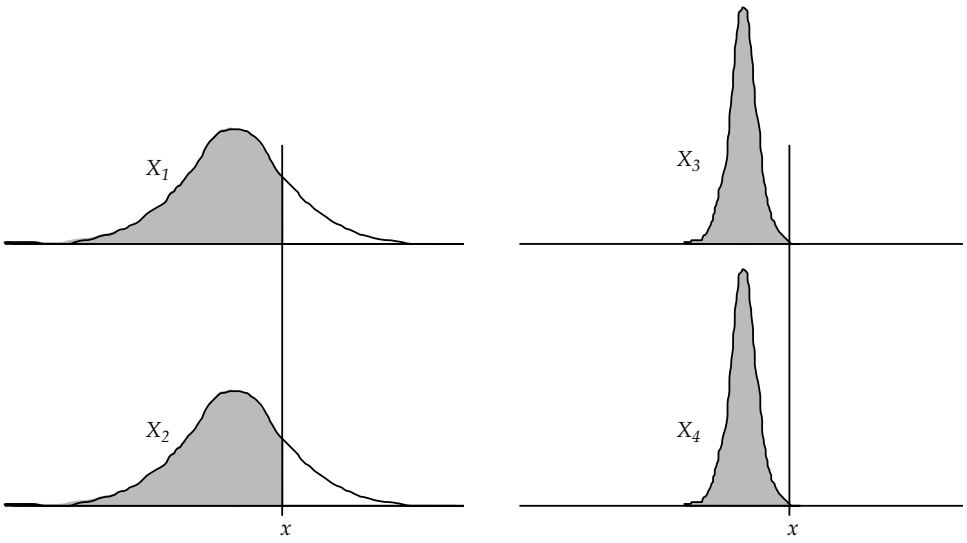


Figure 7: Distributions $X_1 \ldots X_4$

Proof For all x such that $Pr(X_i < x) > 0.5$ and $\sigma_{x_1} > \sigma_{x_3}$, we know that $0.5 < Pr(X_1 < x) < Pr(X_3 < x) < 1.0$. Under these conditions, as proven in appendix E,

$$Pr(X_1 < x)(1 - Pr(X_1 < x)) > Pr(X_3 < x)(1 - Pr(X_3 < x))$$

X_1, X_2 are i.i.d. and X_3, X_4 are i.i.d., so:

$$\begin{aligned} Pr(X_1 < x)(1 - Pr(X_2 < x)) &> Pr(X_3 < x)(1 - Pr(X_4 < x)) \\ Pr(X_1 < x) - Pr(X_1 < x)Pr(X_2 < x) &> Pr(X_3 < x) - Pr(X_3 < x)Pr(X_4 < x) \end{aligned}$$

Adding one to both sides and converting probabilities,

$$\begin{aligned} Pr(X_1 < x) + 1 - Pr(X_1 < x)Pr(X_2 < x) &> Pr(X_3 < x) + 1 - Pr(X_3 < x)Pr(X_4 < x) \\ Pr(X_1 < x) + Pr(X_{max_a} \geq x) &> Pr(X_3 < x) + Pr(X_{max_b} \geq x). \end{aligned}$$

Adding negative one to both sides and converting probabilities:

$$\begin{aligned} -1 + Pr(X_1 < x) + Pr(X_{max_a} \geq x) &> -1 + Pr(X_3 < x) + Pr(X_{max_b} \geq x) \\ Pr(X_{max_a} \geq x) - (1 - Pr(X_1 < x)) &> Pr(X_{max_b} \geq x) - (1 - Pr(X_3 < x)) \\ Pr(X_{max_a} \geq x) - Pr(X_1 \geq x) &> Pr(X_{max_b} \geq x) - Pr(X_3 \geq x) \end{aligned}$$

X_1,X_2 are i.i.d. and X_3,X_4 are i.i.d., so:

$$\begin{aligned} Pr(X_{max_a} \geq x) - Pr(X_{i_a} \geq x) &> Pr(X_{max_b} \geq x) - Pr(X_{i_b} \geq x)) \\ \mathcal{E}_a &> \mathcal{E}_b \end{aligned}$$

Similarly, for all x such that $Pr(X_i < x) < 0.5$, we know that $0 < Pr(X_1 < x) < Pr(X_3 < x) < 0.5$. Under these conditions, as proven in appendix E,

$$Pr(X_1 < x)(1 - Pr(X_1 < x)) > Pr(X_3 < x)(1 - Pr(X_3 < x))$$

and we can prove $\mathcal{E}_a > \mathcal{E}_b$ as above. In only one special case — $Pr(X_i < x) = 0.5$ — is $\mathcal{E}_a = \mathcal{E}_b$.

D Influence of Difference in Expected Value on the Maximum Score

Proposition

$$\mathcal{E}_a > \mathcal{E}_b$$

where $\mathcal{E} = Pr(X_{max} \geq x) - Pr(X_1 \geq x)$, $x_{max_a} = max(x_1, x_2)$, $x_{max_b} = max(x_3, x_4)$, $E(X_1) = E(X_2) = E(X_3) < E(X_4)$, $X_1 \ldots X_4$ are otherwise identically and independently distributed.

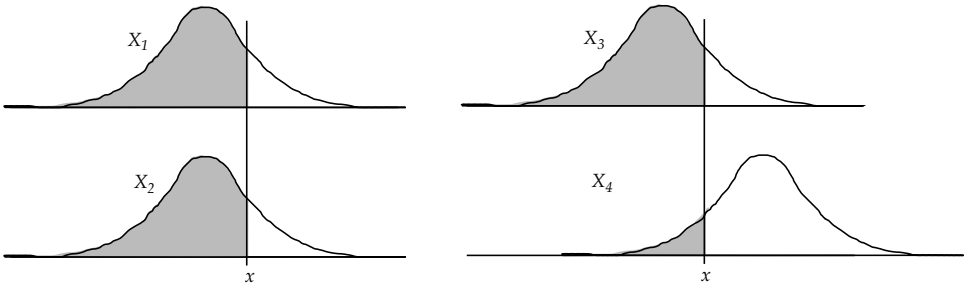


Figure 8: Distributions $X_1 \ldots X_4$

Proof Given $E(X_2) < E(X_4)$ and X_2, X_4 otherwise i.i.d., for all x

$$Pr(X_2 < x) > Pr(X_4 < x).$$

X_1 and X_3 are i.i.d., so

$$\begin{aligned}
Pr(X_2 < x)Pr(X_1 \geq x) &> Pr(X_4 < x)Pr(X_3 \geq x) \\
Pr(X_2 < x)(1 - Pr(X_1 < x)) &> Pr(X_4 < x)(1 - Pr(X_3 < x)) \\
Pr(X_2 < x) - Pr(X_1 < x)Pr(X_2 < x) &> Pr(X_4 < x) - Pr(X_3 < x)Pr(X_4 < x) \\
Pr(X_2 < x) - Pr(X_1 < x, X_2 < x) &> Pr(X_4 < x) - Pr(X_3 < x, X_4 < x)
\end{aligned}$$

Adding one to both sides and converting probabilities:

$$\begin{aligned}
Pr(X_2 < x) + 1 - Pr(X_1 < x, X_2 < x) &> Pr(X_4 < x) + 1 - Pr(X_3 < x, X_4 < x) \\
Pr(X_2 < x) + P(X_{max_a} \geq x) &> Pr(X_4 < x) + Pr(X_{max_b} \geq x).
\end{aligned}$$

Subtracting one from both sides and converting probabilities:

$$\begin{aligned} -1 + Pr(X_2 < x) + P(X_{max_a} \geq x) &> -1 + Pr(X_4 < x) + Pr(X_{max_b} \geq x) \\ P(X_{max} \geq x) - Pr(X_2 \geq x) &> Pr(X_{max} \geq x) - Pr(X_4 \geq x). \end{aligned}$$

X_4 has the maximum expected value, so we should measure $\mathcal{E}$ with respect to it, rather than with respect to X_3. X_1,X_2 are i.i.d., so

$$\begin{aligned} Pr(X_{max_a} \geq x) - Pr(X_{i_a} \geq x) &> Pr(X_{max_b} \geq x) - Pr(X_4 \geq x)) \\ \mathcal{E}_a &> \mathcal{E}_b \end{aligned}$$

E Probability Relations Used in Prior Proofs

Proposition If x and y are probabilities and $0.5 < x < y < 1$, then

$$x - x^2 > y - y^2$$

Proof Given $0.5 < x < y < 1$, then

$$x > 1 - y$$

Since $y - x > 0$

$$x(y - x) > (1 - y)(y - x)$$

Adding $x(1 - y)$ to both sides

$$\begin{aligned} x(1-y) + x(y-x) &> x(1-y) + (1-y)(y-x) \\ x - xy + xy - x^2 &> x - xy + y - x - y^2 + xy \\ x - x^2 &> y - y^2 \end{aligned}$$

The same proposition can be proven for values of x and y greater than 0.5.

Proposition If x and y are probabilities and $0 < y < x < 0.5$, then

$$x - x^2 > y - y^2$$

Proof Given $0 < y < x < 0.5$, then

$$1 - x > y$$

Since $x - y > 0$

$$(1 - x)(x - y) > y(x - y)$$

Adding $y(1 - x)$ to both sides

$$\begin{aligned} y(1-x) + (1-x)(x-y) &> y(1-x) + y(x-y) \\ y - xy + x - y - x^2 + xy &> y - xy + xy - y^2 \\ x - x^2 &> y - y^2 \end{aligned}$$

References

Blumer, A.; Ehrenfeucht, A.; Haussler, D.; and Warmuth, M. 1989. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the Association for Computing Machinery* 34:929–965.

Breiman, L.; Friedman, J.; Olshen, R.; and Stone, C. 1984. *Classification and Regression Trees*. Wadsworth International.

Brodley, C., and Rissland, E. 1993. Measuring concept change. In *Working Notes of the AAAI 1993 Spring Symposium on Training Issues in Incremental Learning*, 98–107.

Cohen, P. R. 1995. *Empirical Methods for Artificial Intelligence*. Cambridge: The MIT Press.

Edgington, E. S. 1995. *Randomization Tests*. Marcel Dekker.

Einhorn, H. 1972. Alchemy in the behavioral sciences. *Public Opinion Quarterly* 36:367–378.

Fayyad, U., and Irani, K. 1992. The attribute selection problem in decision tree generation. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, 104–110.

Feelders, A., and Verkooijen, W. 1995. Which method learns the most from data? Methodological issues in the analysis of comparative studies. In *Preliminary Papers of the Fifth International Workhop on Artificial Intelligence and Statistics*.

Gaines, B. 1989. An ounce of knowledge is worth a ton of data: Quantitative studies of the trade-off between expertise and data based on statistically well-founded empirical induction. In *Proceedings of the Sixth International Workshop on Machine Learning*, 156–159.

Gascuel, O., and Caraux, G. 1992. Statistical significance in inductive learning. In Neumann, B., ed., *ECAI92: Proceedings of the 10th European Conference on Artificial Intelligence*, 435–439. Chichester: Wiley.

Geman, S.; Bienenstock, E.; and Doursat, R. 1992. Neural networks and the bias/variance dilemma. *Neural Computation* 4:1–58.

Iba, W.; Wogulis, J.; and Langley, P. 1988. Trading off simplicity and coverage in incremental concept learning. In *Proceedings of the Fifth International Conference on Machine Learning*, 73–79.

Jensen, D., and Schmill, M. 1997. Adjusting for multiple comparisons in decision tree pruning. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*.

Jensen, D. 1991. Knowledge discovery through induction with randomization testing. In *Proceedings of the 1991 Knowledge Discovery in Databases Workshop*, 148–159.

Jensen, D. 1992. *Induction with Randomization Testing: Decision-Oriented Analysis of Large Data Sets*. Ph.D. Dissertation, Washington University.

Kass, G. 1980. An exploratory technique for investigating large quantities of categorical data. *Applied Statistics* 29(2):199–127.

Kohavi, R., and Wolpert, D. H. 1996. Bias plus variance decomposition for zero-one loss functions. In *Proceedings of the Thirteenth International Conference on Machine Learning (ICML96)*.

Kohavi, R. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the Fourteenth Joint Conference on Artificial Intelligence*.

Liu, W., and White, A. 1994. The importance of attribute selection measures in decision tree induction. *Machine Learning* 15:25–41.

Miller, R. G. 1981. *Simultaneous Statistical Inferences*. New York: Springer-Verlag.

Mingers, J. 1989a. An empirical comparison of pruning methods for decision tree induction. *Machine Learning* 4:227–243.

Mingers, J. 1989b. An empirical comparison of selection measures for decision-tree induction. *Machine Learning* 3:319–342.

Morgan, J., and Andrews, F. 1973. A comment on Einhorn's "Alchemy in the behavior sciences.". *Public Opinion Quarterly* 37:127–129.

Murthy, S., and Salzberg, S. 1995. Lookahead and pathology in decision tree induction. In *Proceedings Fourteenth International Joint Conference on Artificial Intelligence*, 1025–1031.

Noreen, E. W. 1989. *Computer-Intensive Methods for Testing Hypotheses: An Introduction*. New York, NY: John Wiley & Sons, Inc.

Oates, T., and Jensen, D. 1997. The effects of training set size on decision tree complexity. In *Proceedings of the Fourteenth International Conference on Machine Learning*, 254–262.

Pearl, J. 1978. On the connection between the complexity and credibility of inferred models. *International Journal of General Systems* 4:255–264.

Quinlan, J., and Cameron-Jones, R. 1995. Oversearching and layered search in empirical learning. In *Proceedings Fourteenth International Joint Conference on Artificial Intelligence*, 1019–1024.

Quinlan, J. R., and Rivest, R. 1989. Inferring decision trees using the minimum description length principle. *Information and Computation* 80:227–248.

Quinlan, J. 1986. Induction of decision trees. *Machine Learning* 1(1):81–106.

Quinlan, J. R. 1987. Simplifying decision trees. *International Journal of Man-Machine Studies* 27:221–234.

Quinlan, J. 1988. Decision trees and multi-valued attributes. *Machine Intelligence* 11:305–318.

Quinlan, J. 1996. Improved use of continuous attributes in C4.5. *Journal of Artificial Intelligence Research* 4:77–90.

Rao, R. B.; Gordon, D.; and Spears, W. 1995. For every generalization action, is there really an equal and opposite reaction? Analysis of the conservation law for generalization performance. In *Proceedings of the 12th International Conference on Machine Learning*, 471–479.

Ross, S. 1984. *A First Course in Probability*. Macmillan.

Salzberg, S. 1997. On comparing classifiers: Pitfalls to avoid and a recommended approach. *Data Mining and Knowledge Discovery* 1(3):(to appear).

Schaffer, C. 1993. Overfitting avoidance as bias. *Machine Learning* 10:153–178.

Schaffer, C. 1994. A conservation law for generalization performance. In *Proceedings of the Eleventh International Conference on Machine Learning*, 259–265.

Sonquist, J.; Baker, E.; and Morgan, J. 1971. *Searching for Structure (Alias, AID-III): An Approach to Analysis of Substantial Bodies of Micro-Data and Documentation for a Computer Program (Successor to the Automatic Interaction Detector Program)*. Survey Research Center, Institute for Social Research, The University of Michigan, Ann Arbor, Michigan.

Weiss, S. M., and Kulikowski, C. A. 1989. *Computer Systems That Learn*. San Mateo, CA: Morgan Kaufmann Publishers, Inc.

White, A., and Liu, W. Z. 1995. Superstitious learning and induction. *Artificial Intelligence Review* 9:3–18.

Wolpert, D. H. 1992. On the connection between in-sample testing and generalization error. *Complex Systems* 6:47–94.

Wolpert, D. 1994. Off-training set error and a priori distinctions between learning algorithms. Technical report, Santa Fe Institute, Santa Fe, NM.