Overfitting Explained

Paul R. Cohen, David Jensen
Department of Computer Science,
University of Massachusetts
Amherst, MA 01003

Abstract

Overfitting arises from bad statistical decisions. Whether or not they include explicit statistical tests, all induction algorithms make statistical decisions and thus are prone to overfitting. Most modeling algorithms calculate a score for each of several components, then test whether the best score is good enough to warrant adding the corresponding component to the model. We show that if the scores are independent, the reference distribution for any one score underestimates the reference distribution for the maximum score; thus scores will appear significant when they are not and model components will be added when they should not be. We demonstrate that the incidence of overfitting decreases as the dependence among scores increases, although this effect is generally quite small and cannot be exploited to prevent overfitting. We demonstrate an equivalence between our explanation of overfitting and the well-known statistical theory of multiple comparisons or simultaneous inference, which suggests a simple, effective method to control overfitting. Finally, we argue that overfitting is a fundamentally statistical pathology and the only way to get precise control of overfitting is to introduce appropriate statistical tests into modeling algorithms.

Keywords: Inductive learning, prediction, overfitting, statistical hypothesis testing, multiple comparisons.

`cohen@cs.umass.edu`
413-545-3613

Overfitting Explained

Abstract Overfitting arises from bad statistical decisions. Whether or not they include explicit statistical tests, all induction algorithms make statistical decisions and thus are prone to overfitting. Most modeling algorithms calculate a score for each of several components, then test whether the best score is good enough to warrant adding the corresponding component to the model. We show that if the scores are independent, the reference distribution for any one score underestimates the reference distribution for the maximum score; thus scores will appear significant when they are not and model components will be added when they should not be. We demonstrate that the incidence of overfitting decreases as the dependence among scores increases, although this effect is generally quite small and cannot be exploited to prevent overfitting. We demonstrate an equivalence between our explanation of overfitting and the well-known statistical theory of multiple comparisons or simultaneous inference, which suggests a simple, effective method to control overfitting. Finally, we argue that overfitting is a fundamentally statistical pathology and the only way to get precise control of overfitting is to introduce appropriate statistical tests into modeling algorithms.

1 An Analogy and an Informal Explanation of Overfitting

Overfitting is a statistical mistake, a result of bad statistical decisions. One purpose of this paper is to show that machine learning algorithms in general make numerous statistical decisions—even algorithms that contain no explicit statistical tests. Implicitly or explicitly, statistical hypotheses are being tested incorrectly, with overfitting as the consequence. Let's begin with an analogy and leave the statistical theory until the next section:

Suppose you are looking for an investment advisor. This person's job will be to predict whether the stock market will close up or down on any given day. In response to an advertisement, $n = 10$ candidates are identified, and to each you pose a challenge: "Predict whether the market will close up or down on each of the next 14 day. The person who predicts correctly most often will be my adviser, provided he or she predicts correctly 11 times or more." Why 11? Well, if a respondent is a charlatan—which for our purposes is a person who performs no better than chance—then he or she has a 50% chance of guessing correctly on each day, and the probability that this person will guess correctly on 11 or more days is only .0287. You reason that a charlatan probably won't pass the eleven-or-more test, or conversely, that anyone who passes the test probably isn't a charlatan. Further, you reason that the probability of ending up with a charlatan as an advisor is no greater than .0287. But here your logic fails you: It is true that a person who predicts correctly on eleven or more days has a probability of .0287 or less of being a charlatan, but the probability of hiring a charlatan depends on the number of applicants, n, and is .0287 only if $n = 1$. In general, the probability of hiring a charlatan is $1 - (1 - .0287)^n$, which for $n = 10$ is .253. By not accounting for the number of applicants, 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. This doesn't mean the individual in question is performing better than chance.

Hiring a charlatan on the basis of a sample of responses is analogous to overfitting: You accept a predictor because it performs well on a sample (e.g., 14 days of predictions) but its apparently good performance is the result of chance variation. If you evaluated only one candidate by some criterion (e.g., the eleven-or-more rule), then your probability of overfitting is p (e.g., $p = .0287$), but if you evaluate the best of n candidates by the same rule, your probability of overfitting is $1 - (1 - p)^n$.

Although the analogy doesn't extend perfectly, it is worth considering the case in which you hire a second investment advisor A_2 to corroborate the first, A_1. Whenever A_1 predicts that the market will close up, A_2 confirms or refutes the prediction. Notice that the eleven-or-more rule won't work as a criterion for selecting A_2 because A_2 deals only with "closing up" predictions, and we'd expect A_1 to produce only 7 of these in a 14-day sample. Even so, the probability that a charlatan could make 5 or more correct predictions in seven opportunities is only .0625, so you select the best of the remaining $n-1$ advisor candidates, and if he passes the five-or-more threshold, you hire him. As before, the probability that one charlatan passes the test underestimates the probability that the best of $n - 1 = 9$ candidates does; in fact, the probability of overfitting in this case is $1 - (1 - .0625)^9 = .44$.

This is an analogy to decision tree induction where A_1 is the first split in the tree and A_2 is a split at one of A_1's children. The analogy tells us that *at each split*, the probability of overfitting can be seriously underestimated unless the number of candidate attributes is accounted for. It also tells us that the criterion for accepting a split must account for the number of training instances that are channeled through each potential split. Failure to account for these factors results in decision tree induction algorithms that overfit dramatically.

Having outlined this argument informally, we now consider its foundation in statistical hypothesis testing, and there we will find several solutions to the overfitting problem [4].

2 Iterative Modeling Algorithms

Iterative modeling algorithms (IMAs) generate a search space $\mathcal{M}$ of models by repeatedly selecting a model $m(\cdot) \in \mathcal{M}$ and adding a component c_i from a list of components $\mathcal{C} = c_1, c_2, ..., c_n$ to $m(\cdot)$, producing $m(\cdot, c_i)$. For example, $m(\cdot)$ may be the regression equation $\hat{y} = \beta_3 c_3 + \beta_1 c_1$, and $m(\cdot, c_5)$ is $\hat{y} = \beta_3 c_3 + \beta_1 c_1 + \beta_5 c_5$. Generally, IMAs do not add every possible component to each model $m(\cdot)$—this would result in exhaustive search—but rather, they add the component that appears best according to some evaluation function $x_i = \mathcal{V}(c_i, m(\cdot), \mathcal{S})$. We call x_i the *score* of component c_i given model $m(\cdot)$ and a sample of data $\mathcal{S}$. For example, $\mathcal{V}$ might compute information gain or classification accuracy for decision tree induction algorithms, F ratios for stepwise multiple regression algorithms, and so on. We may define a general IMA algorithm as follows:

IMA: Initially, $\mathcal{M}$ contains the empty model $m()$. Now iterate:

1. Select a model $m(\cdot) \in \mathcal{M}$
2. Remove components from $\mathcal{C}$ on logical grounds if necessary, producing $\mathcal{C}'$. For example, regression models shouldn't contain multiple occurrences of the same variable; whereas decision trees can in some circumstances.

3. Find the component, $c_{max} \in \mathcal{C}'$, with the highest value $x_{max} = max(x_1, x_2, ...x_n)$, where $x_i = \mathcal{V}(c_i, m(\cdot), \mathcal{S})$

4. If $x_{max} > T_{\mathcal{V}}$, where $T_{\mathcal{V}}$ is a possibly dynamic threshold value, then add c_{max} to $m(\cdot)$.

5. Revise $\mathcal{M}$ by adding $m(\cdot, c_{max})$ and perhaps removing one or more models.

IMA terminates when no component can be added to any $m(\cdot) \in \mathcal{M}$ according to step 4.

A model $m(\cdot)$ *overfits* a dataset $\mathcal{S}$ when it includes one or more components c_i that have sufficient scores $x_i > T_{\mathcal{V}}$ given $\mathcal{S}$, but c_i would not have sufficient scores in general—that is, in other datasets drawn from the same population or in the population itself. Obviously, overfitting can occur if the threshold $T_{\mathcal{V}}$ is set too low. Said differently, if $T_{\mathcal{V}}$ is set in a way that underestimates the distribution of x_{max}, then overfitting will occur. In particular, if $T_{\mathcal{V}}$ is based on the distribution of scores x_i instead of the distribution of maximum scores x_{max} then overfitting is inevitable.[1] For example, when decision tree induction algorithms use a threshold $T_{\mathcal{V}}$, they almost always base it on the distribution of x_i instead of x_{max}, which is why they overfit, often dramatically.

Clearly, $T_{\mathcal{V}}$ must respect the distribution of x_{max}, so we begin by examining this distribution under some simplifying assumptions. We focus on the probabilities $Pr(x_{max} \geq k)$ and $Pr(x_i \geq k)$, and on the expected values $E(x_{max})$ and $E(x_i)$. In general, the distribution of x_i underestimates the probability of x_{max}. Then we consider how $T_{\mathcal{V}}$ is set, focusing on the common view of $T_{\mathcal{V}}$ as a critical value in a reference distribution. It will then be obvious how the problem of overfitting is a version of the classical statistical problem of multiple comparisons. This equivalence suggests numerous overfitting-avoidance techniques, which have been tested empirically (see [3, 4]).

[1] In fact, overfitting can occur even when the appropriate reference distribution is used, but its probability can be controlled and made arbitrarily small.

3 The Distribution of the Maximum Score

Recall that a *score* is an evaluation of a component c_i that IMA is considering adding to a model $m(\cdot)$: $x_i = \mathcal{V}(c_i, m(\cdot), \mathcal{S})$. Suppose IMA is considering n components $c_1, c_2, \ldots, c_n$ with scores $x_1, x_2, \ldots, x_n$. Each score is the value of a random variable. The distribution of the maximum score will depend on the distributions of the random variables, and, in general, the variables are not identically and independently distributed (i. i. d.). The following results are for i. i. d variables, and for independent but not necessarily identically distributed variables. (Although we have not extended the results to non-independent variables, the errors introduced by non-independence are small relative to the errors incurred by not using the distribution for the maximum. See Sec. 5).

For simplicity and concreteness, assume x_1 and x_2 are random variables drawn from a uniform distribution of integers $(0 \ldots 6)$. The distribution of $max(x_1, x_2)$ 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 i. i. d. and uniform, every joint event has the same probability, 1/49, but the probability of a given maximum score is generally higher; for example, $Pr(max(x_1, x_2) = 4) = 9/49$. In fact, the probability $Pr(max(x_1, x_2) = k)$ increases with k; for example, $Pr(max(x_1, x_2) = 6) = 13/49$.

	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
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 random variables, each of which takes integer values (0...6).

For i. i. d. random variables $x_1, x_2, \ldots, x_n$, it is easy to state the relationship between cumulative probabilities of individual scores and cumulative probabilities of maximum scores:

$$\text{If } Pr(x_i < k) = q, \text{ then } Pr(max(x_1, x_2, \ldots, x_n) < k) = q^n. \tag{1}$$

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:

$$\text{If } Pr(x_i \geq k) = p, \text{ then } Pr(max(x_1, x_2, \ldots, x_n) \geq k) = 1 - (1 - p)^n \tag{2}$$

These expressions and the distribution in table 1 make clear that the distribution of any random variable x_i from i. i. d. variables $x_1, x_2, \ldots, x_n$ underestimates the distribution of the maximum $x_{max} = max(x_1, x_2, \ldots, x_n)$. $Pr(x_i \geq k)$ underestimates $Pr(max(x_1, x_2, \ldots, x_n) \geq k)$ for all values k 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 the number of random variables, $x_1, x_2, \ldots, x_n$. Imagine three variables distributed in the same way as the two in table 1. Then,

$$\begin{aligned} Pr(x_i \geq 4) &= 3/7 = .43 \\ Pr(max(x_1, x_2, x_3) \geq 4) &= 1 - (1 - 3/7)^3 = .81. \end{aligned}$$

The distribution of x_i underestimates $Pr(max(x_1, x_2, x_3) \geq 4)$ by almost one half its value.

The expected value x_i, $E(x_i)$, generally underestimates the expected value of the maximum. This is easily demonstrated for two random variables x_1 and x_2 which are statistically independent but not necessarily identically distributed; the extension to more independent variables is obvious. The expected values of x_1 and x_2 are

$$E(x_1) = \sum_{i=1}^{n} x_{1_i} Pr(x_{1_i}), \; E(x_2) = \sum_{j=1}^{n} x_{2_j} Pr(x_{2_j}).$$

Likewise, the expected value of $max(x_1, x_2)$ is

$$E(max(x_1, x_2)) = \sum_{i=1}^{n} \sum_{j=1}^{n} max(x_{1_i}, x_{2_j}) Pr(x_{1_i}) Pr(x_{2_j}) \quad (3)$$

$$= \sum_{i=1}^{n} Pr(x_{1_i}) \sum_{j=1}^{n} max(x_{1_i}, x_{2_j}) Pr(x_{2_j}). \quad (4)$$

For any value x_{1_i}, $max(x_{1_i}, x_{2_j}) \geq x_{2_j}$. Consequently,

$$\sum_{j=1}^{n} max(x_{1_i}, x_{2_j}) Pr(x_{2_j}) \geq E(x_2) \quad (5)$$

Thus, expression 4 becomes an inequality:

$$\begin{aligned} E(max(x_1, x_2)) &\geq \sum_{i=1}^{n} Pr(x_{1_i}) E(x_2) \\ &\geq E(x_2) \sum_{i=1}^{n} Pr(x_{1_i}) \\ &\geq E(x_2) \end{aligned}$$

We can prove $E(max(x_1, x_2)) \geq E(x_1)$ in the same way. In sum,

$$E(max(x_1, x_2)) \geq max(E(x_1), E(x_2)) \quad (6)$$

In fact, $max(E(x_1), E(x_2))$ nearly always underestimates $E(max(x_1, x_2))$; more dramatically as the number of random variables increases.

These properties of the distribution of x_{max} depend on $x_1, x_2, \ldots, x_n$ being independently (if not identically) distributed. In the general case, where $x_1, x_2, \ldots, x_n$ are dependent, the probability $Pr(max(x_{1_i}, x_{2_j}, \ldots, x_{n_m}) \geq k)$ is not so easy to estimate (but see [2]). It is not simply a product of probabilities, as in expressions 2 and 4, because $Pr(a, b) \neq Pr(a)Pr(b)$ when a and b are dependent. Empirical assessments of the effects of nonindependent variables are described in Section 5.

4 Underestimation and Overfitting

Underestimating the maximum of n random variables can lead to overfitting. Recall that IMA adds component c_i to model $m(\cdot)$ when c_i is the best component (step 3) and c_i's score, x_i, exceeds the threshold $T_{\mathcal{V}}$ (step 4). There are many ways to set $T_{\mathcal{V}}$, but however one does it, $T_{\mathcal{V}}$ ought to reflect the number of components being considered, the variances of the distributions of the components, the size of sample $\mathcal{S}$, and the number of components already in model $m(\cdot)$. These factors suggest treating $T_{\mathcal{V}}$ as a critical value in a reference distribution; said differently, $x_i \geq T_{\mathcal{V}}$ can be tested with the machinery of statistical hypothesis testing. In fact, this is how many IMA algorithms decide whether to add components. We will briefly review the logic of statistical hypothesis testing.

Suppose we want to test whether a component, c_1, contributes enough to model $m(\cdot)$ to warrant generating a new model $m(\cdot, c_1)$. The usual approach is to derive a reference distribution F_1, for the scores, x_{1_i}, under the *null hypothesis*, H_0, that c_1 contributes *nothing* to $m(\cdot)$. Then, given a particular score $x_1 = k$, one calculates the probability $p = Pr(x_1 \geq k)$, and if it is very low, one rejects H_0 and concludes that c_1 probably does contribute something to $m(\cdot)$. The probability p bounds one's confidence in this conclusion. Typically, one selects a high quantile of F_1, say, the 95th quantile, $F_1(95)$. If $x_1 > F_1(95)$, then one rejects the hypothesis that c_1 contributes nothing to $m(\cdot)$, with a probability of error $p \leq .05$.

$F_1(95)$ is called the .05 *critical value* for the reference distribution F_1.

The hypothesis testing strategy can be misapplied in incremental modeling algorithms, with overfitting as the consequence. Here is the *incorrect* implementation of hypothesis testing in IMA:

Incorrect Hypothesis Testing in IMA: For a given model $m(\cdot)$, and components $\mathcal{C}' = c_1, c_2, \ldots, c_n$ with scores $x_1, x_2, \ldots, x_n$,

1. Find the best component c_i for which $x_i = x_{max} = max(x_1, x_2, \ldots, x_n)$.

2. Formulate the null hypothesis that c_i contributes nothing to $m(\cdot)$ and derive the reference distribution F_i under this hypothesis.

3. Set $T_{\mathcal{V}} = F_i(95)$ (or some other confidence level). If $x_i \geq T_{\mathcal{V}}$ reject the null hypothesis and add c_i to $m(\cdot)$.

In this procedure, the null hypothesis, and thus the reference distribution, are incorrect. The correct null hypothesis is, "The *best* of n components adds nothing to the model," and the correct reference distribution is the distribution of F_{max} under this null hypothesis. It is easy to see how one might erroneously use F_i to test x_i when x_i is the maximum score, but F_i underestimates F_{max}—as we demonstrated earlier for i. i. d. variables, and have shown to be generally true even for non-independent variables—so x_i might easily exceed $F_i(95)$ but fall short of $F_{max}(95)$.

It is now clear how this procedure causes overfitting: In general a reference distribution F_i will underestimate F_{max}, so any value $T_{\mathcal{V}}$ based on F_i will be too low. Thus, components will be added because their scores seem statistically unlikely (e.g., $x_i \geq F_i(95)$) when, according to the correct reference distribution, they are not unlikely at all (i.e, $x_i < F_{max}(95)$).

Equation 2 provides an estimate of the probability of overfitting for any given model $m(\cdot) \in \mathcal{M}$. For example, if any one of ten components could be added to a model, and the components' scores are i. i. d., and we use a 0.10 critical value for F_i instead of for F_{max} as $T_{\mathcal{V}}$, then the probability of overfitting is

$$1 - (1 - 0.10)^{10} = .6513.$$

Keep in mind that this result characterizes the probability of incorrectly adding a *single* component to a model. After adding one component, most modeling algorithms then consider adding another, and another, and each of these decisions also has an elevated probability of being incorrect. One can easily build models in which *most* of the components shouldn't be there. Decision tree induction algorithms, for instance, are exquisitely prone to overfitting [4, 8].

5 The Effects of Dependent Scores

The distribution F_i underestimates F_{max} to the greatest degree when scores $x_1, \ldots, x_n$ are independent. Dependency among the scores lessens the degree to which F_i underestimates F_{max} but doesn't eliminate it unless the scores are all completely dependent, that is, identical. You can see this effect graphically in Figure 1. The left panel represents the distribution of independent, uniform, random variables x_1 and x_2. Arbitrarily, we selected 90% of the range of each variable as a threshold for "high" scores; hence, the blackened dots represent events in which either x_1 or x_2 or both had high values. Twenty-four of one hundred values are high. The right panel represents the distribution of x_1 and $x_1 + \epsilon$, where ϵ is an error term; clearly, these quantities are not independent. Once again we define high values to be those in the upper 10% of the range of each variable. Fewer high values (14 of 100) are seen in this plot. The reason is easy to see: If x_1 is low then $x_1 + \epsilon$ is less likely to be high than x_2 is, simply because $x_1+\epsilon$ is not independent of x_1 but x_2 is. Thus, $max(x_1, x_2)$ will be a high value more often than $max(x_1, x_1+\epsilon)$ will be. Said differently, the distribution of x_{max} is less weighty at its upper extremes if $x_1, \ldots, x_n$ are dependent.

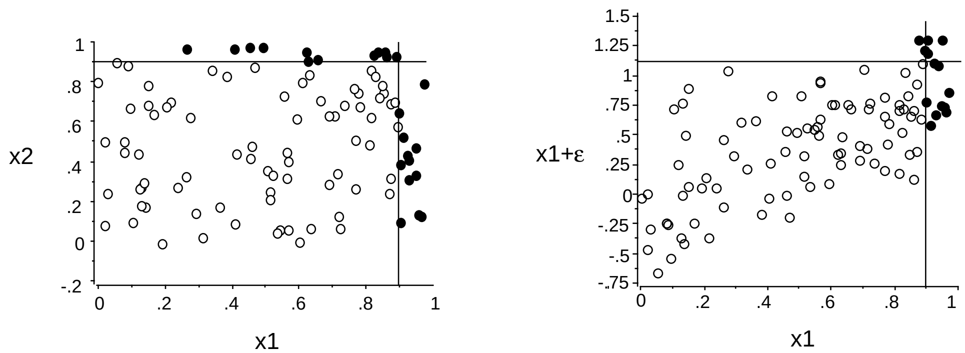

Figure 1: The effect of dependencies on the frequency of extreme values.

Thus, one would expect the degree of overfitting to decrease as the dependency among n scores increases. Figure 2 illustrates how non-independence of the scores $x_1, \ldots, x_n$ affects the probability of incorrectly rejecting the null hypothesis and thus accepting a model component incorrectly. In each trial, ten binary attributes with equal class probability and 50 instances were compared to a randomly-generated binary classification variable. The scores for these attributes, $x_1, \ldots, x_{10}$, measure strength of association between the attribute and the binary classification variable. These scores are expected to be small because, as noted, the classification variable is random. The horizontal axis of Figure 2 is the median pairwise correlation between the attributes. The leftmost value, 0.50, means the attributes are i. i. d. and higher values reflect increasing dependence among the attributes and thus their scores. The chi-square scores were compared with conventional reference distributions for chi-square tests with critical value $F(90)$. That is, the probability of incorrectly rejecting the null hypothesis is 0.10. When the attributes are i. i. d., the probability of spuriously accepting one into a model on the basis of chi-square scores is roughly 0.65—no 0.10 as it should be. As the attribute scores become more dependent, the probability of overfitting drops. In the extreme case of perfect correlation, all the attributes behave identically, so either they all reject the null hypothesis or none does.

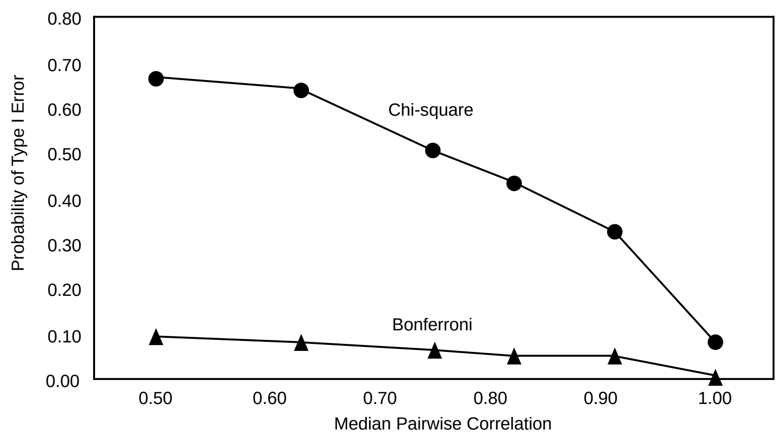

Figure 2: The joint effects of underestimating F_{max} and non-independent scores.

One cannot exploit this relationship to avoid overfitting, however, because the degree of dependence among attributes is a property of one's dataset, not a property one can manipulate. However, one *can* control the degree of overfitting, very precisely, with the methods discussed in section 7.

6 Underestimation and Multiple Comparisons

There is a direct mapping from the problem of estimating the distribution of the maximum to the well-known problem of multiple or simultaneous comparisons.

Suppose $\mathcal{C} = c_1, c_2, \ldots, c_n$ with scores $x_1, x_2, \ldots, x_n$, and assume these scores are independently and identically distributed (i. i. d.) random variables. Consider two null hypotheses:

Simultaneous : Every component c_i contributes nothing to model $m(\cdot)$.

Max : The best component, c_{max}, contributes nothing to model $m(\cdot)$.

Suppose one tests each of the **simultaneous** null hypotheses against a reference distribution, F_i (which is the same for all scores because they are i. i. d.) For example, one tests c_1 by comparing x_1 to F_i, then one tests c_2 by comparing x_2 to F_i, and so on. Alternatively, one might test the **max** null hypothesis by comparing x_{max} to F_i. Assuming i. i. d. scores, these testing strategies produce identical *type I errors*.

A type I error involves rejecting the null hypothesis when it is true (i.e., accepting a model component when one shouldn't). In the simultaneous case, above, α_c denotes the probability that a test of a single component will erroneously reject the null hypothesis, and α_e denotes the probability that at least one test of n components will erroneously reject the null hypothesis. Think of α_c as a bias on a coin: if $\alpha_c = .05$, then with probability .95, a toss will land tails, and no error will occur. Clearly, if one tosses the coin twice, the probability of landing tails twice, and avoiding a type I error, is $.95^2$. If one performs n independent statistical tests, each with α_c probability of a type I error, then the probability of at least one type I error in all n tests is

$$\alpha_e = 1 - (1 - \alpha_c)^n \tag{7}$$

α_e is called the *experimentwise* type I error.

Now suppose we have $x_1, x_2, \ldots, x_n$ i. i. d. random variables, and we set k so that $Pr(x_i \geq k) = \alpha_c$. What is the probability that the maximum of the variables exceeds k? From expression 2 we see

$$Pr(max(x_1, x_2, \ldots, x_n) \geq k) = 1 - (1 - \alpha_c)^n. \tag{8}$$

That is, the probability of a type I error committed by comparing $max(x_1, x_2, \ldots, x_n)$ to a reference distribution for F_i is identical to the experimentwise probability of a type I error in n comparisons. The **simultaneous** and **max** null hypotheses, above, are identical in terms of the resulting type I error probabilities, that is, they are identical in terms of the probability of accepting a model component erroneously. This is not surprising, because finding the maximum of n random variables and then testing whether it exceeds a critical value requires n pairwise comparisons of random variables.

The upshot of this result is that we may apply techniques developed for problems of multiple comparisons (such as the Bonferroni adjustment) to control overfitting [7, 6]. All these techniques adjust $T_{\mathcal{V}}$ to account for the fact that we are testing not one, but the best of several, model components.

7 Avoiding Overfitting

There are two general ways to control overfitting. One is to use the reference distribution F_i, but adjust the critical value to control the probability of one or more type I errors when $n > 1$ components are evaluated. The other is to derive the reference distribution of F_{max}. Each has advantages and disadvantages.

Adjusting the Critical Value in F_i. Referring back to the IMA procedure, suppose one is evaluating components $\mathcal{C} = c_1, c_2, \ldots, c_n$ with χ^2 scores $\chi_1^1, \chi_2^2, \ldots, \chi_n^2$. ($\chi^2$ is a simple measure of classification accuracy; see [6]) We know the reference distribution for χ^2 scores under the null hypothesis that a component is independent of the class variable: It is a chi-square distribution with $i - 1$ degrees of freedom, where i is the number of unique values of the class variable. If $i = 2$, for example, the .1 critical value for rejecting the null hypothesis is 2.7. If we set $T_{\mathcal{V}} = 2.7$ and compare the best score $\chi^2_{max} = max(\chi_1^1, \chi_2^2, \ldots, \chi_n^2)$ to this threshold, then the probability of incorrectly rejecting the null hypothesis (i.e., overfitting) is not .1 as we hoped but approximately $1 - (1 - .1)^n$. To control this type I error probability we can adjust the critical value, making it more stringent, with a Bonferroni adjustment [6, 7, 4]. The line marked "Bonferroni" in Figure 2 is for Bonferroni-adjusted chi-square scores, and when the attributes are i. i. d., the adjustment gives us exactly the probability of overfitting that we stipulated, 0.10, but as the attribute scores become more correlated, the Bonferroni adjustment becomes overly stringent. Nevertheless, the Bonferroni adjustment is very simple. A companion paper to this one presents detailed empirical analysis of this approach to overfitting [4]

Deriving the Distribution F_{max}. Instead of using F_i with adjusted critical values, we may find the reference distribution that we really want, F_{max}, and set $T_{\mathcal{V}} = F_{max}(90)$ or some other critical value. F_{max} can be derived by randomization [3], which is a Monte Carlo sampling procedure [1]. The advantage of this approach is that it gives precise control of type I errors even when the attributes are not independent. Its only disadvantage is the computational cost of randomization.

8 "My algorithm isn't statistical, so why does it overfit?"

Any modeling algorithm that compares the best of n scores to a threshold can overfit if the scores are *estimates* and if $n > 1$. The comparison need not be an explicitly statistical test (e.g., a chi-square test). All machine learning algorithms are prone to overfitting because they all work with *samples*—training data—and thus component scores are always estimates. Whether or not a machine learning algorithm contains explicit statistical tests, it is fundamentally a statistical algorithm in the sense that it bases decisions on samples of data.

Some learning algorithms do not appear to compare scores to thresholds, and so appear to be immune to overfitting. But closer examination shows otherwise. For example, some algorithms hill climb through a space of models, adding the component that most improves the score of the resulting model. Yet implicit in these algorithms is a statistical judgment: Is the model I'm considering *really* uphill from the current model, as it appears to be on the basis of sample data, or is the current model actually a (local) maximum? Whether or not the question is framed as a statistical test, it is a statistical question: Is the node that is reached by following the steepest slope really uphill from the current node, or is the apparent slope a spurious result of sampling? The probability of a spurious result increases with the branching factor of the search. Hill-climbing algorithms that work with samples are prone to overfit, to "go too far" in the search space, to end up at a node that appears to be a (local) maximum but isn't, really.

Although overfitting is a pathology of statistical decision making, it is wrong to think that one can avoid overfitting by keeping statistical hypothesis testing out of one's algorithms. The only way to control overfitting precisely is to understand it statistically and build appropriate statistical tests into one's algorithms. Said differently, the only way to control overfitting precisely and predictably is to make the threshold $T_{\mathcal{V}}$ a critical value in the reference distribution of the maximum score. A Bonferroni adjustment to $T_{\mathcal{V}}$ will suffice but doesn't provide equally precise control of overfitting. Either method is more principled, and so provides better control of overfitting, than ad hoc pruning methods. Ad hoc thresholds $T_{\mathcal{V}}$ for component scores demonstrably do not control overfitting [8].

No amount of fiddling with scoring metrics (e.g., classification accuracy, information gain, χ^2, etc.) will provide control of overfitting, either. Overfitting afflicts algorithms that use χ^2 because these scores are not compared to the appropriate reference distribution—χ^2_{max}—or corrected for the number of scores. The same is true for information gain, classification accuracy, and the rest. The score isn't the problem, the reference distribution of the score is.

9 Acknowledgments

This research is supported by DARPA/Rome Laboratory contract F30602-93-C-0076 and by a Subcontract from Sterling Software Inc. 7335-UOM-001 (DARPA/Rome Laboratory F30602-95-C-0257). 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.

References

[1] Cohen, P.R. (1995). *Empirical Methods for Artificial Intelligence.* Cambridge, MA: MIT Press.

[2] J. Galambos (1978). The Asymptotic Theory of Extreme Order Statistics. New York: Wiley.

[3] Name removed for blind review. (1992). Induction with Randomization Testing: Decision-Oriented Analysis of Large Data Sets. Identifying information removed for blind review.

[4] Name removed for blind review. (1997). Adjusting for multiple testing in decision tree pruning. Submitted to ICML-97.

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

[6] Kotz, S. and N.L. Johnson (Eds.) (1982-1989). *Encyclopedia of Statistical Sciences.* New York: Wiley.

[7] Miller, Rupert G. (1981). Simultaneous Statistical Inference. New York: Springer-Verlag.

[8] Name removed for blind review (1997). The effects of training set size on decision tree complexity. Submitted to ICML-97.