BOOTSTRAPPING RULE INDUCTION TO ACHIEVE RULE STABILITY AND REDUCTION

Lemuel R. Waitman, Douglas H. Fisher*, Paul H. King
Department of Biomedical Engineering, Vanderbilt University
*Department of Electrical Engineering and Computer Science, Vanderbilt University

russ.waitman@vanderbilt.edu, douglas.h.fisher@vanderbilt.edu, paul.h.king@vanderbilt.edu

contact author: Doug Fisher (http://www.vuse.vanderbilt.edu/~dfisher/)

Abstract Most rule learning systems posit hard decision boundaries for continuous attributes and point estimates of rule accuracy, with no measures of variance, which may seem arbitrary to a domain expert. These hard boundaries/points change with small perturbations to the training data due to algorithm instability. Moreover, rule induction typically produces a large number of rules that must be filtered and interpreted by an analyst. This paper describes a method of combining rules over multiple bootstrap replications of rule induction so as to reduce the total number of rules presented to an analyst, to measure and increase the stability of the rule induction process, and to provide a measure of variance to continuous attribute decision boundaries and accuracy point estimates. A measure of similarity between rules is also introduced as a basis of multidimensional scaling to visualize rule similarity. The method was applied to perioperative data and to the UCI (University of California, Irvine) thyroid dataset.

Keywords: rule induction, rule stability, algorithm stability, bootstrapping, rule similarity, rule visualization, multidimensional scaling, perioperative medicine

Accepted to the Journal of Intelligent Information Systems, 2005.

1. Introduction

Rule induction (Michalski & Chilausky, 1980; Clark & Niblett, 1989; Cohen, 1995; Mitchell, 1997, pp. 274-306; van den Eijkel, 1999; Klösgen, 2002) identifies conditions that are associated with particular outcomes. For example in the domain of *perioperative medicine* (i.e., the process of preoperative evaluation, providing anesthesia, and managing postoperative recovery), 34,926 "outcomes" (generally adverse events) have been identified (Forrest, Rehder, Cahalan, & Goldsmith, 1990; 1992) and are used to evaluate post-surgical status of patients in cases that involve general anesthesia. Rule induction from data in this setting yields rules like the following:

```
IF Height <  158 AND Age <  49 AND ASAClass >= 3 THEN NauseaVomit = Significant
IF Hypertension = yes AND Phase1Recovery >= 84 AND Age <  49
         THEN Pain = Severe
IF Age >= 61 AND BloodLoss >= 100 AND BloodPressureVariability >= 16.7371
         THEN ExtendedPhase1Recovery = yes
```

The incidence of serious injury is very low and even cases with intraoperative incidents resolve with minimal postoperative complication. But these minimal incidents may result in added cost, delay, patient discomfort, or extended recovery time. Since there may be multiple outcomes/incidents per patient and most of the patients are "uneventful" with respect to any or even all adverse outcomes, perioperative KDD (knowledge discovery in databases) seeks to identify subpopulations at risk for the various adverse outcomes so that preoperative safeguards can be taken.

Rule induction fits this objective because induced rules focus on positive examples which "represent some surprising occurrence or anomaly we wish to monitor" (Riddle, Segal, & Etzioni, 1994). This is in contrast to the classification of positive and negative examples by classifiers (e.g., decision lists or decision trees), which classify all data,

typically with respect to mutually-exclusive outcomes. If the majority of the examples are negative, a classifier may be constructed to optimize overall classification at the expense of creating branches that isolate the abnormalities (Kubat, Holte, & Matwin, 1998).

Though rule induction represents a good starting point for analyzing perioperative data and like domains, current systems have drawbacks, which include:

a) No measure of variance around a rule's accuracy and no variance around continuous attribute decision boundaries (e.g., Height < 158), which can diminish a domain expert's trust in discovered knowledge. If a boundary for a rule involving weight is 67 kilograms, for example, it is valuable to know that the standard deviation is 3 kilograms instead of 10. It is also useful to know that one rule might have lower accuracy but a smaller standard deviation than another rule.

b) Various forms of rule *instability*, so that different rules are learned with small changes to a training data set (including changes to attribute decision boundaries as in (a)), which can again diminish trust in discovered rules. One of our goals is to introduce this notion of instability with respect to rule induction (as opposed to classifier learning), and to characterize it with respect to a selected rule induction system.

c) The discovery of a large number of rules, some of which can be quite similar and in any case may be difficult for a domain expert to filter.

This paper describes a method of combining rules over multiple bootstrap replications of rule induction so as to mitigate these problems. A measure of similarity between rules is also introduced as a basis of multidimensional scaling to visualize rule similarity.

Section 2 briefly describes the Brute system, which we use as the basic rule induction system. Section 3 describes the bootstrapping procedure as applied to rule induction, and processes of comparing, combining, and visualizing rules. Section 4 presents experimental results with the bootstrapped-Brute system, indicating substantial reduction in the number of discovered rules and providing variability information that is helpful in rule interpretation. Section 5 discusses other issues of algorithm stability and computational cost.

2. The Brute System for Rule Induction

Brute version 1.2 (Riddle, et al, 1994; Segal, 1997) was chosen to discover rules over which *summary rules* would eventually be induced across multiple bootstrap replications. For Brute, a rule antecedent is a conjunction of conditions defined over discrete and/or continuous attributes. A rule consequent is a discrete outcome. Sample rules include:

```
IF A=X and B=Y THEN C=Z

IF A<X AND A>=Y AND B=W THEN C=Z

IF A<>X AND B=Y THEN C=Z
```

The depth of the rule is the number of attribute conjuncts in its antecedent. The first example above has a depth of 2, the second a depth of 3 and the third a depth of 2. For discrete attributes, such as Sex=Female or AnesthesticAgent <> Isoflurane, the relational operator is either = or <>. For continuous attributes, the operator is either < or >=.

Theoretically, Brute can exhaustively search the space of conjunctive rules with a single specified outcome[1], but has a variety of options to limit the search in practice. Options include limiting the depth of search (i.e., the number of conjuncts that can occur in a rule's antecedent) and rejecting rules that cover less than a certain percentage of positive examples. Brute also includes other filters to eliminate rules from consideration:

(a) eliminate rules with antecedents that are very similar to another, better scoring rule or *redundant rule elimination*,

(b) eliminate rules with parents that are very similar to the parent[2] of another, better scoring rule or *alternate specialization elimination*,

(c) eliminate rules with antecedents that are subsumed by another good rule or *uninformative specialization elimination*, and

(d) eliminate rules that are not deemed statistically independent of another better scoring rule or *non-homogenous rule elimination*.

All filters are applied to choose between rules with the same outcome.

Of the rules discovered in the constrained search, Brute returns the best *N* rules according to an objective function, to be described shortly. Brute is executed by specifying options on how to perform the search, the data file to search, and the outcome to be predicted. An example would be:

```
brute -d2 -c10 -r5 mushroom POISONOUS
```

[1] If the data includes more than a single outcome then Brute can be run separately for each possible outcome. Each run would discover rules for the specified outcome, implicitly treating other outcomes as a separate class.

[2] A parent rule has an antecedent that is a subset of one less conjunct than any of its childrens' antecedents. For example, "Color=Blue" and "Shape=Round" are parents of "Color=Blue and Shape=Round".

In this example, a data file from the University of California at Irvine data repository (Bay, 1999) describing mushrooms has a binary class attribute of either POISONOUS or NONPOISONOUS and other attributes such as ODOR and GILL-COLOR. A search is conducted through the space of rules for the 5 best rules with depth less than or equal to 2 and that cover at least 10% of the positive training examples. After finding these rules in the training set, the performance of the rules is evaluated on a test set. Brute lists the rules found, their accuracy, coverage, and the result of a chi square test for statistical significance. These results are provided for the training set and a test set. The output for the example is shown in Table 1.

Table 1. Brute Results

	Data			Test		
	Acc	Cov	Chi	Acc	Cov	Chi
IF ODOR = FISHY THEN CLASS = POISONOUS	100.0	14.3	450.8	100.0	15.5	207.1
IF ODOR = FOUL THEN CLASS = POISONOUS	100.0	55.3	1741.1	100.0	54.4	727.0
IF ODOR = SPICY THEN CLASS = POISONOUS	100.0	14.4	454.3	100.0	15.2	203.7
IF GILL-COLOR = BUFF THEN CLASS = POISONOUS	100.0	43.8	1380.0	100.0	44.4	594.2
IF STALK-COLOR-ABOVE-RING = BUFF THEN CLASS = POISONOUS	100.0	10.5	332.3	100.0	12.0	160.9

For this example, the five rules were 100 percent accurate with very high statistical significance. *Coverage* refers to the percentage of training or test cases that satisfy a rule's antecedent side. *Accuracy* refers to the percentage of training or test cases that are covered by the rule and for which the consequent side is true. In the case of the Mushroom example, accuracy is the criterion used to select the best *N* (i.e., 5) rules, but Brute supports several objective functions for evaluating rule quality. We modified Brute to use *extended Laplace Accuracy*, which builds on simple accuracy and Laplace accuracy used by other learning systems (Clark & Niblett, 1991; Smyth & Goodman, 1991).

If n is the number of examples in the (test) data set for which the antecedent of the rule holds and e is the number of examples for which the consequent and the antecedent holds, then simple data accuracy of the rule is

$$A_D = \frac{e}{n} \ . \tag{1}$$

The problem with this measure is that it does not account for data coverage. A rule that predicts a single example correctly scores higher than a rule that covers 999/1000 examples. Laplace accuracy, which takes into account coverage, is calculated as

$$A_L = \frac{e+1}{n+2}. \tag{2}$$

This measure assumes *a priori* that the two possible outcome classifications are equally probable. The Laplace accuracy is not rational for rules below 50% accuracy because it assumes an *a priori* rule accuracy distribution of 50% (Segal, 1997). This can be corrected by using an extended-Laplace accuracy function (Good, 1965),

$$A_{LE} = \frac{e + k \times A_D}{n + k}, \tag{3}$$

where A_D is the proportion of positive examples in the data and k is a small integer, commonly set to 2 or the depth of search. When k is set to the rule induction search depth, Segal refers to this accuracy criterion as LaplaceDepth (Segal, 1997). This improved measure is now centered on the frequency of positive examples in the data instead of 50%. All of the clinical datasets in our studies had positive data proportions below 50%, typically 10% to 20%.

3. Bootstrapping Rule Induction and Summary Rule Extraction

Our work seeks to reduce the number of rules that need be examined by an analyst, to assign variance values to decision boundaries and point estimates, and to measure and improve the stability of the rule induction process. To achieve these goals, we repeatedly apply rule induction using Brute to different, but overlapping subsets of the available data, and abstract rules that occur across multiple rule-induction trials. We use bootstrapping as the basis of multiple rule induction trials, though forms of cross validation could be adapted to this purpose as well. In Section 5, we discuss alternative strategies for achieving several of the goals that we have outlined here.

3.1 Bootstrapping

The bootstrap (Efron, 1979; Efron & Tibshirani, 1993) is a computer-based method to estimate the standard error of a parameter. Bootstrap samples, also called replications, are created by uniformly sampling n times with replacement from a dataset of size n. Some instances in the original data set will appear zero times while others will appear multiple times. The bootstrap samples are used for training the classifier or rule induction algorithm. For large sample sizes, approximately 36.8 percent of the original samples will not be included in the bootstrap sample. These are reserved for testing. The bootstrap estimate, often referred to as the 0.632-bootstrap estimate, combines the accuracies from the testing and training sets as

$$acc^{.632} = \frac{1}{b}\sum_{i=1}^{b}(0.632 \cdot acc_i^{test} + 0.368 \cdot acc_i^{train}), \quad (4)$$

where b is the number of bootstrap samples, acc^{train} is the accuracy of the classifier or rule on the training data (i.e., the bootstrap sample) and acc^{test} is the accuracy on the test data (i.e., data not included in the bootstrap sample).

Bootstrap sampling underlies the machine learning method of *bagging* classifiers (Breiman, 1996), which is an acronym for "bootstrap aggregating". Breiman applied this technique to CART classification trees and nearest neighbor classifiers. Kohavi (1995) provides another example of applying bootstrap sampling to accuracy estimation for C4.5 decision trees and Naïve Bayes classifiers. Breiman conducted trials with between 10 and 100 bootstrap replications and Kohavi's experiments varied from 1 to 100 bootstrap replications. Breiman found that most of the improvement in bagging was gained with only 10 bootstrap replications. This is important for our study because rule induction using Brute requires significantly more computational time than a greedy method like decision tree construction.

Nonetheless, bootstrapping rule induction is different than bootstrapping a classifier, and while bagging experiments have provided us with some guidance on choosing a number of replications, the relevance of bagging and other methods of combining classifiers such as *boosting* (Freund & Schapire, 1996) are of limited relevance. Bootstrapping or bagging a classifier has the goal of increasing accuracy, but there is no equivalent overall accuracy criterion for bootstrapping rule induction.[3] Each rule independently classifies only a portion of the data. A rule found during one bootstrap replication might not appear in another replication. A rule with the same attributes might exist in other replications, but the attribute values (or value ranges) differ. Instead of maximizing overall classification accuracy, we seek to find rules whose basic form persists across multiple bootstrap replications. These rules, which we define shortly as *nearly-identical* rules, are combined into summary rules, which reflect

[3] Segal and Etzioni (1994) show how Brute can be extended to learn a classifier, called a decision list, from induced rules. Nonetheless, this extension is distinct from Brute, and we mention the decision list work here for the interested reader only.

variability in rule accuracy and antecedent conditions. Notably, domain experts expect variability, and hard rule boundaries seem arbitrary. It is also useful to know that one rule might have lower accuracy but a smaller standard deviation than another rule.

Bootstrapping is used also by Riddle and Fresnedo (1996). In their system, however, rules were induced once and then bootstrapped 1000-fold with the same data to determine the accuracy of the rule. This is not the same as bootstrapping the induction process. Very recently, Freidman and Popescu (2005) have described a process of creating rule ensembles (as opposed to classifier ensembles), by aggregating the "important" rules from decision tree classifier(s) through bagging. Evans and Fisher (1994, 2002) formed rule ensembles by aggregating the best classifying rules from multiple decision trees constructed by semi-automated induction, a collaboration of an expert and a learning program. In their industrial application, these best classifying rules were combined into a single quality control procedure.

Bootstrapping rule induction provides a means of evaluating the stability of the algorithm, as well as determining rule accuracy and variance. There has been considerable research into evaluating and bounding the stability of learning algorithms with respect to accuracy: how do changes in training sample influence differences in the accuracies of classifiers constructed from these varying samples (Breiman, 1996; Kearns & Ron, 1999; Bousquet & Elisseeff, 2002; Kutin & Niyogi, 2002; Evgeniou, Pontil, & Elisseeff, 2004; Elisseeff, Evgeniou, & Pontil , 2005). This work has not been concerned with "an algorithm's hypothesis itself, but the error of the algorithm's hypothesis" (Kearns & Ron, p. 1430). In contrast, Turney (1995) defines stability in terms of the *form* of learned classifiers. In Turney's view, unstable algorithms discover very different

"looking" classifiers, but with roughly the same accuracy over the input data distribution. This form of instability can cause experts to be skeptical of the rule induction process. Consistent with Turney's treatment, our goal is to define and evaluate stability in the form of discovered *rules*.

3.2 Summary Rule Generation

Once all bootstrap replications are complete, the best *N* (e.g., 50) rules, if that many are found, are compared to the rules from the remaining replications. For this study, the criterion for determining the best rules was the 0.632 bootstrap estimate (Equation 4) of the rule's extended-Laplace accuracy (Equation 3), and 10 bootstrap replications are performed. "Nearly identical" rules across the replications are identified. For a given rule, the rules from other replications must involve the same attributes with identical relational operators. For discrete attributes, the attribute value must be identical. For continuous attributes, the attribute values are allowed to vary. For example,

```
IF CPTCode = 29 AND Height < 164 AND HeartRateVariability >= 28.6
  THEN NauseaGreaterThanMild = yes
```

is nearly identical to

```
IF CPTCode = 29 AND Height < 158 AND HeartRateVariability >= 25.7
  THEN NauseaGreaterThanMild = yes
```

but is not nearly identical to

```
IF CPTCode = 29 AND Height >= 140 AND HeartRateVariability >= 21.3
  THEN NauseaGreaterThanMild = yes
```

and also not nearly identical to

```
IF CPTCode <> 27 AND Height < 164 AND HeartRateVariability >= 31.5
  THEN NauseaGreaterThanMild = yes
```

If nearly identical rules exist across multiple replications, a summary rule is created. The summary rule stores the mean and standard deviation of the summary rule's bootstrapped extended-Laplace accuracy and coverage (i.e., the percentage of cases that satisfy the rule's antecedent), and the number of replications that contained a nearly identical rule that support the summary rule. The summary rule also contains basic statistics regarding the variability of the continuous attributes included in the rule.

3.3 Retrieving Summary Rules and Continuous Attribute Range Filtering

After the summary rules are generated, they are retrieved for review. A minimum *level of support* is specified to limit the number of summary rules displayed. The level of support is the number of replications that contain a supporting *base* rule. Depending on the dataset and the number of rules found, the analyst may want, for example, to focus on rules that occur in all ten replications. A sample summary rule is shown below. The continuous attribute means are followed by the standard deviations in parentheses. The odds-ratio, which is the ratio of the rule's bootstrapped extended-Laplace accuracy and the prevalence of the outcome for the entire dataset, is also shown in parentheses on the following line, along with the bootstrapped extended-Laplace accuracy and coverage. The odds-ratio indicates how many times as likely the outcome occurs for the population which satisfies the rule antecedent, relative to the data as a whole.

```
SummaryRuleID: 7820, SourceRuleID: NVPreo0048005237

IF CPT = 29 AND HeartRateVariability >= 28.6 (4.5) AND Height <  164
(6.0)  THEN NauseaGreaterThanMild = yes

(2.87x as likely) Accuracy: 51.2 (10.9), Coverage: 5.9 (1.1), 10/10
```

The summary rules that contain the same continuous attribute twice were also statistically tested to further filter out inferior rules. For example, consider a rule that specifies a range for the Height variable by stating the upper and lower boundary:

```
IF Height >= 159 AND Height < 163
     AND Phase2Recovery < 27
THEN NauseaGreaterThanMild = Yes
```

In this example, Height is constrained to a fairly narrow range, which is unexpected. In this example, nearly identical rules occur in all 10 replications and the generated summary rule is:

```
SourceRuleID: NVPreo0048005115, SummaryRuleID: 7851
IF Height >= 158.6 (1.2) AND Height <  162.9 (0.7)
    AND Phase2Recovery <  29.4 (6.24)
THEN NauseaGreaterThanMild = Yes
(2.89x as likely) Accuracy: 51.7 (8.0), Coverage: 6.6 (2.6), 10/10
```

In this case, there may be concern that the distance between the two boundaries is small, but the small standard deviations indicate that the summary rule is significant. In this case, we would choose to retain the rule, but generally we wish to eliminate rules in which the specified range is meaningless due to a large standard deviation relative to the distance between the lower and upper boundaries.

A test for the difference between means (Strait, 1983) is applied. The null hypothesis is that distance between the means of two boundaries is negligible. A normal population distribution and equal variance, using sample variance, is assumed. Details of the test statistic are found in Appendix A.

3.4 Similarity Between Rules

After rule induction and summary rule generation, the analyst will want to compare the rules to determine if rule induction is discovering rules from different regions of the problem space or merely minor variations from a smaller portion of problem space. For example, all the rules related to the pain outcome might involve age and weight with few additional conjuncts. For nausea and vomiting, several rules might involve sex and weight while another group of rules consistently includes surgery time and anesthetic agent as attributes. The analyst may also compare rules that have different outcomes to see if they involve the same attributes. Prior related research includes the clustering of association rules (Lent, Swami, & Widom, 1997), measures of interestingness (Silberschatz & Tuzhilin, 1995), and defining rule distance (Gago & Bentos, 1998).

Interpreting rule similarity can be addressed using similarity measures, traditionally used in engineering cluster analysis, and multidimensional scaling, commonly used in psychology. Multidimensional scaling (MDS) is used for visualizing complex *N* dimensional problem spaces as 1, 2, or 3 dimensional graphs (Taylor, 1999). A classic example is visualizing cities on a map (Forrest & Harris, 1993). One approach would be to have the latitude and longitude coordinates of each city and plot the cities upon a two dimensional plane or more precisely on the surface of a sphere. The MDS approach assumes distances between each pair of cities are given, from which cities can be positioned in a two-dimensional space that approximates the map obtained by using absolute longitude and latitude coordinates. In general, MDS algorithms take a matrix of the distances or similarities between the items as input and attempt to find a configuration

in a low dimensional coordinate system that matches the order of the original distances as closely as possible (Johnson & Wichern, 1992).

Rules are different from most multivariate data because each rule is not a discrete point in the original multidimensional problem space. A rule represents a region of the problem space, but we are interested in measuring and visualizing similarities/differences in the *form* (i.e., *morphological* similarities/differences) of discovered rules, not the coverage of the rules with respect to the distribution of the data. The basic approach taken was inspired by the work of Gower (Gower, 1971; Everitt, 1993). Gower proposed a method for calculating similarity measures for variables of mixed type, both quantitative (continuous) and discrete:

$$s_{ij} = \frac{\sum_{k=1}^{p} w_{ijk} s_{ijk}}{\sum_{k=1}^{p} w_{ijk}}. \tag{5}$$

Everitt (1993) explains "In this formula,, s_{ijk} is the similarity between the *i*th and *j*th individuals as measured by the *k*th variable and w_{ijk} is typically 1 or 0 depending on whether or not the comparison is considered valid for the *k*th variable. Weights of zero are assigned when variable *k* is unknown for one or both individuals, or to binary values where it is required to exclude negative matches. For categorical data the component similarities, s_{ijk}, are 1.0 when the two individuals have the same value and 0.0 otherwise. For quantitative variables the similarity is measured by

$$s_{ijk} = 1 - \left| x_{ik} - x_{jk} \right| / R_k, \tag{6}$$

where x_{ik} and x_{jk} are the two individuals' values for variable *k*, and R_k is the range of the variable *k*, usually in the set of individuals to be clustered."

While Gower's method handles discrete and continuous variables, it must be modified because a continuous attribute reference is not a point (e.g., x_{ik} in (6) above), but a rule's antecedent defines a range of values. Thus, rule similarity is calculated using the overlap between two rules' ranges for continuous attributes and shared specification of discrete attributes. The method can be applied to either the initial rules discovered by Brute or the summary rules. For summary rules, the mean values of the continuous-attribute decision boundaries are used. Following Turney (1995), for purposes of morphological rule comparison, the attributes are assumed to be *independent* and *uniformly* distributed.

The general equations developed by Gower will be used to calculate the similarity measure but the similarity for each attribute, $s_{ijk,}$ can vary from –1 to 1 (in contrast to [0,1]). The weights, defined by Gower's method, are set to 1 when the attribute is to be included in the similarity calculation. A s_{ijk} is –1 if the respective ranges in two rules are maximally distant (given the observed range over all data of the attribute) and s_{ijk} is 1 if the ranges are identical.

For two rules x_i and x_j, let us examine one continuous attribute, k.

Let x_{ikmin} be the minimum value of k specified by the first rule.

Let x_{jkmin} be the minimum value of k specified by the second rule.

Let x_{ikmax} be the maximum value of k specified by the first rule.

Let x_{jkmax} be the maximum value of k specified by the second rule.

$R_{ik} = x_{ikmax} - x_{ikmin}$ is the range of k for the first rule

$R_{jk} = x_{jkmax} - x_{jkmin}$ is the range of k for the second rule

R_k is the same as in Gower's method, the range of k across all exemplars

Let $x_{jkmax} > x_{ikmax}$ so we can refer to the x_j rule as having a higher upper boundary for variable k when compared to x_i.

Using these definitions, continuous attribute similarity falls into three categories:

1. Dissimilar rules, with no overlap, $x_{ikmax} < x_{jkmin}$

$$s_{ijk} = \frac{x_{ik\max} - x_{jk\min}}{R_k - R_{ik} - R_{jk}}. \tag{7}$$

This will result in a negative number, which approaches –1 when the rules are at the opposite ends of the R_k. This measures the distance between the two rules divided by the maximal possible distance between the rules. For the example shown in Figure 1, the similarity would be

$$s_{ijk} = \frac{90-130}{(200-45)-(90-70)-(150-130)} = -0.35. \tag{8}$$

2. Similar rules, with overlap, $x_{ikmax} > x_{jkmin}$

$$s_{ijk} = \frac{x_{ik\max} - x_{jk\min}}{(R_{ik} + R_{jk})/2}. \tag{9}$$

This will result in a positive number that will equal 1 when the rules cover the same range of k. For the example shown in Figure 2, the similarity would be

$$s_{ijk} = \frac{105-100}{((105-85)+(120-100))/2} = 0.25. \tag{10}$$

3. Implicitly similar rules

For cases where one rule does not specify a value for the attribute, the attribute range for the unspecified rule is assumed to take on the entire range. Therefore, it is viewed as overlap between similar rules and the following equation is used

$$s_{ijk} = \frac{R_{ik}}{(R_{ik} + R_k)/2}, \tag{11}$$

where the attribute value is specified for rule *i* but not rule *j*. If only the lower rule, weight between 70 and 90 kilograms, in Figure 1 was specified the similarity would be

$$s_{ijk} = \frac{(90-70)}{((90-70)+(200-45))/2} = 0.23. \tag{12}$$

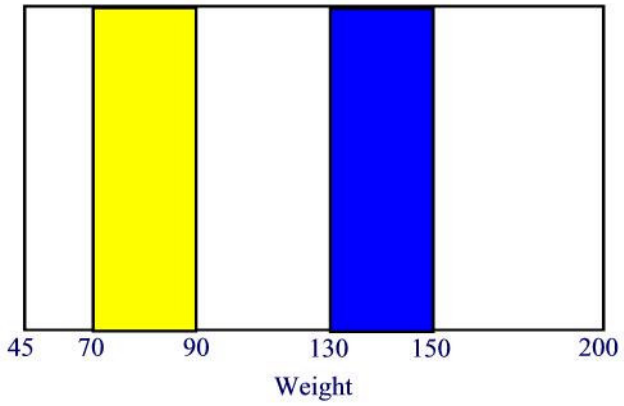

Figure 1. Two rules with no overlap

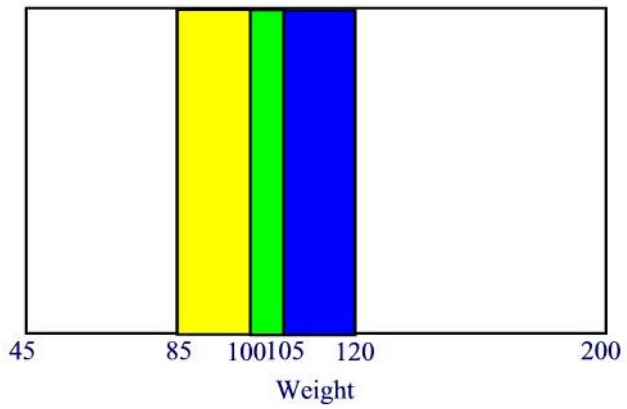

Figure 2. Two overlapping rules

Maximally dissimilar ranges with similarity of –1 are at the extreme opposite ends of the global range. Two rules which have identical range are maximally similar with a similarity of +1.

Similarity is also defined when one rule specifies an attribute (e.g., age) that the other rule does not reference. Consider the following rules and assume age ranges from 10 to 93 years:

```
A.  Weight < 70 THEN NauseaVomit = Significant

B.  Weight < 70 AND 10 < AGE < 15 THEN NauseaVomit = Significant

C.  Weight < 70 AND 10 < AGE < 50 THEN NauseaVomit = Significant
```

The similarity between A and C is greater than the similarity between A and B. This is because there is no restriction on age in rule A. Therefore, rule C's age range from 10 to 50 is closer to rule A's implicit range from 10 to 93, than rule B's range from 10 to 15.

Appendix B details the rules of determining similarity of discrete attributes.

3.5 Multidimensional Scaling (MDS)

Similarity measures are calculated between the summary rules generated during a rule induction session. Summary rules satisfying the minimum support criterion (see Section 3.3) are retrieved into an array and a matrix of similarity values is generated. This similarity matrix is used as input for multidimensional scaling.

Multidimensional scaling allows the analyst to visualize rules' similarity for a rule induction session (Borg & Groenen, 1997). One can see if the rules congregate in a few clusters or they are more uniformly distributed. Multidimensional scaling of the summary rule similarity matrices is accomplished with SPSS version 10.0. The PROXSCAL Version 1.0 procedure, developed at Leiden University by de Leeuw, Heiser, and Meulman (Busing, 1999), is used. Multidimensional scaling is an iterative process, and we used Torgerson scaling (Torgerson, 1958) for the initial configuration, and constrained the solution to two dimensions.

4. Experimental Studies

Our contention is that the refined knowledge provided by stable summary rules is more meaningful to analysts and clinicians than rules generated from a single induction session. This section describes the reduction in rule set size achieved through bootstrapping, and makes an initial examination of rule set stability. We begin by describing the data sets on which experiments will be performed.

4.1 Data Sets

We applied bootstrapped rule induction to data from the Vanderbilt Perioperative Information Management System (Higgins, et al, 1997), which is representative of data

(i.e., similar attributes) found in another large perioperative archive (Bothner, Georgieff, & Schwilk, 2000). The outcomes were *high intraoperative heart rate variability* (533 out of 3655 instances; 14.6%), *postoperative pain* (198 with no pain and 231 with severe pain out of 1583 instances; 12.5% and 14.6%, respectively), *postoperative nausea and vomiting* (494 greater than mild and 270 greater than moderate out of 2533 instances; 19.5% and 10.7%, respectively), and *long recovery time* (1615 out of 8248 instances; 19.6%). These outcomes were chosen because they are clinically meaningful for ambulatory patients, are reliably recorded in the perioperative database, and occur with moderate frequency. The perioperative attributes are shown in Table 2.

Hypothyroid data, used by Ross Quinlan and maintained at the University of California at Irvine's data repository (Bay, 1999), are also examined. The hypothyroid dataset was chosen because the combined training and testing sets contained 2642 instances after instances with missing values are removed and the nature of its attributes (a mixture of Boolean and numeric values). The primary hypothyroid classification (80 instances) and the compensated hypothyroid classification (136 instances) were merged into a single hypothyroid classification (8.2% of the total data) to more closely resemble the distribution of the perioperative datasets. Brute was used to induce rules for this single outcome of *all hypothyroid* in our experiments. The Thyroid attributes are shown in Table 3.

Table 2. Perioperative attributes from the Vanderbilt University datasets

Attribute Name	Range	Description
Sex	Male, Female	
Age	12 to 93	
OthHyper	no, yes	preoperative hypertension
HOMI	no, yes	history of myocardial infarction
Diabetes	no, yes	
ASAClass	1 to 4	preoperative assessment of anesthetic difficulty
SurgProcRelatedRisk	1 to 3	preoperative assessment of surgical difficulty
Height	137 to 208 cm	
Weight	9 to 171 kg	
BodyHab	1 to 5	measure of obesity
PreopSysBP	70 to 210 mmHg	Preoperative systolic blood pressure
PreopDiaBP	40 to 116 mmHg	Preoperative diastolic blood pressure
Pulse	40 to 154 bpm	Preoperative heart rate, beats per minute
PreopO2Sat	72 to 100%	Preoperative blood oxygen saturation
PreopECGAssess	abnormal, none, normal	Preoperative ECG (electrocardiogram) test results
PreopAnesSevereNV	no, yes	Preoperative history of severe nausea from anesthetic agents
KLevels	3to3.8, 3.8to4, 4to4.8, 4.8to5.1, 5.1to6, over6, na	Preoperative potassium test results
PreopBPMedCount	0 to 4	Count of patient's blood pressure medications
PreopAllergyAnaphylaxis	Anaphylaxis, na	Preoperative history of allergic reaction
PreopAllergyBronchospasm	Bronchospasm, na	Preoperative history of allergic reaction
PreopAllergyNausea	na, Nausea	Preoperative history of allergic reaction
CPT	00,10,11,12,13,15,16, 17,19,20,21,23,24,25, 26,27,28,29,30,31,35, 36,38,40,41,42,43,45, 46,47,49,50,51,52,53, 54,55,56,60,61,62,64, 69,92,na.	Current Procedural Terminology code group for patient's main surgical procedure. Actual codes are five digits. First 2 digits define general region.
Laproscopic	na, no, yes	Surgical procedure is laproscopic

Table 3. Thyroid attributes from the UCI Data Repository dataset

Attribute Name	Range
Outcome	hypo, negative
Age	1 to 94
Sex	M, F
On_thyroxine:	f, t
Query_on_throxine	f, t
On_antithyroid_medication	f, t
sick	f, t
thyroid_surgery	f, t
I131_treatment	f, t
query_hypothyroid	f, t
query_hyperthyroid	f, t
lithium	f, t
goitre	f, t
tumor	f, t
hypopituitary	f, t
psych	f, t
TSH	0.005 to 530
T31	0 to 11
TT4	2 to 430
T4U1	0.25 to 2.12
FTI1	2 to 395
referral_source	STMW, SVHC, SVHD, SVI, other

4.2 Rule Induction System Configuration

As previously stated, there is considerable latitude in the configuration of the Brute program. For this study, all data mining was done to a depth of three conjuncts and the minimum positive coverage was five percent. Iterative depth first search was used. The best 900 rules, as measured by extended-Laplace accuracy, were saved. Brute's four standard filters, as described in Section 2, were employed with default settings.

Ten bootstrap replications were made for each data set. After rule induction, the discovered rules were stored in a Microsoft SQL Server 7.0 relational database. Each

base rule was associated with a `RuleID`, and each summary rule was associated with a `SumRuleID`. Appendix C describes the database design.

4.3 Rule Reduction

After completing all 10 replications of a data set, the top 50 rules (of the 900 stored) for each replication were examined, in turn. For each top-50 rule of a replication, all 900 rules of each alternate replication were examined for nearly identical rules from which a summary rule could be constructed. If one or more nearly identical rules were found for a top-50 rule, then a summary rule was generated and stored, along with the base rules that support it. Thus, a rule ranked 32 in replication 4, together with a rule ranked 431 in replication 2, and rule 128 in replication 6, might all support a single summary rule with 3/10 support. Note that each summary rule formed in this way must have at least one top-50 rule in support. Reducing summary rule discovery costs was the primary motivation for the top-50 restriction.

By focusing on summary rules, the number of rules that needs to be analyzed is reduced significantly. For each dataset, Table 4 shows the average number of rules discovered per bootstrap replication[4], the number of summary rules[5], and the number of highly supported summary rules (occurring in exactly 8 of 10, exactly 9 of 10, and exactly 10 of 10 replications). Remember that Brute was configured to find up to 900 statistically significant rules. This limit was reached for the long recovery data set. On average, focusing on the summary rules reduces the number of rules to be analyzed by a

[4] Column 3 of Table 4 gives the total number of rules discovered without regard to the possibility of exact duplicates, which are unlikely, but possible. The average number of base rules per replication is given in parentheses.
[5] Column 4 of Table 4 lists all summary rules generated from base rules found in 2-10 replications.

factor of 4 and evaluating only fully supported rules (10/10 replications) reduces the number of rules by a factor of 20.

Table 4. Number of induced rules and the number of summary rules

Data Set	Sample Size	Average # Rules found per Replication.	# Summary Rules	# 8/10 Summary Rules	# 9/10 Summary Rules	# 10/10 Summary Rules
hypothyroid	2642	290	124	9	17	33
heart rate variability	3655	607	129	14	17	16
nausea over mild	2533	546	166	18	22	17
nausea over moderate	2533	617	169	19	13	16
severe pain	1583	646	169	14	19	15
no pain	1583	537	166	17	20	15
long recovery	8248	900	152	11	11	56

Table 5 shows the summary rule file for the `high heart rate variability` rules that persisted across all ten bootstrap replications. It is provided to give an idea of the kinds of rules that were found. All the rules include low preoperative pulse as an attribute. Most of the rules share another common attribute such as age, weight, or blood pressure.

Table 5. Summary Rules for high heart rate variability

```
 SourceRuleID: HRHigh0009001014, SummaryRuleID: 1445
Height <  177.7 (2.869) AND Age <  47.1 (10.999) AND Pulse <  58.9 (3.9)
THEN HRVarOver30 = yes
(4.398x as likely) Accuracy: 64.572 (4.3879), Coverage: 4.3287 (1.0141), 10/10


SourceRuleID: HRHigh0009001011, SummaryRuleID: 1450
Weight <  70 (3.756) AND Age <  50.6 (11.52) AND Pulse <  61.4 (4.115)
THEN HRVarOver30 = yes
(4.081x as likely) Accuracy: 59.926 (7.6929), Coverage: 3.8528 (1.0394), 10/10
```

Table 5 continued

SourceRuleID: HRHigh0009001010, SummaryRuleID: 1443

PreopSysBP < 115.5 (6.916) AND Age < 46.3 (9.719) AND Pulse < 61 (4.853) THEN HRVarOver30 = yes

(4.066x as likely) Accuracy: 59.707 (7.5339), Coverage: 4.9437 (1.2526), 10/10

SourceRuleID: HRHigh0009001022, SummaryRuleID: 1462

Height < 176.3 (2.214) AND PreopSysBP < 117.7 (9.742) AND Pulse < 58.9 (4.306) THEN HRVarOver30 = yes

(4.004x as likely) Accuracy: 58.790 (8.1828), Coverage: 4.0274 (0.4675), 10/10

SourceRuleID: HRHigh0009001003, SummaryRuleID: 1449

Age < 51.7 (8.994) AND Pulse < 60.6 (1.713) AND PreopDiaBP < 70.8 (1.932) THEN HRVarOver30 = yes

(3.888x as likely) Accuracy: 57.086 (8.9951), Coverage: 4.9846 (1.4420), 10/10

SourceRuleID: HRHigh0009001017, SummaryRuleID: 1453

Pulse < 60.4 (5.254) AND Weight < 73.4 (4.926) AND PreopO2Sat >= 97.2 (0.789) THEN HRVarOver30 = yes

(3.770x as likely) Accuracy: 55.354 (8.2066), Coverage: 4.63 (0.8047), 10/10

SourceRuleID: HRHigh0009001052, SummaryRuleID: 1471

IF Sex = Female AND Age < 45.2 (9.727) AND Pulse < 63.5 (3.629)

THEN HRVarOver30 = yes

(3.531x as likely) Accuracy: 51.841 (7.6053), Coverage: 4.5077 (1.5166), 10/10

SourceRuleID: HRHigh0009001015, SummaryRuleID: 1459

Weight < 73.75 (7.878) AND PreopSysBP < 115.8 (10.141) AND Pulse < 62.9 (4.067) THEN HRVarOver30 = yes

(3.373x as likely) Accuracy: 49.530 (7.6227), Coverage: 4.2196 (1.3793), 10/10

SourceRuleID: HRHigh0009001008, SummaryRuleID: 1458

Age < 48 (11.215) AND Pulse < 63.8 (3.327) AND PreopO2Sat < 97.6 (0.516) THEN HRVarOver30 = yes

(3.361x as likely) Accuracy: 49.356 (3.5509), Coverage: 4.0493 (0.9021), 10/10

SourceRuleID: HRHigh0009001083, SummaryRuleID: 1473

IF PreopECGAssess = none AND Weight < 68.65 (5.457) AND Pulse < 64.9 (5.507) THEN HRVarOver30 = yes

(3.354x as likely) Accuracy: 49.246 (6.3210), Coverage: 5.4851 (1.6223), 10/10

Table 5 continued

SourceRuleID: HRHigh0009001004, SummaryRuleID: 1452

Age < 52.5 (9.384) AND Pulse < 67.8 (6.106) AND SurgProcRelatedRisk >= 2 (0) THEN HRVarOver30 = yes

(3.039x as likely) Accuracy: 44.623 (5.0952), Coverage: 4.7620 (1.4132), 10/10

SourceRuleID: HRHigh0009001041, SummaryRuleID: 1489

Height >= 165.9 (2.601) AND Weight < 64.8 (4.917) AND Pulse < 67.1 (4.818) THEN HRVarOver30 = yes

(2.976x as likely) Accuracy: 43.691 (4.3287), Coverage: 4.184 (0.7397), 10/10

SourceRuleID: HRHigh0009001036, SummaryRuleID: 1465

Weight < 68.1 (7.82) AND PreopDiaBP < 67.9 (2.846) AND Pulse < 67.3 (3.683) THEN HRVarOver30 = yes

(2.700x as likely) Accuracy: 39.644 (5.5937), Coverage: 5.1311 (4.5090), 10/10

SourceRuleID: HRHigh0009001068, SummaryRuleID: 1476

Age < 41.3 (6.413) AND Weight >= 79.9 (12.476) AND Pulse < 66.8 (4.826) THEN HRVarOver30 = yes

(2.493x as likely) Accuracy: 36.603 (7.2994), Coverage: 5.0636 (2.9221), 10/10

SourceRuleID: HRHigh0009001198, SummaryRuleID: 1480

Age >= 41.3 (11.557) AND PreopSysBP < 125.7 (9.081) AND Pulse < 66.7 (9.105) THEN HRVarOver30 = yes

(2.013x as likely) Accuracy: 29.558 (8.2908), Coverage: 7.6374 (3.0432), 10/10

SourceRuleID: HRHigh0009001209, SummaryRuleID: 1482

Weight >= 67.8 (7.627) AND BodyHab < 3.5 (0.913) AND Pulse < 65.7 (2.312) THEN HRVarOver30 = yes

(1.720x as likely) Accuracy: 25.264 (3.6839), Coverage: 8.7442 (3.0675), 10/10

4.4 Multidimensional Scaling

Figure 5 shows the multidimensional scaling for the high heart rate variability summary rules shown in Table 5. Each rule in the space is labeled with the last three digits of the SourceRuleID.[6] Since all rules include pulse as a conjunct, rules that share two attributes are clustered together in the space. Rules 003, 004, 008, and 014 all involve low age.

[6] SourceRuleID is a candidate key for summary rules. It is conceptually synonymous with the primary key of SumRuleID. See Appendix C.

Rules 011, 015, 017, 036, and 041 all include low weight. Rules 068 and 209 include high weight and are positioned away from the group of rules involving low weight. Note that rules 010 and 068 contradict each other with respect to age and are positioned away from each other. The multidimensional space assists the analyst in identifying clusters of similar rules such as the low weight versus high weight groups. By referring to Table 5, one can see the low weight rules have tighter standard deviations for the weight and more powerful odds ratios than the rules specifying a high weight range.

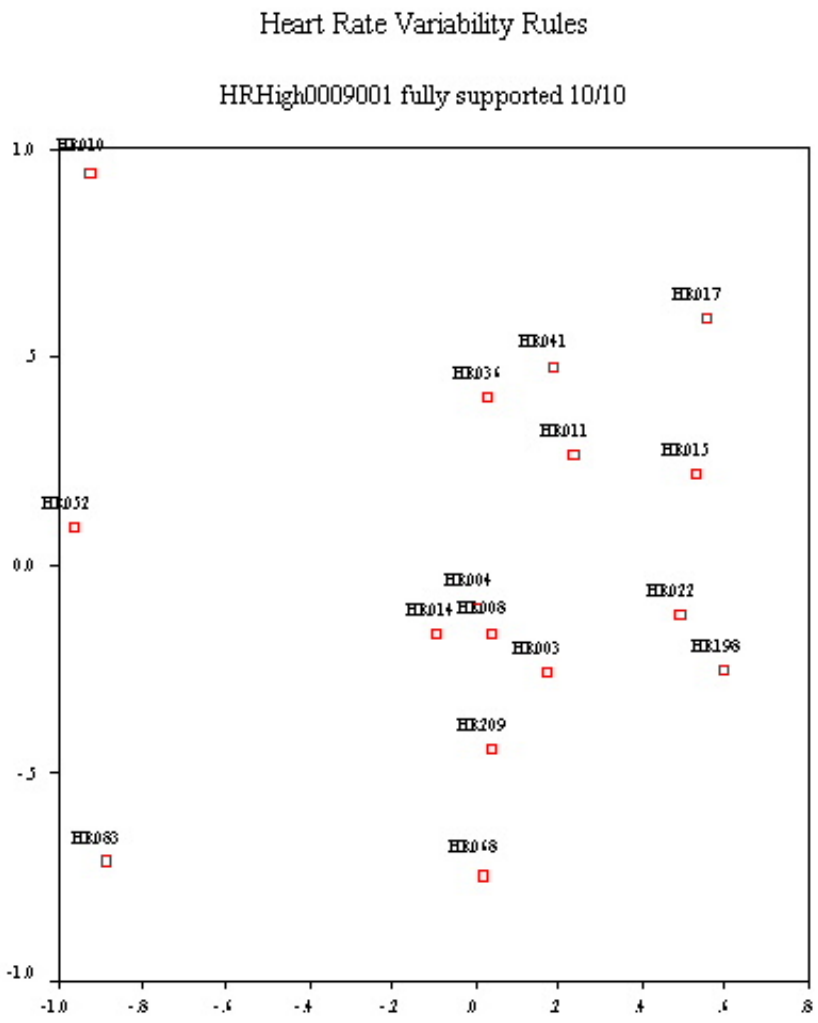

Figure 5. Multidimensional scaling for high heart rate variability summary rules

4.5 Rule Stability Experiments

Experiments were conducted to gain insight into the behavior of the summary rules as the size of the dataset increases. Our hypothesis is that larger datasets should result in more stable rules. Rule stability should be a function of the number of supporting bootstrap replications. Using the methodology above, we would expect that given a sufficiently large training set, 50 summary rules would be constructed and all would have 10/10 level of support.[7]

Other statistics also reflect stability. For instance, the number of rules rejected by continuous attribute filtering is a reflection of rule induction stability. As sample sizes increase, differences in continuous attribute bounds become significant and fewer bad rules are identified by continuous attribute filtering. For each sample size of Table 6, and at each level of bootstrap replication support (2 out of 10 to 10 out of 10), we recorded the (a) number of good rules, (b) number of bad rules rejected by filtering, and other statistics.

[7] This assumes that the underlying rule induction system exhibits stability in the limit.

Table 6. Sample sizes for experiments in rule stability (* reflects use of entire dataset)

Data Set Name	Sample sizes
hypothyroid	100, 200, 500, 1000, 2000, 2642*
no pain	100, 200, 500, 1000, 1583*
severe pain	100, 200, 500, 1000, 1583*
nausea over mild	100, 200, 500, 1000, 2000, 2533*
nausea over moderate	100, 200, 500, 1000, 2000, 2533*
heart rate variability long recovery	100, 200, 500, 1000, 2000, 3655* 100, 200, 500, 1000, 2000, 5000, 8248*

Figures 6 and 7 show the number of good and bad summary rules (rejected by filtering) at all levels of support as a function of sample size for the two largest data sets: `long recovery` and `heart rate variability`. Sample size is displayed on a logarithmic scale. As the sample size increases, the number of filtered rules decreases and the number of good rules stabilizes. There is an initial increase in the number of summary rules (both good and bad) as sample size increases, as a greater variety in the data set is introduced. With sufficient data, however, the number of summary rules decrease, good rules leveling out, and the number of bad rules approaching zero. The other domains exhibit similar peaking behavior, though training set size is not large enough to see a final tapering off of bad rules.

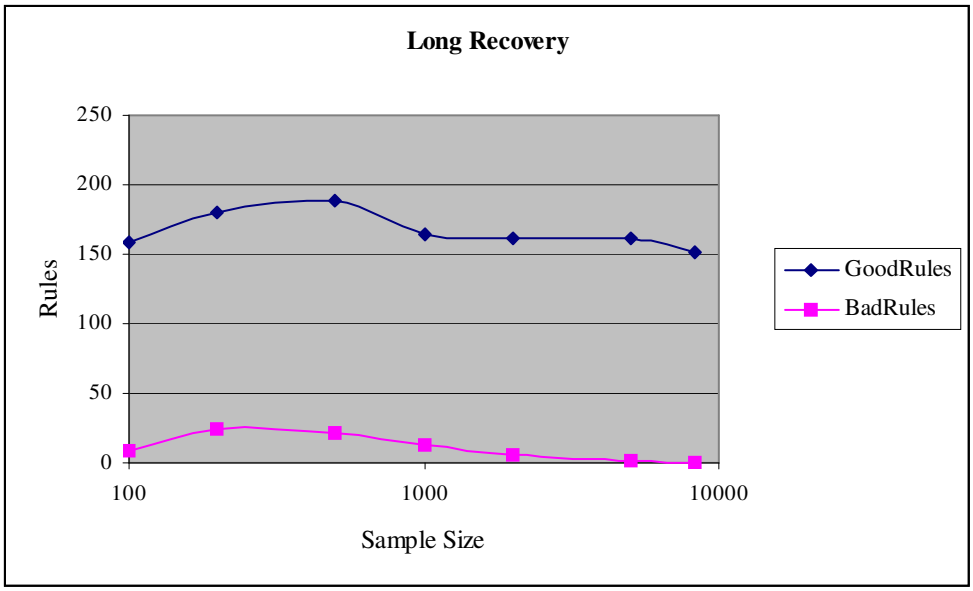

Figure 6: Rules discovered versus sample size for long recovery data set

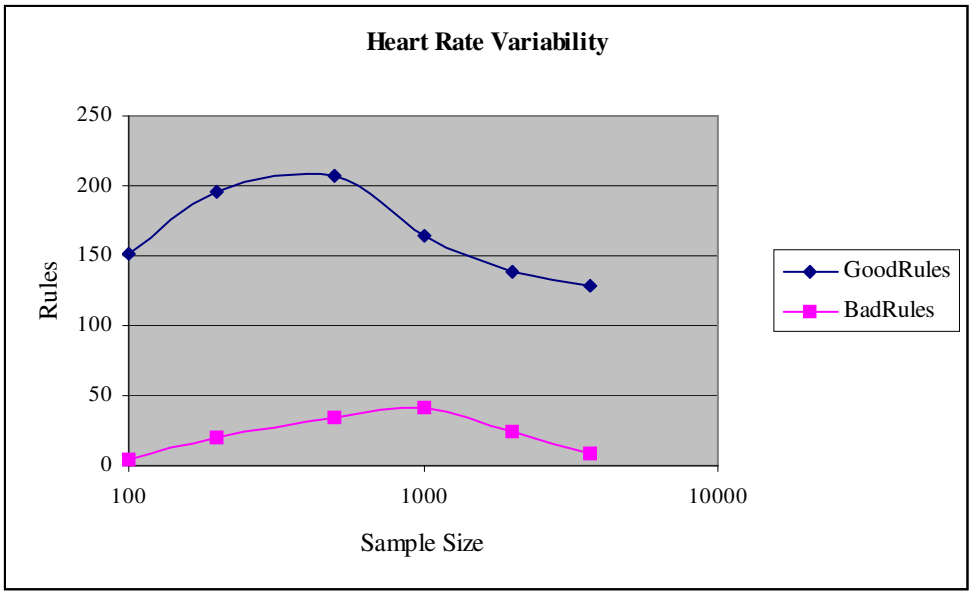

Figure 7. Rules discovered versus sample size for heart rate data set

5. Discussion

While the number of rules found is often a good metric for comparing the performance of different machine learning algorithms (Riddle, et al, 1994), creating hundreds or thousands of rules is unacceptable for a human analyst. Ordering summary rules by their level of supporting bootstrap replications combined with multidimensional scaling visualization of rule clusters provides a path for analysis. The analyst may start by looking at clusters of rules in the MDS graph of the summary rules that occurred in all ten replications or the rules with the highest accuracy or lowest variance. Once that knowledge is gleaned, attention can be shifted to summary rules with nine supporting replications, lower accuracy, or greater variance. Since the analyst is concerned with multiple risks, future work may use similarity measurement and multidimensional scaling to look for commonality between rules with different outcomes. Rules with pain as an outcome can be combined in a multidimensional space with long recovery time rules. A cluster that is composed of rules with dissimilar outcomes might suggest certain events are correlated for a specific subpopulation.

The disadvantage of the summary rule approach is increased computation time. All work was accomplished with identical Intel Pentium III 500 MHz computers with 256MB. One computer contained the database and carried the computational burden of running stored procedures, storing, and retrieving data. The other computer acted as an analysis workstation by executing the Brute algorithm and other client applications. Rule induction at a depth of three and rule storage for a single bootstrap replication took between ten seconds and three hours depending of the dataset. Preliminary experimentation with some datasets took over twelve hours to mine to a depth of four.

Since the process is repeated for ten bootstrap replications and experiments were conducted with multiple datasets, it was decided to limit depth of search to three. Calculating and retrieving summary rules took between one minute and two hours. Calculating the similarity matrices took between two and six seconds per rule comparison. For example, a matrix of seventy-seven rules took more than five hours. A matrix of forty-one rules took ninety minutes. From a practical standpoint, large matrices do not lend themselves to visual analysis so most MDS plots will be likely accomplished for matrices with fewer than fifty rules. Nevertheless it may be more efficient to implement the similarity calculation stored procedures outside the database and look at methods that do not require a complete matrix. SPSS typically took less than thirty seconds to compute and display the multidimensional scaling graph for the corresponding matrix.

Despite the computational cost of our approach, it is of practical significance in many domains. We limited rule depth to three for reasons of cost, but nonetheless, medical literature often focuses on statistical tests of single attributes (e.g., is a drug efficacious or not for a given study population, identification of a gene marker for a cancer), or logistic regression models, which do not have the precision of induced rules at describing subpopulations. Moreover, the attributes used are often the result of nontrivial lab tests, histories (e.g., history of nausea), and assessments (e.g., surgical risk), which take into account a variety of more primitive factors, are opaque to the rule induction process, and are standard medical practice. Thus, a rule-depth of three can be quite informative, and even with a rule-depth of three, rule discovery is beyond manual capabilities.

Generally, when compared to (a) the cost of collecting data, (b) the cost of wading through and cleaning large data stores, (c) the cost of expert analysis of large numbers of rules, (d) the lesser quality of results obtained by greedy algorithms, and (e) the cost of dealing with operative complications that rule induction may anticipate and mitigate, even a method that requires many hours or even days, can be highly cost effective in a medical setting. Importantly, *scaleup* is relative, and in the medical setting even an "expensive' algorithm may scale well, though in a real-time setting, it may not. This is not to say that efficiency enhancements should not be exploited where possible, and we discuss some alternative approaches that are undoubtedly more efficient than our bootstrapping approach shortly.

The experiments involving rule stability were observational because induced rule sets are complex when compared to a machine learning technique that results in a single classifier. The development of a theoretical foundation for induced rule stability and behavior is a worthy goal for future research. While the results are not definitive, we suggest the following observations are worth attention. (1) The reduction in filtered or bad rules seems an indication of stability. (2) Rule induction stability is reflected in an increased number of highly supported rules. (3) The observations that may be related to stability are also a reflection of the data. Stability is not achieved at the same rate or to the same degree for all data sets. (4) Finding stable rules requires a considerable sample size. It is questionable if rule induction is applicable to data sets with fewer than 1000 records. Our experiments would have benefited from the inclusion of larger datasets but the performance of Brute on data containing over 100,000 records has not been explored. Future research may find that moderately sized samples, perhaps between 1000 and

50,000 records, are sufficient for inducing stable rules. A method that characterizes data stability would allow the analyst to use a subset instead of all the data, thereby reducing computation time. While the underlying assumptions are different, the overall approach is the same as determining sample size for a desired power when conducting analysis of variance (Neter, Wasserman, & Kutner, 1990).

Using summary rule generation from multiple bootstrap trials for determining decision boundary and accuracy variance can be viewed as a rapid prototype that allowed us to adapt an off-the-shelf rule induction engine to the task of learning an extended and stable form of rule. Future work might develop more efficient methods for determining these variances within a single application of the rule induction engine. Such an engine would search the space of rules that have much the same form as summary rules, with variances attached to continuous attribute antecedents and accuracy point estimates. For example, attribute thresholds of a rule could be perturbed in small ways and checked against the data to arrive at such "summary" rules.[8] In contrast to this "model-driven" approach (i.e., where a rule is evaluated against data), Brute itself could be modified to perform much this same functionality in a more data-driven fashion. Brute uncovers many rules of the same form, which we chose to prune (see Section 2), but these rules of like form could be retained and Brute modified to combine them into "summary" rules within a single replication (rather than across replications).

6. Conclusion

Rule induction is well suited for problem domains with a multitude of risks and events. Conjunctive rules are easy to understand but the absolute boundaries of the rules,

[8] This model-driven approach was suggested by an anonymous reviewer.

algorithm instability with respect to these boundaries, and the sheer number of rules invite skepticism from domain experts. Summary rules display the variability in conjunct boundaries expected by the domain expert and reduce the number of rules which must be analyzed. Highly supported, stable summary rules give greater confidence that this knowledge is not just an artificial, over-fitted construct of the machine learning algorithm. Multidimensional scaling is a valuable tool for evaluating sets of induced rules but is more subjective. Future research should focus on advancing understanding of induced rule stability in larger datasets.

Acknowledgements

A very abbreviated version of sections 1-3.3 and 4.1-4.3 appear as Waitman, Fisher, and King (2003). We thank an anonymous reviewer for many helpful comments.

Appendix A

The test statistic used to filter summary rules (Section 3.3) is

$$T = \frac{\overline{X}_1 - \overline{X}_2 - d}{\sqrt{\frac{(n_1-1)S_1^2 + (n_2-1)S_2^2}{n_1+n_2-2}} \cdot \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \tag{13}$$

X_1 is the mean for the upper boundary and X_2 is the mean for the lower boundary, d is the minimum acceptable distance between the two boundaries. S_1 and S_2 are their standard deviations. The number of instances is the same for both boundaries, $n_1 = n_2$ so this equation reduces to

$$T = \frac{\overline{X}_1 - \overline{X}_2 - d}{\sqrt{\frac{(n-1)S_1^2 + (n-1)S_2^2}{2n-2}} \cdot \sqrt{\frac{2}{n}}} = \frac{\overline{X}_1 - \overline{X}_2 - d}{\sqrt{\frac{S_1^2 + S_2^2}{n}}} \tag{14}$$

Since this calculation is applied to all summary rules with two conjuncts (upper and lower bounds) of the same attribute, some assumptions are made to reduce the comparison of the test statistic to a constant value so it could be included in a stored procedure. Distance was set to 1 and it was assumed that analysts would be interested in summary rules with at least 7 supporting rules. The test statistic must exceed 2.18 in order to reject the null hypothesis for $n = 7$ and for $\alpha = 0.05$ ($|t| > t_{\alpha/2,2n-2}$). If the test statistic was less than 2.18, the summary rule was not retrieved for analysis. In the example summary rule of Section 3.3, the test statistic is 7.52 so the null hypothesis is rejected and the summary rule is retained. An example of a summary rule that is rejected is:

```
IF Height < 165.5 (9.264) AND HeartRateVariability >= 31.2 (9.4) AND
HeartRateVariability < 35.0 (10.8) THEN NauseaGreaterThanMild =
significantNV

(2.08x as likely) Accuracy: 37.2 (9.5), Coverage: 4.1 (1.1), 10/10
```

The test statistic for this summary rule is 0.62 so the null hypothesis is accepted and the rule is not displayed for analysis.

Appendix B

Similarity Calculation for Discrete Attributes

Discrete attribute similarity does not involve ranges and overlap but instead focuses on the number of discrete attribute values in common between both rules. In the continuous attribute similarity measure, the numerator was the overlap or distance between the ranges of each rule and the denominator contained the ranges of each rule individually. For discrete attributes, the numerator contains the percentage of the

attributes shared by both rules. The denominator contains the average of the attributes covered by the two rules individually. There are seven possible categories for discrete rule similarity. Each category will be followed by an example using the attribute anesthetic agent. The possible anesthetic agents are: Desflurane, Isoflurane, Lidocaine, Propofol, Sevoflurane, or Unspecified.

1. If only one of the rules specifies an attribute and the operator is "equal to" then similarity is

$$s_{ijk} = \frac{1/N}{(1/N+1)/2} = \frac{2}{1+N} \tag{15}$$

Where N is the number of possible values for the discrete attribute.

```
Rule 1: Agent =Desflurane  rule 2:                    Similarity: 2/7
```

2. If only one of the rules specifies an attribute and the operator is "not equal to" then the similarity is

$$s_{ijk} = \frac{N-1/N}{(((N-1)/N)+1)/2} = \frac{2N-2}{2N-1} \tag{16}$$

```
Rule 1: Agent <> Isoflurane  Rule 2:                  Similarity: 10/11
```

This generalizes when the one rule specifies an attribute as "not equal to" multiple values. If there are M "not equal to" expressions for the attribute, then the similarity is

$$s_{ijk} = \frac{N-M/N}{((N-M)/N+1)/2} = \frac{2N-2M}{2N-M} \tag{17}$$

```
Rule 1: Agent <> Isoflurane AND Agent <> Desflurane  Rule 2:
                                                      Similarity: 4/5
```

3. If $x_{ik} = x_{jk}$, the attribute values are identical, and the operators for each attribute are the same, then $s_{ijk} = 1$.

```
Rule 1: Agent = Isoflurane  Rule 2: Agent = Isoflurane     Similarity: 1

Rule 1: Agent <> Isoflurane  Rule 2: Agent <> Isoflurane   Similarity: 1
```

4. If $x_{ik} = x_{jk}$, the attribute values are identical, but the operators for each attribute not equal, then $s_{ijk} = -1$.

```
Rule 1: Agent = Isoflurane  Rule 2: Agent <> Isoflurane   Similarity: –1
```

5. If $x_{ik} <> x_{jk}$, the attribute values are different, but the operators are "equal to", then $s_{ijk} = -1$.

```
Rule 1: Agent = Isoflurane  Rule 2: Agent = Desflurane    Similarity: –1
```

6. If one of the rules specifies "equal to" an attribute value and the other rule specifies "not equal to" a different attribute value (Ex. Rule 1: Agent = Des and Rule 2: Agent <>Iso) then the similarity is

$$s_{ijk} = \frac{1/N}{(1/N + \frac{N-1}{N})/2} = \frac{2}{N} \tag{18}$$

```
Rule 1: Agent = Isoflurane  Rule 2: Agent <> Desflurane  Similarity: 1/3
```

This is expanded when the "not equal to" operator is used more than once in one of the rules. If specified M times, then the similarity is

$$s_{ijk} = \frac{1/N}{(1/N + \frac{N-M}{N})/2} = \frac{2}{1+N-M} \tag{19}$$

```
Rule 1: Agent = Isoflurane

Rule 2: Agent <> Desflurane AND Agent <> Sevoflurane     Similarity: 2/5
```

Note that (15) is a special case of (19), where $M = 0$.

7. Finally, when both rules specify <> to the same attribute but they specify multiple attribute values, the similarity is based on the portion of the ranges shared by the rules minus the portion of the ranges which are not shared.

$$s_{ijk} = \frac{(N - (M_1 \bigcup M_2))/N - (M_1 \oplus M_2)/N}{(\frac{N - M_1}{N} + \frac{N - M_2}{N})/2} \tag{20}$$

$$s_{ijk} = \frac{(N - (M_1 \bigcup M_2))/N - ((M_1 \cup M_2) - (M_1 \cap M_2))/N}{(\frac{N - M_1}{N} + \frac{N - M_2}{N})/2} = \frac{2(N - 2(M_1 \bigcup M_2) + (M_1 \cap M_2))}{2N - M_1 - M_2} \tag{21}$$

```
Rule 1: Agent <> Isoflurane AND Agent <> Desflurane

Rule 2: Agent <> Desflurane AND Agent <> Sevoflurane     Similarity: 1/4
```

This measure results to a minimum score of –2 instead of –1. A score of -2 occurs when all of an attribute's values appear in a <> expression of one rule, but no value appears in <> expressions of both rules (i.e., $M_1 + M_2 = N$). Essentially, this measure double counts the number of unshared ranges. Rather than correcting for this exactly, an adequate approximation is to divide by 2 when the score is negative, thus yielding -1 as a minimum. Note that (17) is a special case of (21), where either M_1 or M_2 is 0.

Once the similarity measures for all continuous and discrete attributes are calculated, they are combined using Gower's method (Equation 5). The similarity measure calculation is implemented as a stored procedure in the KDD database.

Appendix C

Choosing to store the induced rules in a database was critical to facilitating this research. With the induced knowledge as data, the analyst is free to apply set based techniques,

statistical tests, and evaluative methods at any time after induction using standard SQL techniques. It will also be easier to integrate discovered rules into the perioperative management system so the clinical decision makers can act on the knowledge. Clinicians can be alerted to risks ahead of time by displaying relevant rules in the perioperative management system based upon the current patient's preoperative and intraoperative findings. Figure 8 provides an overview of the rule induction methodology.

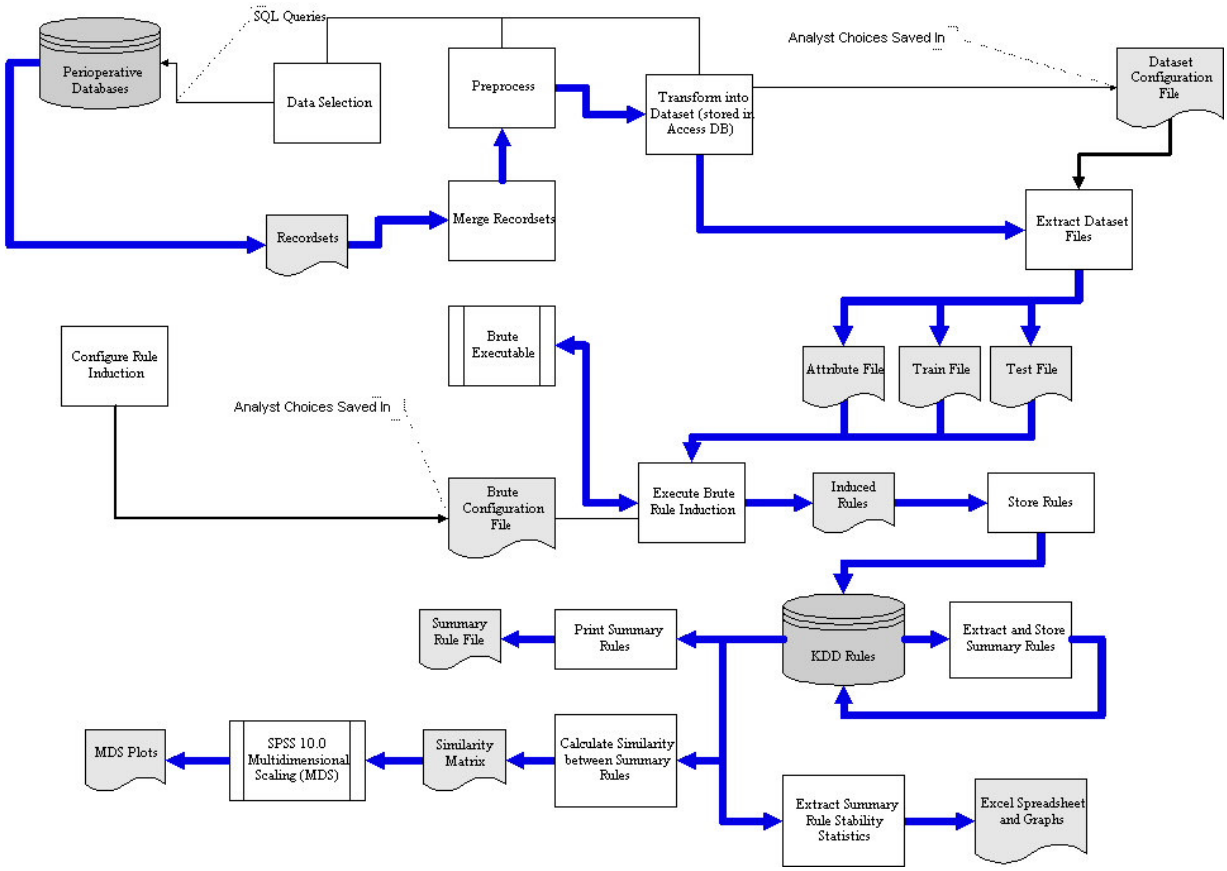

Figure 8. Rule Induction Process

The Figure 9 below shows the entity relationship diagram for storing rules. The CommandLine table stores the information associated with a rule induction session on a particular data set. The data set is identified by the database name and the query name.

Executing Brute rule induction on a bootstrap replication of the data set is referred to as a ValidationTrial in the KDD database.

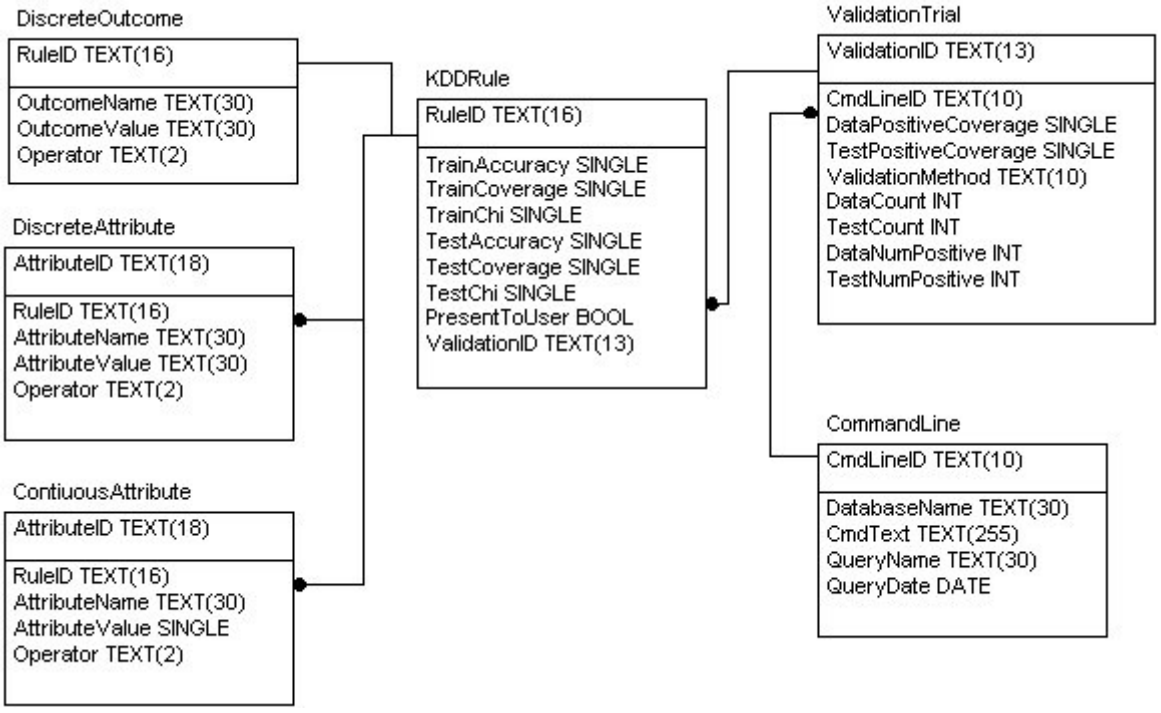

Figure 9. Entity Relationship Diagram for rule storage in the KDD database

Figure 10 shows the entity-relationship diagram concerned with storing summary rules. The SummaryRule table stores the mean and standard deviation of each summary rule's bootstrap Laplace accuracy and coverage, as well as the SumRuleID, which is the primary key for this table. The SourceRuleID is a candidate key for the SummaryRule table[9], and effectively points at one of the base rules from which the summary rule was generated (via a Foreign Key constraint through the DiscreteOutcome table). Rules from other replications from which the summary rule was generated are stored in the

[9] In Section 4.3 on Multidimensional scaling, we actually use the candidate key SourceRuleID in labeling the points, instead of RuleID. There is no conceptual importance to this.

SummaryRuleSupport table. The SummaryAttribute table is used to store basic statistics regarding the variability of the continuous attributes included in the rule.

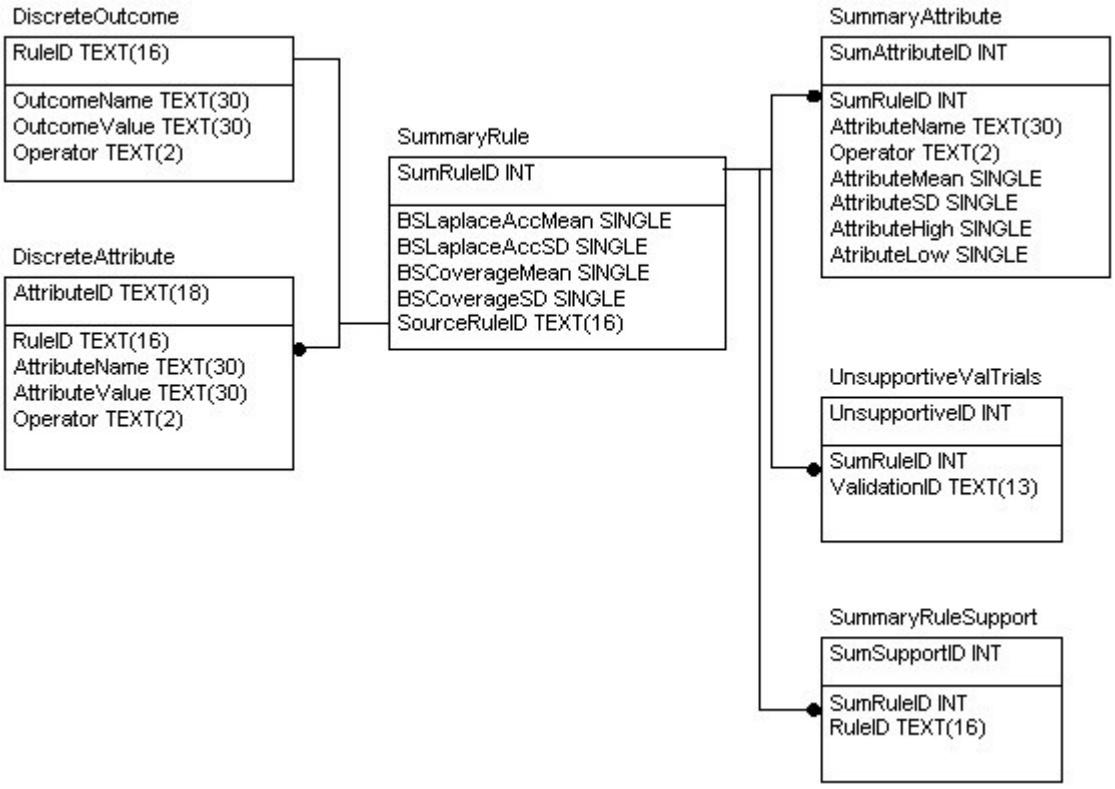

Figure 10. Entity Relationship Diagram for summary rule storage in the KDD database

References

Dorland's Illustrated Medical Dictionary. 26th ed. (1985). W.B. Saunders Company, Philadelphia, PA.

Bay, S. D. (1999). *The UCI KDD Archive. http://kdd.ics.uci.edu.* University of California, Department of Information and Computer Science, Irvine, CA.

Borg, I., & Groenen, P. J. F. (1997). *Modern multidimensional scaling: theory and applications.* Springer, New York.

Bothner, U., Georgieff, M., & Schwilk, B. (2000). Building a large-scale perioperative anaesthesia outcome-tracking database: methodology, implementation, and experiences from one provider within the German quality project. *British Journal of Anaesthesia 85*: 271-280.

Bousquet, O., & Elisseeff, A. (2002). Stability and generalization. *Journal of Machine Learning Research, 2*: 499-526.

Breiman, L. (1996). Bagging predictors. *Machine Learning, 24*: 123-140.

Breiman, L. (1996). Heuristics of instability and stabilization in model selection. *Annals of Statistics, 24*: 2350-2383.

Busing, F. (2001). *PROXSCAL. http://www.fsw.leidenuniv.nl/www/w3_ment/ medewerkers/busing/PROXSCAL.HTM*. Leiden University, Leiden, The Netherlands.

Clark, P., & Niblett, T. (1989). The CN2 induction algorithm. *Machine Learning, 3*: 261-283.

Clark, P., & Boswell, R. (1991). Rule induction with CN2: Some recent improvements. In *Machine Learning - Proceedings of the Fifth European Conference (EWSL-91)*, Springer-Verlag, Berlin: 151-163.

Cohen, W. (1995). Fast effective rule induction. In *Proceedings of the Twelfth International Conference on Machine Learning*. Morgan Kaufmann, San Mateo, CA: 115-123.

Efron, B. (1979). Bootstrap methods: Another look at the jackknife. *The Annals of Statistics, 7*: 1-26.

Efron, B., & Tibshirani, R. J. (1993). *An Introduction to the Bootstrap*. Chapman and Hall, New York, NY.

Elisseeff, A., Evgeniou, T., & Pontil, M. (2005). Stability of randomized learning algorithms. *Journal of Machine Learning Research, 6*: 55-79.

Evans, B., & Fisher, D. (1994). Overcoming process delays with decision tree induction. *IEEE Expert, 9*: 60-66.

Evans, B., & Fisher, D. (2002). Using decision tree induction to minimize process delays in the printing industry. In *Handbook of Data Mining and Knowledge Discovery*. Oxford University Press, Oxford, UK: 874-881.

Everitt, B. (1993). *Cluster analysis. 3rd ed*. E. Arnold, London.

Evgeniou, T., Pontil, M., & Elisseeff, A. (2004) Leave-one-out error, stability, and generalization of voting combinations of classifiers. *Machine Learning, 55*: 71-97.

Fayyad, U. M., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery: An overview. In *Advances in Knowledge Discovery and Data Mining*. AAAI Press/The MIT Press, Menlo Park, CA: 1-34.

Forrest, J. B., Cahalan, M. K., Rehder, K., & Goldsmith, C. H. (1990). Multicenter study of general anesthesia. II. Results. *Anesthesiology, 72*: 262-268.

Forrest, J. B., Rehder, K., Cahalan, M.K., & Goldsmith, C. H. (1992). Multicenter study of general anesthesia III. Predictors of severe perioperative adverse outcomes. *Anesthesiology, 76*: 3-15.

Forrest, W. Y., & Harris, D. F. (1993). Multidimensional scaling. In *SPSS For Windows: Professional Statistics, Release 6.0*. SPSS Inc., Chicago, IL: 155-222.

Freidman, J. H., & Popescu, B. E. (2005). *Predictive Learning via Rule Ensembles.* Stanford University, Department of Statistics. http://www.stat.stanford.edu/people/faculty/friedman/index.html

Freund, Y., & Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning*. Morgan Kaufmann, San Mateo, CA: 148-156.

Gago, P., & Bentos, C. (1998). A metric for selection of the most promising rules. In *Proceedings of the Second European Conference on the Principles of Data Mining and Knowledge Discovery (PKDD'98)*, Nantes, France: 19-27.

Good, I. J. (1965). *The Estimation of Probabilities: An Essay on Modern Bayesian Methods. Research Monograph 30*. MIT Press, Cambridge, MA.

Gower, J. C. (1971). A general coefficient of similarity and some of its properties. *Biometrics, 27*: 857-872.

Higgins, M. S., Beattie, C., St. Jaques, P., Patel, N., Yarborough, J., & Tsai, Y .S. (1997). The Vanderbilt Perioperative Information Management System Forms Foundation for Outcomes Management. In *AMIA Annual Fall Symposium*, Nashville, TN: 948.

Johnson, R. A., & Wichern, D. W. (1992). Clustering. In *Applied Multivariate Statistical Analysis. 3rd ed.* Prentice Hall, Inc., Upper Saddle River, NJ: 573-609.

Kearns, M., & Ron, D. (1999). Algorithmic stability and sanity check bounds for leave-one-out cross validation bounds. *Neural Computation, 11*: 1427-1453.

Kitz, R. J., & Vandam, L. D. (1990). Scope of modern anesthetic practice. In *Anesthesia. 3rd ed.,* Churchill Livingstone Inc., New York, NY: 3-23.

Klösgen, W (2002). Decision rules. In *Handbook of Data Mining and Knowledge Discovery*. Oxford University Press. Oxford, UK: 277-282.

Kohavi, R. (1995). *A study of cross-validation and bootstrap for accuracy estimation and model selection.* Morgan Kaufmann, San Mateo, CA: 1137-1145.

Kubat, M., Holte, R. C., & Matwin, S. (1998). Machine learning for the detection of oil spills in satellite radar images. *Machine Learning, 30*: 195-215.

Kutin, S., & Niyogi, P. (2002). Almost-everywhere algorithmic stability and generalization error. In *Proceedings of the 18th Conference on Uncertainty in AI*. Morgan Kaufmann, San Francisco, CA: 275-282.

Lent, B., Swami, A., & Widom, J. (1997). Clustering association rules. In *Proceedings of the Thirteenth International Conference on Data Engineering (IDCE '97)*, IEEE Computer Society Press, Birmingham, England: 220-231.

Michalski, R. S., & Chilausky, R. (1980). Learning by being told and learning from examples: An experimental comparison of the two methods of knowledge acquisition in the context of developing an expert system for soybean disease diagnosis. *International Journal of Policy Analysis and Information Systems, 4*: 125-160.

Mitchell, T. M. (1997). *Machine Learning*. McGraw Hill, New York, NY.

Neter, J., Wasserman, W., & Kutner, M. H. (1990). *Applied Linear Statistical Models. Third ed.* Richard D. Irwin, Inc, Homewood, IL.

Owens, D. K., & Sox, H. C. (1990). Medical decision making: probabilistic medical reasoning. In *Medical informatics: computer applications in health care*. Addison-Wesley Publishing Company, Inc., Reading, MA: 93.

Riddle, P., Segal, R., & Etzioni, O. (1994). Representation design and brute-force induction in a Boeing manufacturing domain. *Applied Artificial Intelligence, 8*: 125-147.

Riddle, P., & Fresnedo, R. N. D. (1996). Framework for a generic knowledge discovery toolkit. In *Learning From Data: AI and Statistics V*. Springer-Verlag, Berlin: 341-353.

Segal, R. B. (1997). *Machine Learning as Massive Search.* PhD Dissertation. University of Washington, Department of Computer Science, Seattle, WA.

Segal, R., & Etzioni, O. (1994). Learning decision lists using homogeneous rules. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94),* AAAI Press/MIT Press, Menlo Park, CA: 619-625.

Silberschatz, A., Tuzhilin, A. (1995). On subjective measures of interestingness in knowledge discovery. In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD-95),* AAAI Press, Menlo Park, CA: 275-281.

Smyth, P., & Goodman, R. M. (1991). Rule induction using information theory. In *Knowledge Discovery in Databases*. MIT Press, Cambridge, MA: 159-176.

Strait, P. T. (1983). Some standard tests of statistical hypotheses. In *A First Course in Probability and Statistics with Applications*. Harcourt Brace Jovanovich, Inc, New York, NY: 410-422.

Taylor, P. (1999). Statistical methods. In *Intelligent Data Analysis: an Introduction*. Springer- Verlag, Berlin: 113-118.

Torgerson, W. S. (1958). *Theory and Methods of Scaling*, John Wiley and Sons, New York.

Turney, P. (1995). Bias and the quantification of stability. *Machine Learning, 20*: 23-33.

Van Den Eijkel, G.C. (1999). Rule induction. In *Intelligent Data Analysis: An Introduction*. Springer-Verlag, Berlin: 195-216.

Waitman, L. R., Fisher, D., & King, P. (2003). Bootstrapping rule induction. In *Proceedings of the IEEE International Conference on Data Mining*, IEEE Computer Society, Los Alamitos, CA: 677-680.

Zar, J. H. (1999). *Biostatistical Analysis*. Prentice-Hall, Inc., Upper Saddle River, NJ.