Abstract

This section discusses basic features of error estimators, including bias, variance, and loss functions. It outlines the logic behind classical hypothesis testing and explains the special challenges faced by knowledge discovery algorithms that search large model spaces. It discusses the statistical effects of multiple comparison procedures (MCPs), and several methods to adjust for those effects, including mathematical adjustments, cross-validation, and randomization tests. Finally, it outlines the basic concepts behind overfitting reduction and pruning.

8.1 Introduction

Many knowledge discovery algorithms estimate error and use those estimates to make important decisions. In particular, many algorithms search a space of alternative models and attempt to select models that best capture useful regularities in the data. Without accurate error estimates, and effective methods to interpret those estimates, algorithms will select suboptimal models and infer the existence of structure in purely random data. Accurate estimates can be obtained, but often only with substantial additional data or computation.

8.1.1 Error Estimation

Knowledge discovery algorithms use error estimates to determine the relative utility of alternative models and select among alternative search paths. In addition, human designers evaluate algorithms by estimating the error of models the algorithms produce.

The error associated with a model $m$ can be estimated by comparing the predictions of $m$ to known values in a data sample $S$. Many different statistics can be used to estimate the error of a model, including the percentage of misclassified instances, gain or gain ratio, chi-square, and G. Each of these statistics produces a scalar value or score denoted $x$. Some statistics take additional inputs (e.g., a loss function or prior knowledge), and some knowledge discovery systems estimate more than one statistic related to error (e.g., precision and recall).

For example, consider estimating the error of a simple model $m_1$ on a hypothetical data set $S_1$ that records diagnostic test results for 73 hospital patients. The aim of $m_1$ is to predict whether each patient has a given disease. Figure 1 shows the results of applying the model to all patients. While far from perfect, the model appears to have some predictive ability. The value of the chi-square statistic for this table is 3.946. In the next section, we will interpret this score using a statistical hypothesis test.
The calculation of many statistics that measure classification error can be explained in terms of a contingency table. For example, using the cell designators in Figure 1 and the additional notation $N=a+b+c+d$, we can calculate statistics such as the percent misclassification $= (b+c)/N$, the false positive rate $= c/(c+d)$, and the positive predictive value $= a/(a+b)$.

The chi-square and G statistics are functions of all cells of a table. For example, chi-square for the table in Figure 1 can be calculated as:

$$\frac{N(ad - bc)^2}{(a + b)(c + d)(a + c)(b + d)}$$

In general, chi-square is the sum, over all cells, of the normalized squared difference between the observed cell count and the expected cell count. Typically, the expected cell count is determined by assuming independence between the row and column variables (e.g., treatment outcome and gender).

The score $x$ is an estimate of a population parameter $\Psi$. Given the population of all possible instances, $\Psi$ could be calculated directly. In practice, however, the values of parameters are estimated based on samples of instances drawn from the population. Samples are used because of logistical limitations (e.g., it may be impractical to survey all possible customers), temporal considerations (data are only available on current customers, but inferences are desired about all current and future customers), or efficiency reasons (the computational complexity of knowledge discovery algorithms almost always depends on the number of instances in a data set and running a particular algorithm on an entire population may be infeasible). Statistics are calculated on these samples and used to estimate population parameters. Many different samples could, in theory, be drawn from the same population, and few samples will produce precisely the same score. Thus, an individual score almost always varies somewhat from the value of the population parameter.
Bias and variance are two criteria often used to describe the quality of the scores produced by an estimator. The bias of an estimator is the difference between the expected value of its scores $E(X)$ and the value of the population parameter $\Psi$. An estimator is unbiased if $E(X) = \Psi$. While any particular score $x$ produced by an unbiased estimator may be greater or less than $\Psi$, scores will not vary systematically in either direction. Figure 2 shows the distributions of two estimators. One is unbiased and the other has a positive bias. The variance of an estimator is the second moment about the mean of its distribution, or $E(X-\mu)^2$, where $\mu$ is the mean of $X$. Variance measures the dispersion of scores around their mean. Given two unbiased estimators, the one with lower variance is preferred. The two estimators in Figure 2 have different variance. The figure shows one situation where a biased estimator might be preferable to an unbiased one because it has lower variance.

A statistic that estimates error can also be described in terms of its loss function. Loss functions describe the consequences or loss associated with a particular estimate. For example, a loss function in a classification task might reflect the costs associated with different types of misclassifications. Many classification tasks are assumed to have zero-one loss. Either a class prediction is correct (a loss of zero) or incorrect (a loss of one). No partially correct classifications are possible and no one misclassification has more serious consequences than another. In contrast, the loss associated with predictions of continuous functions generally vary smoothly between zero and one. Many systems that make predictions of continuous functions use quadratic loss, where the loss is proportional to the squared difference between the predicted and actual values. Despite the widespread use and theoretical tractability of zero-one loss, it is invalid for many tasks (Provost, Fawcett, and Kohavi 1998). Different types of misclassification

---

1 The fields of statistics and machine learning differ in their definitions of the term "bias." Here we use the statistical definition.
2 That knowledge can sometimes be implicit. For example, Fisher's Exact Test calculates a p-value without explicit reference to a sampling distribution (see below).
often have different costs, and these costs should affect which models are preferred by knowledge discovery algorithms.

8.1.2 Statistical tests of hypotheses

An hypothesis test is one common method of interpreting a score. Hypothesis tests are one way of making statistical inferences about the population, based on a score calculated from a sample. Hypothesis tests compare a score to the distribution of scores that would be expected given that a particular null hypothesis is true of the population. The null hypothesis (denoted H0) makes a quantitative statement about a population parameter (e.g., error on the population) that can be used to derive an expected distribution of scores called a sampling distribution. The null hypothesis and its associated sampling distribution provide a quantitative context for interpreting sample scores.

Once the sampling distribution has been determined, an hypothesis test is performed to determine the probability that a given score, or a more extreme score, would occur given that the null hypothesis (H0) is true. This probability is referred to as a p-value or merely p. To determine whether to accept or reject the null hypothesis, p is often compared to α, a fixed probability threshold that indicates the maximum acceptable probability of rejecting H0 when it is true. If p ≤ α, then the null hypothesis is inferred to be false and is rejected. If p > α, then the null hypothesis cannot be rejected. Commonly-used values of α include 0.05, 0.01, and 0.001. The smallest values of α are used when the loss associated with incorrectly rejecting the null hypothesis is greatest.

The form of the null hypothesis depends on what is being tested. For example, to compare the performance of two models, a possible null hypothesis is that their performance difference in the population is zero. To examine the performance of a single model, a possible null hypothesis is that the model's predictions are independent of the true class labels of instances in the population. Another possible null hypothesis might be that the predictive accuracy of a model on the population does not exceed a given value. In each case there is an alternative hypothesis, H1, that is paired with the null hypothesis, and it represents the necessary conclusion if the null hypothesis is rejected. For example, if H0 states that two models have a performance difference of zero, then H1 would state that the difference is non-zero. If, based on the hypothesis test, H0 is rejected, then H1 is accepted.

In nearly all cases, knowledge of the sampling distribution for a statistic is a prerequisite for hypothesis testing. In some cases, the sampling distribution for a statistic has been derived theoretically. For other statistics, sampling distributions must be derived empirically using computationally-intensive techniques such as randomization (see Section 8.1.6).

As an example of a hypothesis test, consider the example of model evaluation discussed in the previous section. As already noted, a chi-square statistic can be calculated from the contingency table in Figure 1, producing a score of 3.946. What does this value imply about the relationship between the model’s predictions and the actual disease state in the population of hospital patients? One reasonable null hypothesis is that the model’s predictions and disease state are independent — knowing the prediction of the rule indicates nothing about disease state. Under this null hypothesis, the chi-square statistic has a known sampling distribution parameterized by the number of cells in the table. Comparing the sample score of 3.946 to the sampling distribution
of chi-square yields \( p = 0.0470 \) — less than 5 percent of the sampling distribution is greater than or equal to 3.946 (see Figure 3). Given \( \alpha = 0.05 \), the sample score is judged unlikely to have been drawn from the sampling distribution, and the null hypothesis of independence is rejected. Given \( \alpha = 0.01 \), the reverse would be concluded, and the null hypothesis would not be rejected.

![Figure 3: Sampling distribution and sample score](image)

An alternative approach to the chi-square test is Fisher's Exact Test (Sachs 1982). As its name implies, Fisher's Exact Test produces exact values of \( p \), rather than the approximations provided by comparing the value of statistics to a theoretically-derived sampling distribution. Fisher's original formulation applies only to 2\( \times \)2 contingency tables such as the one shown in Figure 1, but generalizations of the test have been formulated for larger tables (Sachs 1982, Mehta and Patel 1983). Fisher's exact test can be calculated as the sum of terms of the hypergeometric distribution:

\[
\frac{(a + b)! (c + d)! (a + c)! (b + d)!}{n!} \sum \frac{1}{\prod a! b! c! d!}
\]

where the summation is over all tables (such as that in Figure 1) that have both identical marginal sums and equal or more extreme cell counts, given the null hypothesis of independence.

Fisher's Exact Test calculates \( p = 0.0618 \) for the example above, rather than the value \( p = 0.0470 \) determined using chi-square. As this example demonstrates, the chi-square test can introduce small but potentially important errors. Fisher's test is particularly useful when cell counts in one or more cells are small (e.g., less than 6), because statistics such as chi-square and G can become inaccurate under such conditions.

When testing an hypothesis, two types of errors are possible. Type I errors are committed by rejecting the null hypothesis when it is true. Type II errors are committed by accepting the null hypothesis when it is false. Hypothesis tests can control the probability of a Type I error, but do not protect against Type II errors. For example, if a very small sample is used to test the null
hypothesis of independence between gender and treatment outcome, then we are likely to obtain a very large \( p \)-value, and thus be forced to accept the null hypothesis. However, the null hypothesis could still be false, and we would commit a Type II error.

Tests predisposed to Type II errors are said to have low statistical power — we will frequently accept the null hypothesis, even when it is false. The probability of a Type II error is denoted by \( \beta \), where \( \text{power} = 1 - \beta \). Power depends on several factors, including the size of the effect (in this case, the magnitude of the correlation between gender and treatment outcome), the sample size, the inherent variability of scores, and \( \alpha \). Statistical power is ignored in most knowledge discovery algorithms, even in those algorithms that explicitly test statistical significance. However, as knowledge discovery algorithms increase in sophistication, power calculations will become increasingly important. Power calculations can help determine when additional instances should be sampled and, conversely, when the sample is already large enough to identify extremely weak relationships.

Hypothesis tests and power calculations only answer questions about the probabilities of different types of errors. Such tests do not indicate whether a given relationship is important or useful. It is quite possible for a model to represent a relationship that is statistically significant, but practically useless. This is particularly likely with extremely large samples, where statistical power is high, and extremely weak relationships can be identified with high statistical confidence. Similarly, it is possible for a relationship to be valid and useful, but not statistically significant. This is particularly likely with extremely small samples, which lack the statistical power to identify all but the strongest relationships. While statistical significance is one of the most basic questions to ask about models and their associated scores, many other possible questions can and should be asked. Among these is whether a relationship is of practical importance within a given domain. For additional discussion about the limitations of significance tests, see Morrison and Henkel (1970) and Glymour, Madigan, Pregibon, and Smyth (1997).

### 8.1.3 Multiple comparisons

Search is a hallmark of most knowledge discovery algorithms. For example, algorithms such as APRIORI (Agrawal, Imieliński, and Swami 1993) and BRUTE (Segal and Etzioni 1994) conduct the equivalent of exhaustive search. Greedy algorithms, such as C4.5 (Quinlan 1993), CN2 (Clark and Niblett 1989), and FOIL (Quinlan 1990), search a space of components that can be added to an existing model. Wrapper approaches (e.g., Kohavi and John 1997) search a space of possible algorithm parameters in order to maximize the performance of induced models.

Each of these search procedures has a similar form:

1) *Generate multiple items* — Generate multiple models \((m_1, m_2, m_3, \ldots, m_n)\), components, or algorithm parameters.

2) *Estimate a score for each item* — Estimate a score \(x_i = f(m_i, S)\) for each item based on a data sample \(S\) (often called a training set).

---

A few knowledge discovery algorithms do not conduct explicit search. For example, neither simple Bayesian classifiers (Domingos and Pazzani 1997) nor instance-based learning algorithms (Aha, Kibler, and Albert 1991) employ explicit search. However, most algorithms explore some sort of space in search of useful models.
3) Select the item with the maximum score — Select the apparent best item \( \text{argmax}(x_1, x_2, x_3, \ldots, x_d) \), based on the estimated scores.

We refer to this as a multiple comparison procedure (MCP).

MCPs affect the statistical properties of the score of the selected item. Specifically, the number of items generated in step one of an MCP affects the sampling distribution of the maximum score in step three. That is, the extent of the search used to select a given item alters the meaning of the score associated with that item. These effects are discussed both in general and mathematical form, and they are covered in greater detail elsewhere (Jensen and Cohen 2000).

Consider a simple example of an MCP. You are challenged to a game of chance by Sam, a statistically-inclined con artist. The game involves simple dice-rolling. You and Sam each bet $10, and the person with the highest scoring die receives $20 and the other person receives nothing (ties result in another dice roll). Sam gives you a standard six-sided die and guarantees that it is fair; no score is more likely than any other. You roll the die and obtain a score of four. Sam produces ten dice from his pocket, rolls them, and selects one with the maximum score (a six). He then declares himself the winner, and claims the $20. You assert that Sam's use of ten dice is unfair, but Sam replies that his selected die is just like yours — it is a fair die, and it genuinely obtained a score of six. Why is selecting the maximum from among ten dice unfair? Why does it differ from rolling only one die?

One way to understand the difference is to examine the sampling distributions of the scores in each case. Figure 4(a) shows the sampling distribution for \( X_i \), the score obtained by rolling a single die where each score in the set \{1 2 3 4 5 6\} is equally likely. Given this distribution, we can easily calculate the expected value of the die's score — \( E(X_i) = 3.5 \) — and the probability of the score exceeding any particular value — e.g., \( p(X_i > 5) = 0.167 \). Figure 4(b) shows the sampling distribution for \( X_{\text{max}} \), the score obtained by selecting the maximum score from among ten dice. The sampling distribution of \( X_{\text{max}} \) is substantially different from that of \( X_i \). The expected value is much larger — \( E(X_{\text{max}}) = 5.6 \) — and the probability of an extreme value is similarly larger — \( p(X_{\text{max}} > 5) = 0.838 \). Thus, comparing scores drawn from the two distributions is unfair, in the sense that values drawn from the distribution of \( X_{\text{max}} \) will almost always be larger than those drawn from \( X_i \).

![Figure 4: Sampling distributions for one die and ten dice](image-url)

\[
\begin{align*}
E(X_i|H_0) &= 3.5 \\
p(X_i > 5|H_0) &= 0.167 \\
E(X_{\text{max}}|H_0) &= 5.8 \\
p(X_{\text{max}} > 5|H_0) &= 0.838
\end{align*}
\]
Knowledge discovery algorithms produce situations that are directly analogous to this example. Just as Sam started with multiple dice, knowledge discovery algorithms generate multiple models, components, or learning parameters. Just as rolling dice produces a score, knowledge discovery algorithms estimate a score for each model based on a data sample. As noted in section 8.1.2, the scores $x_1, x_2, x_3, \ldots, x_n$ are estimates of some population parameter, and thus the probability distribution of each score has non-zero variance. Any two samples, $S_1$ and $S_2$, may differ in the degree to which they reflect the population, and thus, $f(m_1, S_1) \neq f(m_1, S_2) \neq \Psi$. Therefore, just as Sam selected the die with the maximum score, knowledge discovery algorithms select the item with the maximum score.

If algorithms do not adjust for the effects of multiple comparison procedures (MCPs), they can suffer from various pathologies, including overfitting, oversearching, and feature selection errors. Each of these pathologies results from systematic errors either the parameter estimates or hypothesis tests. MCPs introduce substantial upward bias into parameter estimates, and they can cause large increases in the probability that a hypothesis test will incorrectly reject the null hypothesis. Sketches of both proofs are given below, and formal proofs can be found elsewhere (Jensen and Cohen 2000).

As an example of how MCPs affect parameter estimates, consider $n$ models, each with identical performance. That is, $\Psi_1 = \Psi_2 = \Psi_3 = \ldots = \Psi_n$. Suppose that the score $x_1$ calculated for a single item is an unbiased estimator of $\Psi_1$. Although $x_1$ may be larger or smaller than $\Psi_1$ when calculated on any individual sample, when calculated across all samples, $E(X_i) = \Psi_1$. Now, consider the maximum score $x_{\text{max}} = \max(x_1, x_2, x_3, \ldots, x_n)$. For any given sample, $x_1 \leq x_{\text{max}}$. Said differently, the scores $x_2, x_3, \ldots, x_n$ will never decrease the maximum, and will increase it for some samples. As a result, $E(X_{\text{max}}) > E(X_i)$, and thus $E(X_{\text{max}})$ is a biased estimator of any $\Psi_i$.

As an example of how MCPs affect hypothesis tests, consider $n$ classification models, whose predictions are mutually independent, identically distributed, and uncorrelated with the class label. We can determine a threshold $x_{\text{crit}}$, called the critical value, such that $p(x_i > x_{\text{crit}} | H_0) = \alpha$. If a given score exceeds this critical value, it is deemed statistically significant at the $\alpha$ level, meaning that there is only a probability of $\alpha$ that $x_i$ will equal or exceed $x_{\text{crit}}$ if $H_0$ is true. However, what is the probability that $x_{\text{max}}$ will exceed this critical value if $H_0$ is true? Assuming score distributions are independently and identically distributed (i.i.d.) and that $p = p(x_i > x_{\text{crit}} | H_0)$, then $p(x_{\text{max}} > x_{\text{crit}} | H_0) = 1 - (1 - p)^n$. For example, if $p = 0.05$ and $n = 20$, then $p(x_{\text{max}} > x_{\text{crit}} | H_0) = 0.64$. Hypothesis tests will be incorrect if they use the sampling distribution appropriate to $X_i$. Unfortunately, nearly all theoretical sampling distributions are only appropriate for a single score $X_i$, not the maximum of several scores $X_{\text{max}}$.

The need to adjust for the effects of MCPs is a critical task in knowledge discovery, and it arises in at least two areas. First, empirical evaluations of knowledge discovery algorithms often need to adjust for MCPs. For example, the effects of MCPs underlie the common advice to separate data sets for training and testing when evaluating knowledge discovery algorithms (e.g., Weiss and Kulikowski 1991). It has also been noted as an important factor in empirical comparisons of multiple algorithms (Feelders and Vehokens 1996; Gascuel and Caraux 1992; Salzberg 1997). This advice has been standard in the statistical literature on experimental design for decades (Miller 1981). Second, knowledge discovery algorithms themselves often need to adjust for MCPs, and inadequate adjustments cause a variety of pathological behaviors (Einhorn 1972, Jensen and
Specific pathologies attributable to MCPs include overfitting, oversearching, and feature selection errors.

Overfitting is one of the most commonly identified pathologies of knowledge discovery algorithms. Many algorithms generate models with components that either reduce the model's accuracy on new data, or leave it unchanged (Einhorn 1972; Dietterich 1995; Oates and Jensen 1997, 1998). This behavior, often referred to as overfitting, results from conducting an explicit or implicit hypothesis test with an inappropriate sampling distribution. Algorithms often add components to a model only when the score of those components exceeds some critical value. In many cases, however, algorithms do not adequately adjust this value for the effects of MCPs. Their hypothesis tests use sampling distributions (or critical values) appropriate for $X_i$, not $X_{\text{max}}$, and thus the tests have extremely inflated probabilities of Type I error. Such errors cause algorithms to mistakenly infer that components will improve model accuracy, and they add those components to the model.

Oversearching is a more recently identified pathology of knowledge discovery algorithms. Several studies (Quinlan and Cameron-Jones 1995; Murthy and Salzberg 1995) show that increasing the size of an algorithm's search space can reduce the accuracy of the final model on test data. This contradicts common experience with other AI algorithms, where the quality of the final solution generally increases with the size of the search space. Oversearching occurs because of differing biases in parameter estimates resulting from two MCPs. The maximum scores from two MCPs are compared directly, even though one MCP considers a set of models that is a superset of the other. The maximum score from the larger model space can never be less than the maximum score from the smaller space, although it can be greater. The two maximum scores are not directly comparable because they are drawn from different sampling distributions with different degrees of bias. Algorithms that directly compare these biased scores can construct suboptimal models.

Feature selection errors have been identified most frequently in algorithms for decision tree induction (Quinlan 1988; Fayyad and Irani 1992). Most tree induction algorithms apply an MCP to each variable to select a candidate feature to consider using at the root of a new subtree. For example, an MCP might be applied to a discrete variable (eye-color) to select which binary feature (e.g., eye-color = {blue or green}) provides the greatest increase in classification accuracy if used as the root of a new subtree. An algorithm would generate possible features (e.g., eye-color={blue}, eye-color={blue or green}, eye-color={blue or brown}, etc.), score each feature in the context of a current tree, and select the feature with the maximum score. Algorithms then compare the best features from each variable, select the best overall feature, and install that feature at a node in a tree. During classification, instances for which the feature is true are sent down one branch, and instances for which the feature is false are sent down the other. Features drawn from variables with large features sets (e.g., eye-color or postal-code) are often preferred over features drawn from variables with small sets (e.g., gender), even though the opposite decision would improve performance of the model on test data. As with oversearching, this pathology results from errors in parameter estimates. Algorithms compare the maximum scores from two or more MCPs, where the MCPs consider different numbers of items. Thus, the maximum scores from the MCPs are not directly comparable, because they are biased to differing

---

4 Here we use the terminology suggested by Kohavi and Provost (1998) where eye-color is termed a variable and eye-color={blue or green} is termed a feature.
degrees. The relative ordering of the maximum scores is a poor indicator that the same ordering will hold among the respective population parameter values. As a result, suboptimal features are often selected and added to a model.

The magnitude of the parameter estimation bias introduced by MCPs is determined by several other factors other than the number of items compared. First, increasing the sample size of the training set can reduce the variance of the sampling distribution of each score $X_i$, and this reduces the bias $E(X_{max})-\Psi$. Second, if a small number $n_s$ of population scores are much larger than the other scores, where $n_s << n$, then the bias introduced by an MCP will decrease. Finally, if the scores distributions are highly correlated, then bias will also decrease. Additional discussion and proofs for each of these effects are provided elsewhere (Jensen and Cohen 2000).

### 8.1.4 Mathematical adjustments

One of the simplest ways to adjust for the effects of an MCP is based on simple probability theory. If each score $x_i$ has some probability $p$ of exceeding a given threshold then the probability $p^*$ that the maximum of $n$ scores $X_{max}$ will exceed the threshold can be approximated by

$$p^* = np$$

(1)

if $p$ is small. This equation, known as a Bonferroni adjustment, can result in $p^* > 1$ if $n$ or $p$ are large, so the equation is sometimes written

$$p^* = \min(1, np).$$

A more versatile and precise equation is the Sidak adjustment

$$p^* = 1 - (1-p)^n$$

(2)

Given equation 1 or 2, a per-comparison threshold can be determined such that $p^* = \alpha$. A variety of related techniques have been developed and are discussed in detail elsewhere (Miller 1981).

Techniques such as the Bonferroni and Sidak adjustments are efficient and simple to calculate. Despite several assumptions (discussed below), they can be surprisingly accurate, particularly when $p$ is small (Miller 1981). For these reasons, they have been incorporated into several knowledge discovery algorithms (Kass 1975, 1980; Gaines 1989; Jensen and Schmill 1997; Megiddo and Srikant 1998).

Unfortunately, Bonferroni and Sidak adjustments make several assumptions that are often violated by knowledge discovery algorithms. First, these adjustments assume that the distributions of individual scores are uncorrelated. This assumption is manifestly false for many situations. For example, multiple models with highly correlated scores are generated by many iterative search algorithms that hill-climb in the vicinity of a high-performing model. Second, non-

---

5 The standard deviation of a sampling distribution is called the *standard error*. 

*David Jensen*  
*8/8/00*  
*DRAFT*  
*Version 1.0*
normal or discrete sampling distributions can lead to substantial errors in adjusted probabilities (Westfall and Young 1993).

The effects of correlated scores can be demonstrated with a simple experiment. For each trial, we randomly generate fifty instances with a class variable and twenty predictor variables, each drawn from a uniform distribution of binary values. We introduce varying degrees of correlation among the predictor variables, measured using median pairwise correlation. We search the predictor variables to find the single predictor with the maximum chi-square score, and we accept or reject the hypothesis of independence between the best predictor and the class based on two types of hypothesis tests: an unadjusted chi-square test and an adjusted chi-square test using the Sidak equation. The class and predictors are generated independently, so any rejection of the null hypothesis is necessarily incorrect.

Figure 5 shows the results of this experiment. The x-axis corresponds to the median pairwise correlation among the predictor variables. The results at each level of correlation represents 3000 trials. The y-axis shows the probability of rejecting the null hypothesis with $\alpha = 0.10$. The correct value is shown by the horizontal dotted line at $p = 0.10$. As would be expected, when predictor variables are independent (pairwise correlation 0.5), the unadjusted test performs quite poorly, rejecting the null hypothesis with a probability of approximately $1-(1-0.10)^{20} = 0.8784$. The adjusted test performs quite well, rejecting the null hypothesis with probability of approximately 0.10. However, as correlation among the predictor variables increases, the probability of rejecting the null hypothesis drops for both tests. When all twenty predictors are perfectly correlated (pairwise correlation = 1.0) the adjusted test rejects the null hypothesis with approximately probability $1-(1-0.10)^{1/20} = 0.0053$ and the corresponding probability for the conventional test is 0.10. Perfect correlation among multiple predictor variables is equivalent to testing only a single predictor variable and produces equivalent adjusted and unadjusted p-values. Thus, the adjusted and unadjusted probabilities only bound the correct probability of incorrectly rejecting the null hypothesis, and those bounds become increasingly loose as $n$ or $p$ increase.

---

6 See Freedman, Navidi, and Peters (1986) for a similar experiment.
The presence of correlation among scores need not invalidate the use of adjusted probabilities, but it does require substantial modification in how they are used. Many adjustments have been developed for specific correlation structures among the individual scores (see Miller (1981) for details). In addition, some knowledge discovery techniques (e.g., Kass 1975, 1980) attempt to adjust n to account for the correlation among individual scores, but these methods are heuristic, at best. A widely-applicable and robust theoretical adjustment for MCPs is elusive, and attention has turned to more computationally-intensive techniques (Westfall and Young 1993).

8.1.5 Cross Validation

The simplest method to adjust for the effects of MCPs is to use new data or reserve part of the original data for the sole purpose of adjusting for MCPs. This approach tests the selected item on a data set S’ disjoint from the original sample S used to score all items generated by an MCP. Such a disjoint set can obtained by splitting an original sample into a training set S and a test set S’. However, this approach reduces the size of S, thus increasing the variance of scores and possibly causing the selection of a suboptimal item. Ideally, the size of the S should be maximized.

Cross-validation (Stone 1974; Weiss and Kulikowski 1991; Kohavi 1995) is one way to maximize the size of S while still providing an equivalently-sized test set S’. Cross-validation divides the sample S into k disjoint subsets, Si, where each subset contains approximately 1/k of the total instances in S. Then, for 1 ≤ i ≤ k, an algorithm is applied to the sample S with the subset Si removed (S-Si). The results of that analysis are then evaluated on the sample Si, producing k different nearly unbiased scores. The k scores are then combined to produce a single score by some method of aggregation (e.g., averaging).
Cross-validation with \( k \) trials is often called \( k \)-fold cross-validation. Two-fold cross-validation splits the sample evenly, uses each subset alternately for training and testing, and combines the results. Ten-fold cross-validation is the most common approach, and it has been show empirically to be nearly optimal in most cases (Kohavi 1995). Leave-one-out cross-validation uses \( k=N \), so that each subset \( S_i \) contains only a single instance. In theory, leave-one-out cross-validation maximizes accuracy by providing the largest training sets. In practice, however, this method can increase both the bias and variance of the resulting estimates.

Several knowledge discovery algorithms use cross-validation to adjust for the effects of MCPs. For example, Breiman et al. (1984) apply ten-fold cross-validation within the CART algorithm to determine the value of a pruning parameter they call alpha.\(^7\) Using each subset \( S-S_1 \), CART constructs a large classification tree. CART prunes that tree by systematically increasing the value of alpha, a parameter that controls the extent of pruning. CART tests each pruned tree on the subset \( S_1 \) and records the value of alpha corresponding to the tree with maximum accuracy. CART then combines the ten values of alpha to obtain an estimate of its optimal value, and uses that value to prune the tree constructed from the entire set \( S \).

This example illustrates a common approach to using cross-validation in knowledge discovery. Algorithms use cross-validation to estimate the optimal value of some critical parameter — often a parameter that controls the complexity of an induced model. Then that parameter estimate is used to construct a model from the entire data sample \( S \).

Cross-validation is a robust method that can be used to adjust for the effects of MCPs. It makes relatively few assumptions, and it produces nearly unbiased parameter estimates. However, it increases the computational complexity of an algorithm by a factor of nearly \( k \). In addition, cross-validation has significant drawbacks. It can introduce additional variance into parameter estimates (Kohavi 1995), and, if used incorrectly, it can also introduce significant bias.

Bias can result when cross-validation is used to estimate a population parameter \( \Psi \) for the item selected by an MCP, rather than to estimate the optimal value of some critical parameter of an algorithm (as illustrated above). This type of bias can be demonstrated with a simple experiment. For each trial, we randomly generate fifty instances with a class variable and twenty predictor variables, each drawn from a uniform distribution of binary values. To determine whether the best predictor is statistically significant, we use \( k \)-fold cross-validation in an attempt to obtain an unbiased estimate of the score associated with the best predictor. In theory, this score can then be compared to a standard sampling distribution appropriate for a single score. The class and predictors are generated independently, so rejection of the null hypothesis is a Type I error.

Figure 6 shows how the probability of incorrectly rejecting the null hypothesis increases with \( k \), the number of cross-validation folds. As \( k \) increases, the sample \( S-S_1 \) approaches the size of the entire sample \( S \), and an MCP is more likely to select the same predictor as it would if the entire sample \( S \) were used. In the extreme case — leave-one-out cross-validation — an MCP applied to each sample \( S-S_1 \) will select the same predictor nearly every time, approximating the score that would be obtained if cross-validation were never used. Thus, cross-validation with large values of \( k \) should not be used to obtain unbiased scores for the items selected by MCPs, but instead should be applied to estimate the optimal value of some critical parameter, as illustrated by CART.

\(^7\) This parameter is distinct from the significance level \( \alpha \).
### 8.1.6 Randomization tests

Randomization tests are a third approach to adjusting for MCPs. Randomization tests, also known as permutation tests, generate an empirical sampling distribution which can be used to test hypotheses about \( X_{\text{max}} \) (Westfall and Young 1993; Jensen 1991, 1992; Edgington 1995; Noreen 1989; Cohen 1995a; Good 2000). To generate that distribution, randomization tests explicitly generate and test possible samples that could be drawn from the population if the null hypothesis were true. They estimate \( p(X_{\text{max}} \geq x \mid H_0) \) by estimating the probability density of all samples that would produce a score equal to or greater than \( x_{\text{max}} \). Randomization tests are closely related to Fisher's exact test (Section 8.1.2).

<table>
<thead>
<tr>
<th>Class Variable</th>
<th>Predictor Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A1</td>
</tr>
<tr>
<td>+</td>
<td>A</td>
</tr>
<tr>
<td>-</td>
<td>B</td>
</tr>
<tr>
<td>+</td>
<td>A</td>
</tr>
<tr>
<td>+</td>
<td>B</td>
</tr>
<tr>
<td>-</td>
<td>B</td>
</tr>
<tr>
<td>-</td>
<td>A</td>
</tr>
</tbody>
</table>

**Figure 7: A hypothetical data sample**

For example, consider the problem of estimating \( p(X_{\text{max}} \geq x \mid H_0) \) for the data shown in Figure 7, where each \( x_i \) corresponds to the classification accuracy of a single-feature rule (e.g., if \( A3 = \text{T} \), then \( \text{Class} = + \), else \( \text{Class} = - \)). By applying an MCP, we can determine the rule or rules with maximum classification accuracy \( x_{\text{max}} \). We can calculate \( p(X_{\text{max}} \geq x \mid H_0) \) by applying the same
MCP to data samples where the class variable is a permutation of the order of the actual class variable and the predictor variables are identical to the actual data. Under the null hypothesis, each of these data samples \( S^* \) is equally likely, thus the proportion of these randomized samples that result in scores \( x_{\text{max}}^* \) that equal or exceed \( x_{\text{max}} \) is equal to \( p(X_{\text{max}} \geq x | H_0) \). For the example data in Figure 7, the number of all unique randomized samples is:

\[
\binom{6}{3} = \frac{6!}{3!3!} = 20.
\]

Each of the class variables for the samples \( S^* \) is shown in Figure 8.

<table>
<thead>
<tr>
<th>Actual Class Variable</th>
<th>Randomized Class Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>+ + + + + + + + - - + + - - + + + - - - - + + - - - -</td>
</tr>
<tr>
<td>-</td>
<td>+ + + + + + + + - - + + + - - + + + - - - - + + + +</td>
</tr>
<tr>
<td>+</td>
<td>- - + - + - + + - - + + + + + + + + + + + + + +</td>
</tr>
<tr>
<td>-</td>
<td>- - - - + - - - - + + + + + + + + + + + + + +</td>
</tr>
</tbody>
</table>

Figure 8: Randomized class variables for Figure 7

For most data samples, the number of all unique randomized samples is extremely large. In a data sample in which there are \( c \) classes and \( N_i \) instances in each class \( i \), the number of unique randomized samples is:

\[
\frac{N!}{\prod_{i} N_i}
\]

For example, for a data sample with 30 instances and three uniformly-distributed values for the class variable, there are more than \( 5 \times 10^{12} \) unique randomized samples.

Fortunately, to estimate \( p(X_{\text{max}} \geq x | H_0) \) it is not necessary to consider every possible randomized sample \( S^* \). A randomization test can draw randomly from the space of all samples \( S^* \), and estimate \( p \) based on that random sample. Such tests are properly known as approximate randomization tests (Edgington 1995, Noreen 1989), but they are far more common than exact randomization tests, and so are commonly known by the abbreviated name. To date, randomization tests have been used only infrequently in knowledge discovery algorithms (Jensen 1992, 1991; Frank and Witten 1998; Oates and Jensen 1998).

Randomization tests produce accurate estimates of sampling distributions in a wide variety of situations. They can be used with nearly any measure of error because they do not require knowledge of the theoretical sampling distribution of a statistic. They avoid making the
independence assumption made in the Bonferroni and Sidak adjustments, nor they do not assume a specific correlation structure among the scores. Finally, the precision of their estimates of $p(X_{max} \geq x | H_0)$ can be increased merely by increasing the number of randomization trials.

The drawback of randomization tests is time complexity. Randomization tests typically introduce a constant factor of 100-1000 into knowledge discovery algorithms. They are also relatively poor at estimating very small p-values, because of the number of randomization trials required. However, most hypothesis tests require only that p-values be less than a relatively large value, typically between 0.10 and 0.01. Determining $p$ to this level of accuracy usually requires no more than 100-1000 trials.

8.1.7 Overfitting Reduction and Pruning

Many knowledge discovery algorithms explicitly correct for their tendency toward overfitting by using a two-stage process. First, the algorithm constructs an extremely large model. Second, the algorithm attempts to identify extraneous components in that model and eliminate them. The latter phase is generally called pruning. Algorithms that incorporate pruning include CART (Breiman et al. 1984), C4.5 (Quinlan 1993), CN2 (Clark and Niblett 1989), and Ripper (Cohen 1995b). Pruning can be implemented using any of the previously discussed approaches to adjusting for MCPs, including new data (Quinlan 1987), mathematical adjustments (Kass 1975), cross-validation (Breiman et al 1984), and randomization tests (Oates and Jensen 1998).

Many algorithms employ pruning because it can increase accuracy. Figure 9 shows the idealized behavior of error as model complexity increases (Breiman et al 1984; Weiss and Kulikowski 1991).

![Figure 9: The relationship between model complexity and error](image)

Initially, error declines sharply as model complexity increases. Error stabilizes as model complexity becomes sufficient for those aspects of the domain that the model can represent. Finally, error increases slightly as extraneous components are added to the model. Curves of roughly this shape have been observed for a variety of knowledge discovery algorithms.
The error curve is distinctly asymmetric with respect to model complexity. Overfitting slightly increases error, but underfitting by the same amount greatly increases error. Thus, if maximizing accuracy is the primary goal, algorithms with high statistical power (but a tendency to slightly overfit) are preferable to algorithms that construct models with high statistical significance (but may slightly underfit).

In recognition of this asymmetry, some knowledge discovery algorithms incorporate features that encourage slight overfitting. First, some algorithms use the two-phase construction and pruning process rather than attempting to limit the size of the model during the initial construction phase. By initially constructing a model that is likely to have extraneous components, designers hope that their algorithms will identify more useful components than algorithms that risk stopping too early. Second, some algorithms explicitly introduce a tendency toward overfitting into their pruning algorithms. For example, CART (Breiman et al. 1984) explicitly selects the value of a pruning parameter so it is slightly biased in favor of larger classification trees.

Why does overfitting decrease accuracy? Consider the effect of overfitting in the case of classification trees. Given a training sample of fixed size, adding components to a tree reduces the size of the subsample that arrive at a leaf node. Those subsamples are used to estimate parameters of the node (e.g., a class label or a probability distribution over possible class labels). The variance of estimates increases as the size of the subsample decreases. Small subsamples will result in poor estimates and will decrease the accuracy of the model.

However, overfitting does not invariably reduce accuracy. Error curves such as the one in Figure 9 may have a large region over which complexity can vary with little impact on error. This is particularly likely when the training sample is large in relation to the minimum sufficient model size. In these cases, models can contain many extraneous components without affecting accuracy.

Again, consider the case of a classification tree. In cases where error is measured with zero-one loss (e.g., percent correct or chi-square), the leaf parameters are class labels. C4.5 (Quinlan 1993), for example, estimates a class label based on the majority class of instances at a leaf. Class labels can be estimated correctly with relatively small samples, and increasing sample size does nothing to improve those estimates, particularly when the correct class represents an overwhelming majority of instances. As a result, moderate amounts of excess structure in the tree has little effect on accuracy. For example, Oates and Jensen (1997, 1998) show that classification trees constructed with C4.5 and pruned with several standard techniques have large amounts of unnecessary structure. In one experiment (Oates and Jensen 1997) with 19 data sets, a median value of 32% of the nodes in pruned trees were unnecessary, and in a quarter of the data sets over 50% of the nodes were unnecessary, yet eliminating these nodes had little or no impact on accuracy.

In some knowledge discovery tasks, overfitting avoidance can actually impair accuracy. If the training sample contains insufficient data to reach inflection point of an error curve such as the one in Figure 9, then any attempt to limit complexity can increase error. The shape of the error curve depends on the power of induction algorithm, the ability of models to represent knowledge in the domain, and the inherent variability of the data. Thus, overfitting avoidance can be seen as an a priori learning bias of a knowledge discovery algorithm (Schaffer 1993). It does not universally improve accuracy, and its effectiveness depends on how well it corresponds to specific aspects of the domain.
Citations


