ISTITUTO DI ANALISI DEI SISTEMI ED INFORMATICA
CONSIGLIO NAZIONALE DELLE RICERCHE

G. Felici, K. Truemper, F.-S. Sun

A METHOD FOR CONTROLLING ERRORS IN TWO-CLASS CLASSIFICATIONS

R. 480 Novembre 1998

Giovanni Felici – Istituto di Analisi dei Sistemi ed Informatica del CNR, viale Manzoni 30 - 00185 Roma, Italy. Email : felici@iasi.rm.cnr.it.

Fu-Shin Sun – University of Texas at Dallas Computer Science Program, Box 830688, Richardson, Texas 75083-0688, U.S.A, e-mail: shing@utdallas.edu.

Klaus Truemper – University of Texas at Dallas Computer Science Program, Box 830688, Richardson, Texas 75083-0688, U.S.A, e-mail: truemper@utdallas.edu.

This research was supported in part by the Office of Naval Research under Grant N00014-93-1-0096.

Istituto di Analisi dei Sistemi ed Informatica, CNR
viale Manzoni 30
00185 ROMA, Italy

tel. ++39-06-77161
fax ++39-06-7716461
email: iasi@iasi.rm.cnr.it
URL: http://www.iasi.rm.cnr.it

Abstract

Two types of errors can be associated with classifying data into two categories $\mathcal{A}$ and $\mathcal{B}$: misclassifying an item in $\mathcal{A}$ as one of $\mathcal{B}$ and, conversely, misclassifying an item in $\mathcal{B}$ as one of $\mathcal{A}$. Tight control over these two errors is needed in many practical situations. For example, an error of one type may have far more serious consequences than one of the other type. Most previous work on two-class classifiers does not allow for such control. In this paper, we describe a general approach that supports tight error control for two-class classification and that can utilize almost any two-class classification method as part of the decision mechanism. The main idea is to construct from the given training data a family of classifiers and then, using the training data once more, to estimate two distributions of certain vote totals. The error control is achieved via the two estimated distributions. The control is effective if the estimates of the two distributions are close to the true distributions. The approach has been tested using six well-known classification problems. In each case, the two estimated distributions were close or very close to those obtained from verification data. Accordingly, error control would be good. In addition, when the goal was minimization of total errors, then the accuracy was essentially as good as that of the best prior methods.

Key words: Classification, Data Mining, Error Control

1. Introduction

Almost all existing two-class classification methods—including schemes improved by arcing, bagging, boosting, randomization, or stacking, and also including schemes that combine heterogenous classifiers—are designed to minimize the probability of misclassifying an arbitrary record. At times, that goal is inappropriate. For example, a preliminary screening test for cancer should not declare a malignant case to be benign and, subject to that condition, should avoid misclassification of benign cases as malignant as much as can be done by the test. Put differently, the probability for classifying a malignant case as benign should be zero or almost zero, and, subject to that condition, the probability of classifying a benign case as malignant should be as small as can be achieved by the test.

The example demands a certain kind of error control. Such control problems have been extensively studied; see, for example, Duda and Hart (1973). However, almost all existing two-class classification methods do not allow for such control.

In this paper, we describe a general approach that supports tight error control for two-class classification and that can utilize almost any two-class classification method as part of the decision mechanism. The main idea is to construct from the given training data a family of classifiers and then, using the training data once more, to estimate two distributions of certain vote totals. The error control is achieved via the two estimated distributions. The control is effective if the estimates of the two distributions are close to the true distributions.

The notion of votes for two-class classification is not new. For example, the already mentioned techniques of arcing, bagging, boosting, and stacking explicitly or implicitly use that concept. Here, we employ votes to achieve tight error control.

There exists too much prior work on two-class classification that we could cite all of it here. Hence, we confine ourselves to a listing of representative recent references. There are two-class classification methods using linear inequalities in various ways (for example, see the books by McLachlan (1992), Quinlan (1992), and Nadler and Smith (1993), and the paper by Mangasarian, Street, and Wolberg (1995)), neural network methods (for example, see the books by White (1992) and Gallant (1993)), nearest neighbor methods (for example, see the book by Nadler and Smith (1993)), and logic domain methods (for example, see Kamath, Karmarkar, Ramakrishnan, and Resende (1992), Triantaphyllou, Allen, Soyster, and Kumara (1994), Felici (1995), and Boros, Hammer, Ibaraki, Kogan, Mayoraz, and Muchnik (1996)).

For details on arcing, bagging, boosting, randomization, and stacking, see, for example, Wolpert (1992), Breiman (1996a, 1996b, 1997), Drucker and Cortes (1996), Quinlan (1996), Freund and Schapire (1997), Maclin and Optiz (1997), and Dietterich (1998). For an evaluation of combinations of possibly heterogeneous classifiers, see, for example, Kittler, Hatef, Duin, and Matas (1998).

Error control for two-class classification based on linear programming is discussed in a specific context by Mangasarian, Street, and Wolberg (1995).

Several testing processes—in particular, processes based on cross-validation and bootstrap methods—have been developed that evaluate classification methods; for example, see the books by Efron (1993) and Michie, Spiegelhalter, and Taylor (1994).

We are ready to introduce the terminology employed here and to give an overview of the main results.

We assume the following standard setting. There are two disjoint populations $\mathcal{A}$ and $\mathcal{B}$ of records. We are given subsets $A \subseteq \mathcal{A}$ and $B \subseteq \mathbb{B}$ as training data. Since $\mathcal{A}$ and $\mathcal{B}$ are disjoint, so are A and B. Except for the given sets A and B, we presently cannot obtain further information about the two populations $\mathcal{A}$ and $\mathcal{B}$. We are to construct a classification scheme that achieves a

specified control of classification errors on the population records not used in the training—that is, on the records of $\mathcal{A} - A$ and $\mathbb{B} - B$.

Specifically, let a *type* $\mathcal{A}$ *error* (resp. *type* $\mathcal{B}$ *error*) be the misclassification of a record of $\mathcal{A}$ (resp. $\mathcal{B}$). Define α (resp. β) to be the probability of a type $\mathcal{A}$ error (resp. type $\mathcal{B}$ error) on the records of $\mathcal{A} - A$ (resp. $\mathbb{B} - B$) when a given classification scheme is employed. We want a classification scheme so that α and β is controlled in a certain way. We achieve the desired control via minimization of a function. Details of the function and the control achieved by its minimization follow in a moment.

We use a three-step process to determine a classification scheme that on the records of $\mathcal{A} - A$ and $\mathbb{B} - B$ achieves the as-yet-unspecified error control.

In Step 1, we create a family $\mathcal{C}$ of classification methods using various proper subsets of the training data A and B. This step is independent of the type of error control desired.

One may produce the members of $\mathcal{C}$ by applying almost any existing method for two-class classification to certain subsets of the training data. In our case, we employ a method based on logic that has proved to be fast and reliable. The selection of the subsets for the training of the members of $\mathcal{C}$ is non-random and results in a systematic use of all training records.

For a given record of $\mathcal{A}$ or $\mathcal{B}$, each member of $\mathcal{C}$ produces an integer *vote count*. Informally speaking, that count tends to be positive for a record of $\mathcal{A}$ and negative for a record of $\mathcal{B}$. The sum of the vote counts of the members of $\mathcal{C}$ is the *vote total* z for that record. Since $\mathcal{C}$ was trained on A and B, the vote totals of $\mathcal{C}$ for records of $\mathcal{A} - A$ and $\mathbb{B} - B$ can be viewed as samples of random variables $Z_{\mathcal{A}}$ and $Z_{\mathbb{B}}$, respectively.

In Step 2, we use the training data once more and estimate the distributions of the random variables $Z_{\mathcal{A}}$ and $Z_{\mathbb{B}}$. This can be done since each member of the family $\mathcal{C}$ was established using a proper subset of the training data.

In Step 3, we derive from the family $\mathcal{C}$ a set $\mathcal{D}$ of *decision schemes*, as follows. For each possible vote total z, $\mathcal{D}$ is defined to contain a decision scheme D_z that declares a record to be in $\mathcal{A}$ if the vote total is greater than or equal to z and that declares the record to be in $\mathcal{B}$ otherwise.

In agreement with the earlier definitions, let α (resp. β) be the probability that D_z misclassifies a record of $\mathcal{A} - A$ (resp. $\mathbb{B} - B$). The estimated distributions for $Z_{\mathcal{A}}$ and $Z_{\mathbb{B}}$ directly provide estimates for α and β, say $\hat{\alpha}$ and $\hat{\beta}$.

We achieve error control via an *error control function* $f(\alpha, \beta)$. In the general case, the function $f(\alpha, \beta)$ is rational and monotone increasing in both α and β. We allow infinity as function value. It indicates unacceptable error control. Due to the monotonicity, function values decrease as errors decrease, and decision schemes that minimize $f(\alpha, \beta)$ achieve a certain error control. Accordingly, we want to select a decision scheme D_z of $\mathcal{D}$ whose associated α and β minimizes $f(\alpha, \beta)$. We do not know α and β for the D_z of $\mathcal{D}$, so instead we use the estimates $\hat{\alpha}$ and $\hat{\beta}$ and select a D_z that minimizes $f(\hat{\alpha}, \hat{\beta})$. If $f(\hat{\alpha}, \hat{\beta})$ of the selected decision scheme D_z is finite, then D_z is outputted as the desired decision scheme. If the value of $f(\hat{\alpha}, \hat{\beta})$ is infinite, then we declare that we have not been able to find a decision scheme with acceptable error control.

In a variation of error control, one not only desires minimization of $f(\hat{\alpha}, \hat{\beta})$, which concerns errors for the records of $\mathcal{A} - A$ and $\mathbb{B} - B$, but also demands a certain error performance on the training data A and B. That aspect is readily handled by the introduction of an error control function $g(\gamma, \delta)$ on the training data errors where γ is the error rate on the records of A and where δ is the corresponding rate for B. Here, too, we allow infinity as function value. Since the error rates γ and δ are known for each D_z, computation of the function values $g(\gamma, \delta)$ for the decision schemes D_z of $\mathcal{D}$ is straightforward. The overall error control function is then $f(\hat{\alpha}, \hat{\beta}) + g(\gamma, \delta)$.

Examples of $f(\alpha, \beta)$ and $g(\gamma, \delta)$ are as follows. Suppose we want to select a decision scheme D_z of $\mathcal{D}$ such that α does not exceed a specified value τ. Subject to that condition, we desire β to be as small as possible. This case is handled by the following function $f(\alpha, \beta)$. For any value of β: $f(\alpha, \beta) = \beta$ if $\alpha \leq \tau$, and $f(\alpha, \beta) = \infty$ otherwise. Suppose we want to impose the condition of error-free performance on the training data. We then define $g(\gamma, \delta) = \infty$ if γ or δ is positive and $g(\gamma, \delta) = 0$ otherwise.

We have implemented the approach in a system called L^2 ("Learning Logic") and have tested it using six well-known classification problems. In each case, the estimated distributions for $Z_{\mathcal{A}}$ and $Z_{\mathbb{B}}$ were close or very close to those obtained from verification data. Accordingly, estimation of the function values $f(\alpha, \beta)$ would be quite precise. In addition, when the goal was minimization of total errors—that is, when $f(\alpha, \beta)$ was the sum of appropriately weighted α and β—then the accuracy was essentially as good as that of the best prior methods.

The remainder of the paper proceeds as follows.

Sections 2–4 provide details of Steps 1–3 of the construction process.

Section 5 summarizes the implementation of L^2 and provides the computational results.

The description of Step 1 in Section 2 skips details of the logic-based approach that we use in L^2 to construct the members of the family $\mathcal{C}$. Section 6 contains a brief presentation of that material.

The final section, 7, contains the references.

2. Construction of the Classification Family

Given the training sets A and B, we want to construct a family $\mathcal{C}$ of classification methods. We begin by selecting an integer $d \geq 5$ and partitioning A into d nonempty subsets $A^1, A^2, \ldots, A^d$ of essentially equal cardinality. We justify the bound $d \geq 5$ in Section 3. In our implementation, we use $d = 10$. The assignment of the records of A to the subsets $A^1, A^2, \ldots, A^d$ is done randomly. But if A itself was selected randomly from $\mathcal{A}$, it suffices that we assign records sequentially to the subsets.

Let c be the smallest integer that is larger than $d/2$; thus, $c = \lfloor d/2 \rfloor + 1$. For the moment, view $A^1, A^2, \ldots, A^d$ as a circular list. We use indices in agreement with that convention. In particular, A^{i+j} denotes the j-th successor of A^i. For $i = 1, 2, \ldots, d$, we take the union of A^i and of the $(c-1)$ subsequent A^j and call that union A_i; that is, $A_i = \bigcup_{j=i}^{i+c-1} A^j$. Thus, we obtain $A_1, A_2, \ldots, A_d$. Applying the analogous process to B, we obtain via $B^1, B^2, \ldots, B^d$ the sets $B_1, B_2, \ldots, B_d$, where $B_i = \bigcup_{j=i}^{i+c-1} B^j$.

The derivation of $A_1, A_2, \ldots, A_d$ and $B_1, B_2, \ldots, B_d$ from A and B is the type of process employed in stratified cross-validation; for example, see the books by Efron (1993) and Michie, Spiegelhalter, and Taylor (1994). However, we use $c = \lfloor d/2 \rfloor + 1$ instead of $c = d - 1$ of the cross-validation case. The value $c = \lfloor d/2 \rfloor + 1$ is a compromise of possible values each of which is best for a certain goal. Specifically, we want good accuracy of the classification methods derived from $A_1, A_2, \ldots, A_d$ and $B_1, B_2, \ldots, B_d$. That goal turns out to require $c \geq \lfloor d/2 \rfloor + 1$; we justify that inequality in Section 3. We also want precise estimation of the mean, variance, and shape of certain probability distributions. That goal demands c to be as small as possible. Thus, $c = \lfloor d/2 \rfloor + 1$ is an appropriate compromise.

Let C be any classification method that can be developed for $\mathcal{A}$ and $\mathcal{B}$ by training on some nonempty subsets $\bar{A} \subseteq \mathcal{A}$ and $\bar{B} \subseteq \mathbb{B}$. We assume that any C obtained that way has the following properties. The method C is to output an integer *vote count* for any record of $\mathcal{A}$ or $\mathcal{B}$. For some positive e that is independent of $\bar{A}$ and $\bar{B}$, the vote count is to be $+e$ on the records of $\bar{A}$, $-e$ for the records of $\bar{B}$, and to lie in the interval from $-e$ to $+e$ for any record of $\mathcal{A}$ or $\mathcal{B}$. The

requirement of the values $+e$ and $-e$ for the records of $\bar{A}$ and $\bar{B}$ has attractive consequences, as shown in Section 3. In particular, that condition trivially implies that C can correctly classify all records of $\bar{A}$ and $\bar{B}$. In our implementation, we derive logic-based Cs that have $e = 4$. A compressed description is given in Section 6.

For $i = 1, 2, \ldots, d$, we use the sets A_i and B_i as sets $\bar{A}$ and $\bar{B}$ to obtain a classification method C_i that observes the above conditions for a fixed e. We collect the methods C_1, $C_2, \ldots, C_d$ in a family $\mathcal{C}$ of classification methods. When the methods C_1, $C_2, \ldots, C_d$ of $\mathcal{C}$ are applied to one record of $\mathcal{A}$ or $\mathcal{B}$, then they produce d vote counts. We call the sum of these d vote counts the *vote total* z for that record. Since each vote count ranges from $-e$ to $+e$, the vote total z ranges from $-d \cdot e$ to $+d \cdot e$. We use an abbreviated terminology to describe the above process. For example, we say that $\mathcal{C}$ is *applied* to a record or that it *generates* or *produces* a vote total z.

We base on $\mathcal{C}$ a family $\mathcal{D}$ of *decision schemes* D_z, where z ranges over the possible vote totals of $\mathcal{C}$. Decision scheme D_z declares a record of $\mathcal{A}$ or $\mathcal{B}$ to be in $\mathcal{A}$ if the vote total produced by $\mathcal{C}$ for that record is greater than or equal to z and declares the record to be in $\mathcal{B}$ otherwise.

When $\mathcal{C}$ is applied to the records of $\mathcal{A} - A$, which are the records of $\mathcal{A}$ not used for training of any C_i of $\mathcal{C}$, then the vote totals so produced may be viewed as samples of a random variable $Z_{\mathcal{A}}$. The scheme D_z classifies a record correctly if the vote total is greater than or equal to z, and thus does so with probability $P(Z_{\mathcal{A}} \geq z)$. Conversely, misclassification by D_z of a record of $\mathcal{A} - A$ and thus a type $\mathcal{A}$ error occur with probability $\alpha = P(Z_{\mathcal{A}} < z)$.

Analogous results hold for $\mathcal{B}$. Let $Z_{\mathbb{B}}$ be the random variable of vote counts of $\mathcal{C}$ on $\mathbb{B} - B$. Then correct classification of records of $\mathbb{B} - B$ by D_z occurs with probability $P(Z_{\mathbb{B}} < z)$, and a misclassification and thus a type $\mathcal{B}$ error occur with probability $\beta = P(Z_{\mathbb{B}} \geq z)$.

We conclude that α and β are at hand for each D_z if we know the distributions of $Z_{\mathcal{A}}$ on $\mathcal{A} - A$ and of $Z_{\mathbb{B}}$ on $\mathbb{B} - B$. In the next section, we see how the two distributions can be estimated.

3. Estimating the Distributions

We want to estimate the distributions of $Z_{\mathcal{A}}$ and $Z_{\mathbb{B}}$ on $\mathcal{A} - A$ and $\mathbb{B} - B$, respectively. Due to the symmetry, it suffices that we discuss estimation of the distribution of $Z_{\mathcal{A}}$ on $\mathcal{A} - A$. We carry out that task by first estimating the mean and variance of $Z_{\mathcal{A}}$, then estimating the distribution of a related random variable Y, and finally combining these estimates to an estimated distribution for $Z_{\mathcal{A}}$.

We need the notion of unseen records for any member C_i of $\mathcal{C}$. That is, if a record k of $\mathcal{A}$ was not used in the training of C_i, then it is *unseen* for C_i. We denote the vote count of C_i for such a record k by $v(i, k)$. Let X_i be the random variable that represents the vote count of C_i for unseen records of $\mathcal{A}$. Since the proper subset A_i of A was used in the training of C_i, the records k of $\bar{A}_i = A - A_i$ are precisely the unseen records of A for C_i, and the corresponding vote counts $v(i, k)$ are sample values for X_i that may be used to estimate via standard formulas the mean and variance of X_i as

$$\hat{\mu}_{X_i} = [1/|\bar{A}_i|] \sum_{k \in \bar{A}_i} v(i, k) \tag{3.1}$$

$$\hat{\sigma}^2_{X_i} = [1/(|\bar{A}_i| - 1)] \sum_{k \in \bar{A}_i} [v(i, k) - \hat{\mu}_{X_i}]^2 \tag{3.2}$$

respectively. Since $Z_{\mathcal{A}}$ is the vote total of $\mathcal{C}$ and thus is the sum of the vote counts of the C_i,

$$Z_{\mathcal{A}} = \sum_{i=1}^{d} X_i \tag{3.3}$$

Hence, the mean value for $Z_{\mathcal{A}}$ is estimated by

$$\hat{\mu}_{Z_{\mathcal{A}}} = \sum_{i=1}^{d} \hat{\mu}_{X_i} \tag{3.4}$$

The variance of $Z_{\mathcal{A}}$ is the sum of the entries of the covariance matrix for X_1, $X_2, \ldots, X_d$. For the diagonal entries of that matrix, which are the variances, we have already the estimates from (3.2). We estimate the remaining entries as follows.

For $i \neq j$, the set $\bar{A}_{ij} = \bar{A}_i \cap \bar{A}_j$ is the set of records that are unseen for both C_i and C_j. If $\bar{A}_{ij}$ is nonempty, then we estimate the covariance of X_i and X_j as

$$\hat{\sigma}_{X_i X_j} = [1/|\bar{A}_{ij}|] \sum_{k \in \bar{A}_{ij}} [v(i,k) - \hat{\mu}_{X_i}][v(j,k) - \hat{\mu}_{X_j}] \tag{3.5}$$

Except for the denominator $|\bar{A}_{ij}|$, (3.5) is the standard formula for unbiased estimation of the covariance. The denominator should be $|\bar{A}_{ij}| - 1$ if $\hat{\mu}_{X_i}$ and $\hat{\mu}_{X_j}$ were computed using just the records of $\bar{A}_{ij}$. But here we estimate $\hat{\mu}_{X_i}$ and $\hat{\mu}_{X_j}$ using the sets $\bar{A}_i$ and $\bar{A}_j$, respectively, which properly contain $\bar{A}_{ij}$. We are not aware of results covering that situation, and have chosen the denominator $|\bar{A}_{ij}|$, which seems more appropriate than $|\bar{A}_{ij}| - 1$. At any rate, the cardinalities of $\bar{A}_{ij}$ occurring in practical applications are usually large enough that the two choices produce very similar estimates.

We must still estimate the covariance for the X_i and X_j for which $\bar{A}_{ij}$ is empty. Compared with the number of covariance values already estimated, the number of values yet to be estimated is not large unless d is small. Indeed, it is easily checked that the ratio of the number of covariance values estimated via (3.5) divided by the total number of covariance values is $2(d-c-1)/(d-1)$, which, for example, is 2/3 for $d = 10$, and which rapidly approaches 1 as d increases. On the other hand, that ratio is 0 for $d = 4$ and 1/2 for $d = 5$, which justifies the condition $d \geq 5$ introduced in Section 2. Due to these facts, we estimate the remaining covariance values via a linear function of the form $a \cdot |A_i \cap A_j| + b$. We use ridge regression (Hoerl and Kennard (1970a, 1970b)) and the $\hat{\sigma}_{X_i X_j}$ values computed via (3.5) to determine a and b. Since increasing values of $|A_i \cap A_j|$ reflect a larger common subset of training data for C_i and C_j, such increasing values should produce a larger covariance estimate. Accordingly, we would want the coefficient a of the linear function to be nonnegative. If that is so, then for each empty $\bar{A}_{ij}$ we use the linear function to estimate the covariance of X_i and X_j. If the coefficient a is negative, as we have observed occasionally, then the linear function is not appropriate. In that case, we discard the linear function and instead average all $\hat{\sigma}_{X_i X_j}$ values obtained via (3.5) to obtain the estimated covariance for all X_i and X_j with empty $\bar{A}_{ij}$. The latter calculation may seem a bit simplistic. But later we only use the sum of all estimated covariance values, so overestimation of individual values may be compensated for by underestimation and *vice versa*.

Suppose we have computed the estimates $\hat{\sigma}_{X_i X_j}$ for all X_i and X_j. Since the variance of $Z_{\mathcal{A}}$ is the sum of the entries of the covariance matrix of X_1, $X_2, \ldots, X_d$, we estimate the variance of $Z_{\mathcal{A}}$ as

$$\hat{\sigma}^2_{Z_{\mathcal{A}}} = \sum_{i=1}^{d} \hat{\sigma}^2_{X_i} + 2 \cdot \sum_{i<j} \hat{\sigma}_{X_i X_j} \tag{3.6}$$

When d is not small, say when $d \geq 10$, then one may be tempted to guess that $Z_{\mathcal{A}}$, which is the sum of the d random variables X_1, $X_2, \ldots, X_d$, has a distribution that can be approximated by the normal distribution. But examination of a few test examples quickly dispels that notion.

So we discard that untenable assumption and obtain a reasonable approximation via some other sample distributions, as follows.

In Section 2, we viewed $A^1, A^2, \ldots, A^d$ as a circular list. We now do this for $\bar{A}_1, \bar{A}_2, \ldots, \bar{A}_d$, for $C_1, C_2, \ldots, C_d$, and for $X_1, X_2, \ldots, X_d$ as well. As before, we use indices in agreement with this definition. In particular, A_{i-1} used later is the immediate predecessor of A_i, and X_{i+j} is the j-th successor of X_i.

For $i = 1, 2, \ldots, d$, define Y_i to be the sum of random variable X_i and the next $(d - c - 1)$ random variables X_j. We use $\tilde{C}_i$ to denote the method that to a given record applies C_i and the next $(d - c - 1)$ members C_j and that outputs the sum of the vote counts so obtained. We call that output the *vote sum* of $\tilde{C}_i$. We define a record to be *unseen* for $\tilde{C}_i$ if it is unseen for C_i and the next $(d - c - 1)$ members C_j. Evidently, the vote sums of $\tilde{C}_i$ on unseen data are samples of Y_i that may be used to estimate the distribution of Y_i.

We know that, for each j, $\bar{A}_j$ contains the records of A that are unseen for C_j. Hence, the intersection of $\bar{A}_i$ and the next $(d - c - 1)$ $\bar{A}_j$ contains the records of A that are unseen for $\tilde{C}_i$. It is easily checked that A^{i-1} is that intersection. Hence, the vote sums of $\tilde{C}_i$ for the records of A^{i-1} may be used to estimated the distribution of Y_i.

Let Y be the average of the Y_i. Since

$$\begin{aligned} Y &= [1/d] \sum_{i=1}^{d} Y_i \\ &= [1/d] \sum_{i=1}^{d} (X_i + X_{i+1} + \ldots + X_{i+d-c-1}) \\ &= [(d-c)/d] \sum_{i=1}^{d} X_i \\ &= [(d-c)/d] \cdot Z_{\mathcal{A}} \end{aligned} \tag{3.7}$$

we effectively have an estimate of the distribution of $Z_{\mathcal{A}}$ once we have an estimate for the distribution of Y.

Since no record of A is unseen for all C_i, we cannot compute samples of Y directly from the records of A and estimate the distribution of Y from these samples. But there is a plausible argument that supports estimation of that distribution from estimated distributions of the Y_i, as follows.

First, for any $k \neq l$, X_k and X_l are likely to be positively correlated since C_k and C_l, which were trained on partially identical training data, presumably are similarly voting classifiers. Second, each Y_i is the sum of $(d-c)$ X_k, and several pairs Y_i and Y_j contain common X_k. These facts support the conjecture that any two Y_i and Y_j are positively correlated and have similar distributions. Accordingly, one is justified to guess that a reasonable estimate of the distribution of Y can be obtained by averaging estimated distributions for the Y_i. Toward the end of this section, we report empirical evidence that provides strong support for this guess.

We proceed to estimate the distribution for Y using the just described approach. Since the records of A unseen for $\tilde{C}_i$ constitute the set A^{i-1}, each one of the sets $A^1, A^2, \ldots, A^d$ may be used to obtain samples for a different Y_i. Furthermore, the sets $A^1, A^2, \ldots, A^d$ are essentially of equal cardinality. This implies that we may estimate the distribution of Y as an average of the distributions of the Y_i by simply taking each record of A, applying the unique $\tilde{C}_i$ for which that record is unseen to obtain a vote sum, and finally view the vote sums so found as samples of Y. Accordingly, we compute from these samples by standard formulas an estimated mean $\hat{\mu}_Y$, an estimated variance $\hat{\sigma}_Y$, and an estimated probability density function $h(y)$ for Y.

We could use the relationship $Y = [(d - c)/d] \cdot Z_{\mathcal{A}}$ of (3.7) to convert the estimated density function for Y to one for $Z_{\mathcal{A}}$, but shall not do so. Instead, we derive from the estimated density function for Y an estimated density function for $Z_{\mathcal{A}}$ that has as mean and variance the estimated values $\hat{\mu}_{Z_{\mathcal{A}}}$ and $\hat{\sigma}^2_{Z_{\mathcal{A}}}$ of (3.4) and (3.6). One might argue that the two possible ways of estimating a distribution for $Z_{\mathcal{A}}$ should produce the same outcome. But this need not be the case since the variance estimate of (3.6) is partially based on estimation via a linear function and thus need not match the variance of $Z_{\mathcal{A}}$ computed directly from the estimated variance of Y. This has been confirmed by test calculations. We should note, however, that these tests also showed that the difference between the two variance estimates for $Z_{\mathcal{A}}$ was at most moderate.

The derivation of the estimated distribution for $Z_{\mathcal{A}}$ is therefore done as follows. We take each value y of Y for which we have a positive estimated probability density value $h(y)$ and transform it to a value z using

$$z = (y - \hat{\mu}_Y)(\hat{\sigma}_{Z_{\mathcal{A}}}/\hat{\sigma}_Y) + \hat{\mu}_{Z_{\mathcal{A}}} \tag{3.8}$$

We round the z of (3.8) to the nearest integer for which $Z_{\mathcal{A}}$ may have a positive probability density and assign $h(y)$ as the probability density value to z. It may happen that several probability density values of Y are assigned to the same z value. In particular, this is possible for the largest or smallest possible value z that $Z_{\mathcal{A}}$ may take on with positive probability. For each such case, the $h(y)$ values assigned to the same z are added together. Note that the rounding of z values causes the resulting probability density function to have mean and variance not exactly equal to $\hat{\mu}_{Z_{\mathcal{A}}}$ and $\hat{\sigma}^2_{Z_{\mathcal{A}}}$ of (3.4) and (3.6). In tests, the differences were small enough to be of no concern.

At this point, we have obtained an estimated probability density function $f_{\mathcal{A}}(Z_{\mathcal{A}})$ for $Z_{\mathcal{A}}$. By letting B play the role of A in the above process, we also obtain an estimated probability density function $f_{\mathbb{B}}(Z_{\mathbb{B}})$ for $Z_{\mathbb{B}}$.

For the possible values z of the vote totals of $\mathcal{C}$, we use the estimated distributions for $Z_{\mathcal{A}}$ and $Z_{\mathbb{B}}$ to compute estimates $\hat{\alpha}$ and $\hat{\beta}$ for $\alpha = P(Z_{\mathcal{A}} < z)$ and $\beta = P(Z_{\mathbb{B}} \geq z)$. As discussed earlier, $\hat{\alpha}$ (resp. $\hat{\beta}$) is an estimate of the probability that the decision scheme D_z of $\mathcal{D}$ commits a type $\mathcal{A}$ error (resp. type $\mathcal{B}$ error) on the records of $\mathcal{A} - A$ (resp. $\mathbb{B} - B$). We also apply $\mathcal{C}$ to A and B and compute the error rates γ and δ that each D_z commits on A and B, respectively. In the next section, we use the values $\hat{\alpha}$, $\hat{\beta}$, γ, and δ obtained for each z to select a decision scheme.

For the moment, let us consider $Z_{\mathcal{A}}$ and $Z_{\mathbb{B}}$ over $\mathcal{A}$ and $\mathcal{B}$, respectively. We want to determine the range of values these random variables can take on with nonzero probability in our implementation. As described later in Section 6, our logic approach produces a family $\mathcal{C}$ where the vote of each classification member C_i is even and ranges from $-e$ to $+e$, where $e = 4$. On the records of A (resp. B) used in the training of C_i—that is, on A_i (resp. B_i)—the vote of C_i is equal to $+e$ (resp. $-e$), as required earlier. We use $d = 10$, so the vote total $Z_{\mathcal{A}}$ or $Z_{\mathbb{B}}$ ranges from $-e \cdot d = -40$ to $+e \cdot d = 40$. Any record of A is used in the training of $c = \lfloor d/2 \rfloor + 1 = 6$ of the members C_i of $\mathcal{C}$, so $Z_{\mathcal{A}}$ for any record in A is at least $e \cdot [c - (d - c)] = 8$. Similarly, $Z_{\mathbb{B}}$ for any record of B is at most $-e \cdot [c - (d - c)] = -8$. These results imply that any decision scheme D_z with z in the closed interval from -7 to $+8$ classifies all records of A and B correctly. That desirable conclusion is the main reason for the earlier stated inequality $c \geq \lfloor d/2 \rfloor + 1$ and for the condition that the selected two-class classification subroutine must produce Cs whose vote count is $+e$ and $-e$ for the records of the training subsets taken from A and B.

We conclude this section with results of tests that empirically justify the estimation of the distribution of Y by the average of the estimated distributions for $Y_1, Y_2, \ldots, Y_d$. The test data are taken from the six data sets Australian Credit Card, Boston Housing, Breast Cancer, Congressional Voting, Diabetes, and Heart Disease, which are described in detail and with

applicable references in Section 5.

For each of the six data sets $\mathcal{A}$ and $\mathcal{B}$, we randomly select as A and B half the data and construct a family $\mathcal{C}$. To establish validity of the estimating procedure for the distribution of $Z = Z_{\mathcal{A}}$ or $Z = Z_{\mathbb{B}}$ via estimated distributions of the Y_i, we apply $\mathcal{C}$ to $\mathcal{A} - A$ or $\mathbb{B} - B$, respectively, to get estimated distributions for Z and for the Y_i. We emphasize that we apply $\mathcal{C}$ to $\mathcal{A} - A$ or $\mathbb{B} - B$ and not to any subsets of A or B. The reason is that we want to establish the validity of guessing the distribution of Z from those of the Y_i and thus want to eliminate any variability not related to that comparison. For the same reason, we estimate the mean $\hat{\mu}_Z$ and variance $\hat{\sigma}_Z^2$ of Z from the just determined distribution.

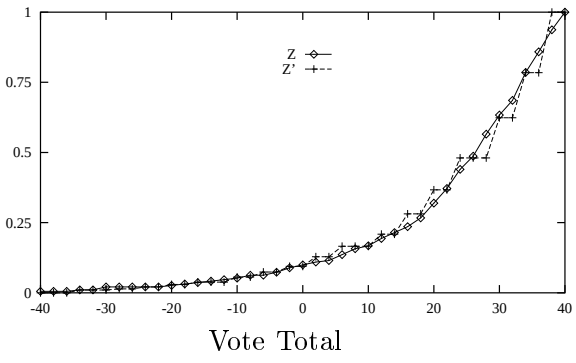
We average the estimated distributions for the Y_i to obtain an estimated distribution for Y. Let $\hat{\mu}_Y$ and $\hat{\sigma}_Y^2$ be the mean and variance of the latter distribution.

We transform the estimated distribution for Y using the equation

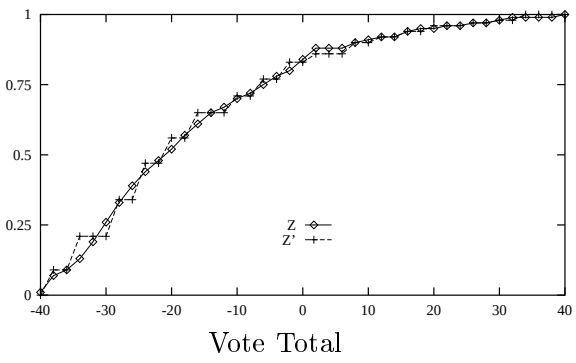
$$z = (y - \hat{\mu}_Y)(\hat{\sigma}_Z/\hat{\sigma}_Y) + \hat{\mu}_Z \tag{3.9}$$

and the rounding process described in connection with (3.8). We denote by Z' the random variable represented by the resulting distribution. Finally, we compare the distribution of Z' with the estimated distribution of Z. A close match would empirically justify the approximation of the distribution of Z via those of the Y_i.

For each pair of A and B derived from the six data sets, we get two cases of Z and Z'. Hence, we have a total of twelve pairs Z and Z'. In each case, the distribution of Z' turned out to be virtually identical to the estimated distribution of Z. Below, we show graphs of the distributions for the pairs obtained from the Australian Credit Card data set.



Australian Credit Card: Distributions for $Z = Z_{\mathcal{A}}$ and related Z'



Australian Credit Card: Distributions for $Z = Z_{\mathbb{B}}$ and related Z'

These results empirically justify the estimation of the distribution of Y by the average of the estimated distributions for $Y_1, Y_2, \ldots, Y_d$.

We are ready for the third step, where we select a decision scheme.

4. Selecting a Decision Scheme

Assume that we have for each decision scheme D_z of $\mathcal{D}$ the estimated error probabilities $\hat{\alpha}$ and $\hat{\beta}$ and the error rates γ and δ. We select appropriate error control functions $f(\alpha, \beta)$ and $g(\gamma, \delta)$. If error control on A and B is not of interest, we declare $g(\gamma, \delta)$ to be identically 0. By straightforward enumeration of the possible vote totals z, we select a decision scheme D_z that minimizes $f(\hat{\alpha}, \hat{\beta}) + g(\gamma, \delta)$. If that minimum is finite, we output a D_z achieving the minimum as the desired decision scheme. Otherwise, we declare that we could not develop a decision scheme with the desired error control.

5. Implementation and Tests

The two-class classification approach and error control process have been implemented in a data mining tool called L^2 ("Learning Logic"). The tool also handles multi-class classification and regression. We shall not describe the latter methods in this paper, but note that they use the two-class classification scheme as a subroutine and employ the error probabilities computed by that scheme in a Bayesian approach. Details are given in Sun (1998).

L^2 constructs the family $\mathcal{C}$ of classification methods using a logic-based approach. Section 6 contains a summary of that technique.

We have tested the accuracy of L^2 with regard to error control using six well-known data sets called Australian Credit Card, Boston Housing, Breast Cancer, Congressional Voting, Diabetes, and Heart Disease. The data for these problems may be obtained from the Repository of Machine Learning Databases and Domain Theories maintained by the University of California at Irvine.

The logic-based approach requires the records of $\mathcal{A}$ and $\mathcal{B}$ to contain $\{0, \pm 1\}$ entries only. We call these $\{0, \pm 1\}$ entries *logic entries* to differentiate them from other types of entries such as Boolean, integer, nominal, or rational. A nominal entry is a member of a finite set of descriptive terms. The interpretation associated with the logic entries 0, +1, and −1 does not matter. For example, one might associate +1 with *True* or *Yes*, −1 with *False* or *No*, and 0 with "Do not know value."

Some entries of the six data sets are not in the $\{0, \pm 1\}$ form and require transformations. We summarize that step as we describe the six sets and the computational results obtained for them.

Australian Credit Card

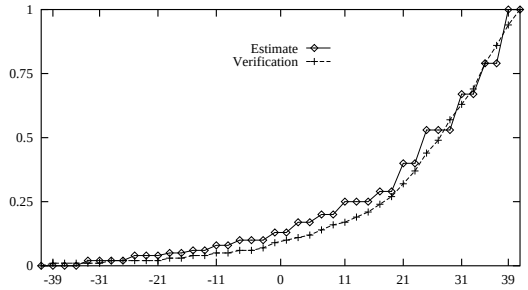
The data were made available by J. R. Quinlan. They represent 690 MasterCard applicants of which 307 are declared as positive and 383 as negative. The data contain 37 records with missing entries. Each record consists of 15 attributes, of which 4 are Boolean, 5 nominal, and 6 rational. The 4 Boolean entries are already in the desired form. The 5 nominal entries result in 29 logic entries. Each one of the 6 rational entries is converted to 5 to 9 logic entries corresponding to suitably selected intervals. After the transformation, each record has 67 logic entries. We declare $\mathcal{A}$ (resp. $\mathcal{B}$) to be the set of negative (resp. positive) records.

The conversion causes one record to occur in both $\mathcal{A}$ and $\mathcal{B}$. To get consistency, we must delete that record from $\mathcal{A}$ or $\mathcal{B}$. Tests showed that the choice of deletion from $\mathcal{A}$ versus deletion

from $\mathcal{B}$ had no significant effect. Hence, we randomly decided to delete the record from $\mathcal{A}$. The resulting two consistent sets $\mathcal{A}$ and $\mathcal{B}$ represent the positive and negative applicants, respectively.

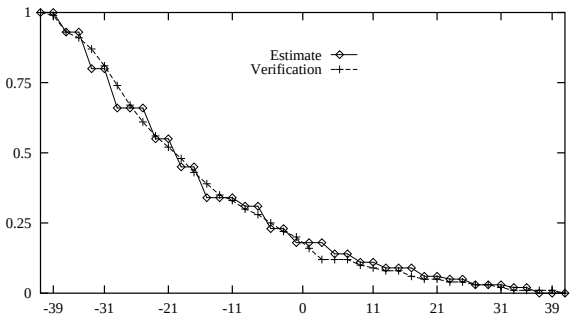
We obtain from $\mathcal{A}$ and $\mathcal{B}$ randomly selected subsets A and B, each containing 50% of the respective source set. We apply L^2 to A and B, obtain the family $\mathcal{C}$ of classification methods, and compute the estimated error probabilities $\hat{\alpha}$ and $\hat{\beta}$. Then we apply $\mathcal{C}$ to $\mathcal{A} - A$ and $\mathbb{B} - B$ to verify the error probabilities.

The graphs below show the results. The curves plotted with diamonds are the estimated $\hat{\alpha}$ and $\hat{\beta}$, while the curves plotted with crosses are the verified values. Evidently, there is very good agreement between the curves.



Decision Parameter z

Australian Credit Card: estimated and verified α



Decision Parameter z

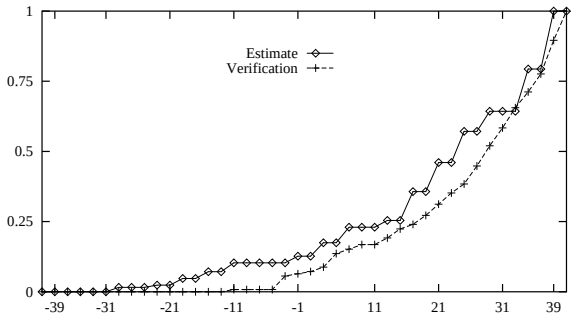
Australian Credit Card: estimated and verified β

Boston Housing

The data set was created by Harrison and Rubinfeld (1978). The data consist of 506 records concerning housing values in the Boston area. Each record is composed of 13 attributes, of which 12 have rational values, while one is Boolean. The median value of the owner-occupied houses is used as a threshold to split the entire set of records into two sets.

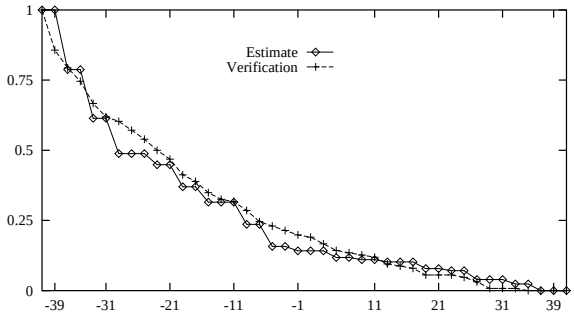
We construct the logic records as for Australian Credit Card, except that throughout we use 9 intervals for each one of the 12 rational attributes. The total number of logic variables is 109.

It turns out that the sets $\mathcal{A}$ and $\mathcal{B}$ so defined from the original two data sets have two records in common. As for the Australian Credit Card case, preliminary tests showed that it did not matter from which set the two records were deleted. We randomly decided to delete two records from $\mathcal{B}$. The remaining process is exactly as for the Australian Credit Data. That is, we randomly select 50% of $\mathcal{A}$ and $\mathcal{B}$ as the training data A and B, etc.



Decision Parameter z

Boston Housing: estimated and verified α



Decision Parameter z

Boston Housing: estimated and verified β

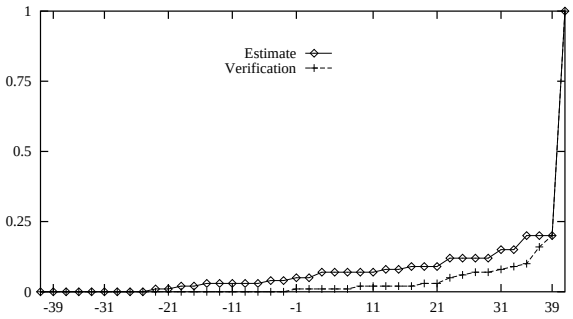
Above are the curves for $\hat{\alpha}$ and $\hat{\beta}$ and the related verified values. The agreement among the curves is good.

Breast Cancer

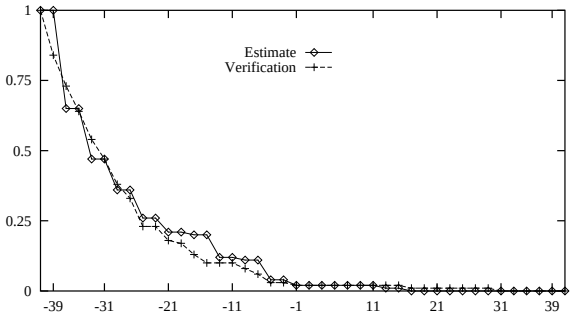
Mangasarian, Setiono, and Wolberg (1990) provide breast cancer data for 699 patients. Each record corresponds to a malignant or benign case, and contains 9 integer entries, with values raining from 1 to 10. Of the 699 records, 16 have missing entries. There are a total of 241 malignant and 458 benign cases.

To convert the integers to logic entries, we group the values 1, 2, ..., 10 into 5 intervals containing 2 consecutive integers each and correspondingly define 5 logic entries. Specifically, if an integer entry falls into the kth interval, then the kth logic entry is $+1$ and the other 4 logic entries are -1. If the integer entry is missing, all 5 logic entries are 0. The transformation produces 45 entries for each logic record. We declare $\mathcal{A}$ (resp. $\mathcal{B}$) to be the records of the benign (resp. malignant) cases. The remaining process is as before.

The curves for $\hat{\alpha}$ and $\hat{\beta}$ and for the corresponding verified values are given below. The agreement among the curves is very good.



Breast Cancer: estimated and verified α

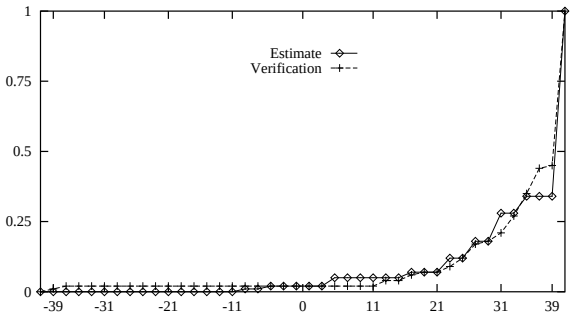


Breast Cancer: estimated and verified β

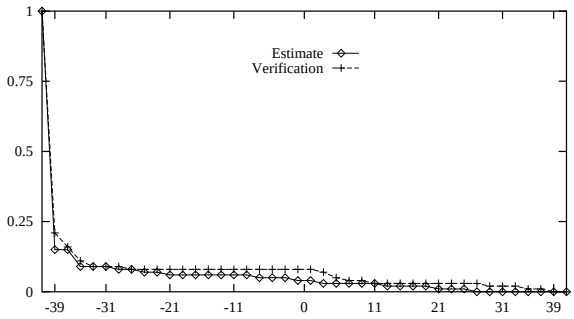
Congressional Voting

The problem concerns the prediction of party affiliation from 435 voting records of 267 Democrats and 168 Republicans. The data were assembled by Schlimmer (1987). Each record contains 16 entries of the form "for", "against", and "did not vote". In the logic data, we represent "for" by 1, "against" by -1, and "did not vote" by 0. We define $\mathcal{A}$ (resp. $\mathcal{B}$) to be the set of records of the Republicans (resp. Democrats). The remaining process is as before.

Below are the results. The agreement among the curves is excellent.



Congressional Voting: estimated and verified α



Decision Parameter z

Congressional Voting: estimated and verified β

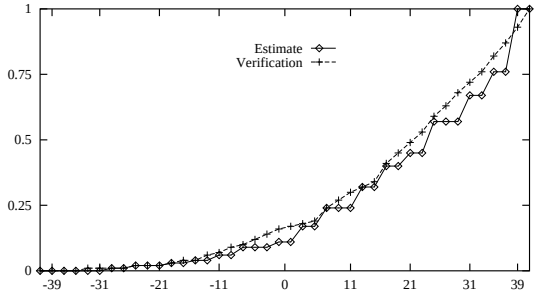
Diabetes

This problem concerns the diagnosis of diabetes based on observations for 768 patients, of which 268 had signs of diabetes, while 500 did not. The data were provided by V. G. Sigillito.

There are 8 attributes, of which 2 have discrete values, and 6 are rational. We construct the logic records as for Australian Credit Card, except that we use 7 intervals for 5 of the 6 rational attributes and 9 intervals for the remaining rational attribute. The transformations produce a total of 55 logic entries for each record.

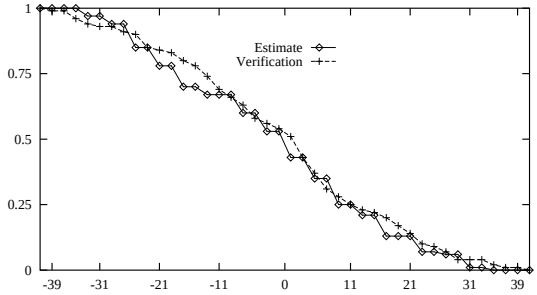
We let $\mathcal{A}$ be the set of records derived from patients not showing signs of diabetes and define $\mathcal{B}$ to be the set of the records derived from the remaining patients. The sets $\mathcal{A}$ and $\mathcal{B}$ have one record in common. A preliminary analysis reveals that it does not matter from which set we delete that record. A random choice results in deletion from $\mathcal{A}$. The rest of the process is as before.

Here are the results. The agreement among the curves is very good.



Decision Parameter z

Diabetes: estimated and verified α



Decision Parameter z

Diabetes: estimated and verified β

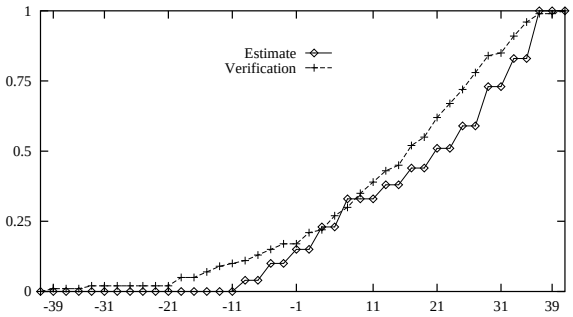
Heart Disease

Observations for 303 patients are given, of which 165 are healthy, while 139 have some heart disease. Among the 303 records, 6 have some missing entries. The data were collected by R. Detrano.

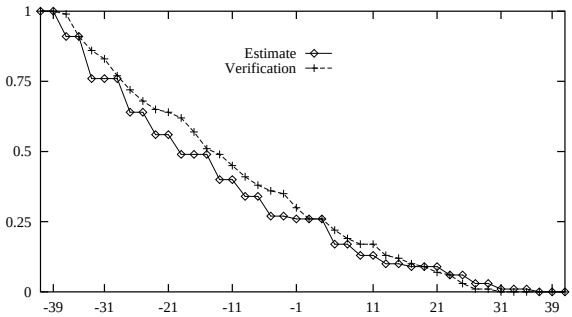
Each record provides 13 attributes, of which 3 are Boolean, 4 nominal, and 6 rational. We transform the records to logic data with the method employed for Australian Credit Card, ending up with a total of 50 logic entries for each record.

We collect in $\mathcal{A}$ (resp. $\mathcal{B}$) the logic records corresponding to the healthy patients (resp. the patients with heart disease). The remaining process is as before.

The computational results are given below. The agreement among the curves is good.



Heart Disease: estimated and verified α



Heart Disease: estimated and verified β

Besides the above tests, we have also checked the average prediction performance of our approach for the above six data sets, as follows. For each of the six $\mathcal{A}$ and $\mathcal{B}$ pairs, we obtain 10 pairs A and B and test prediction accuracy on the verification data $\mathcal{A} - A$ and $\mathbb{B} - B$. We use simply majority of the vote totals to make the decision and break a tie by declaring the record to be in $\mathcal{B}$. This rule corresponds to the decision scheme D_z with $z = 1$. Besides this test, we also use the population proportions and the $\hat{\alpha}$ and $\hat{\beta}$ predicted by the earlier calculations for D_z with $z = 1$ to check how well we predict the average accuracy obtained in the 10 runs. The results are as follows, where in each case the first percentage figure is the average accuracy computed from the 10 runs, while the percentage figure in parentheses is the accuracy predicted via $\hat{\alpha}$ and $\hat{\beta}$ and the population proportions. Except for the Boston Housing and Diabetes data, the predicted accuracy is pleasingly close to the actual one. For the Boston Housing and Diabetes data, the predictions are reasonably accurate.

Australian Credit Card: 86.0% (84.9% predicted)
Boston Housing: 83.5% (86.5% predicted)
Breast Cancer: 97.1% (96.0% predicted)
Congressional Voting: 95.8% (96.8% predicted)
Diabetes: 73.3% (77.7% predicted)
Heart Disease: 80.7% (79.9% predicted)

These average accuracy results essentially match those of the best prior methods. For a comparison, see Boros, Hammer, Ibaraki, Kogan, Mayoraz, and Muchnik (1996).

The computing effort for constructing the family $\mathcal{C}$ of classification methods for a given pair A and B of the six data sets is reasonable. In the worst case, it takes less than 20 minutes on a Sun UltraSparc 1 to process completely any one of the data sets.

The description of the entire approach of error control is complete except for a discussion of the technique used to compute each member of the family $\mathcal{C}$. We cover that aspect in the next section.

6. Logic-Based Technique for Classification

We give a compressed description of our logic-based technique for finding the classification members C of the family $\mathcal{C}$. Details are included in Felici and Truemper (1997). The method is related to prior work by Kamath, Karmarkar, Ramakrishnan, and Resende (1992), Triantaphyllou, Allen, Soyster, and Kumara (1994), and Boros, Hammer, Ibaraki, Kogan, Mayoraz, and Muchnik (1996). In particular, our method may be viewed as that of Triantaphyllou *et al.* (1994) with added optimization of the selected logic formulas and symmetric treatment of the given two training sets. In the interest of brevity, we omit all proofs.

Given are two nonempty subsets A and B taken from populations $\mathcal{A}$ and $\mathcal{B}$, respectively. The populations consist of $\{0, \pm 1\}$ records of the same length $n \geq 1$. We view the entries 0, +1, and -1 just as three symbols and thus do not attach any numerical interpretation to them. The same comment applies to the entries of the vector s defined shortly. Note that the sets A and B used here play the roles of A_i and B_i of the preceding sections. We make that switch to simplify the notation employed here.

Assume A and B to be consistent. Thus,

$$A \cap B = \emptyset \tag{6.1}$$

A $\{0, \pm 1\}$ vector s of length n *separates* a vector $b \in B$ from A if

$$\forall a \in A: \ \exists i \ s_i \neq 0 \text{ and } s_i \neq a_i \tag{6.2}$$

and

$$\forall i: \ s_i \neq 0 \text{ implies } s_i = b_i \tag{6.3}$$

A set S *separates* a subset B^* of B from A if each $s \in S$ satisfies (6.2) and if, for each $b \in B^*$, there exists an $s \in S$ that separates b from A. When such S exists, we say that B^* can be separated from A. The following result is easily established.

(6.4) Lemma. *There exists a unique subset $B^* \subseteq B$ with maximum cardinality that can be separated from A.*

Let S^* be a separating set that separates B^* of Lemma 6.4 from A. Let r be a record of $\mathcal{A}$ or $\mathcal{B}$. Declare r to be in $\mathcal{B}$ if there exists an s in S^* such that (6.3) with $b = r$ holds, and declare

that r in $\mathcal{A}$ otherwise. Then S^* classifies the records of A and B^* correctly and those of $B - B^*$ incorrectly. We see later how we can improve on that performance.

We find an S^* that separates B^* of Lemma (6.4) from A as follows. For $i = 1, 2, \ldots, n$, we use two propositional variables p_i and q_i to represent the element s_i of a separating vector s. The variables p_i and q_i are linked with s_i by

$$s_i = \begin{cases} 1 & ; p_i = \mathit{True},\ q_i = \mathit{False} \\ -1 & ; p_i = \mathit{False},\ q_i = \mathit{True} \\ 0 & ; p_i = q_i = \mathit{False} \end{cases} \tag{6.5}$$

Note that the case $p_i = q_i = \mathit{True}$ is not accounted for. In the logic formulations below, we rule out that case by the clause $\neg p_i \vee \neg q_i$.

We find a maximum subset B' of B that can be separated from A by one separating vector by solving the following logic optimization problem (6.6) with propositional variables d_b, $b \in B$, with the already mentioned p_i and q_i, and with an integer function $c(\cdot)$ defined on $\{\mathit{True}, \mathit{False}\}$ by $c(\mathit{True}) = 1$ and $c(\mathit{False}) = 0$.

$$\begin{array}{lll} \min & \sum_{b \in B} c(d_b) & \\ & & \\ \text{s. t.} & \neg p_i \vee \neg q_i & i = 1, 2, \ldots, n \\ & (\bigvee_{i \ni a_i \neq -1} q_i) \vee & \\ & (\bigvee_{i \ni a_i \neq 1} p_i) & \forall a \in A \\ & \neg q_i \vee d_b & \forall b \in B,\ \forall i \ni b_i \neq -1 \\ & \neg p_i \vee d_b & \forall b \in B,\ \forall i \ni b_i \neq 1 \end{array} \tag{6.6}$$

Using the optimal *True/False* for each d_b, the set $B' = \{b \in B \mid d_b = \mathit{False}\}$ is the desired one. It is possible that B' is empty. In that case, no vector of B can be separated from A, and we stop.

So assume that B' is nonempty. We want a vector s that separates B' from A and that has a maximum number of nonzeros. We obtain such a vector by solving the following variation (6.7) of (6.6) where $c'(\cdot)$ used in the objective function is defined by $c'(\mathit{True}) = 0$ and $c'(\mathit{False}) = 1$.

$$\begin{array}{lll} \min & \sum_{i=1}^{n} [c'(p_i) + c'(q_i)] & \\ & & \\ \text{s. t.} & \neg p_i \vee \neg q_i & i = 1, 2, \ldots, n \\ & (\bigvee_{i \ni a_i \neq -1} q_i) \vee & \\ & (\bigvee_{i \ni a_i \neq 1} p_i) & \forall a \in A \\ & \neg q_i & \forall b \in B',\ \forall i \ni b_i \neq -1 \\ & \neg p_i & \forall b \in B',\ \forall i \ni b_i \neq 1 \end{array} \tag{6.7}$$

From the optimal solution values of the p_i and q_i, we derive via (6.5) the vector s. That vector separates B' from A and has a maximum number of nonzeros. Finally, we reduce B to $B - B'$. If the new B is empty, the set $S^* = \{s\}$ separates the original set B from A, and we are done. Otherwise, we carry out the above process recursively, starting each recursive step with A and the current B. When the process stops, we declare S^* to contain the separating vectors found during the various iterations and define B^* to be the difference between the original B and the final one.

It is possible that the solution of the first instance of (6.6) produces an empty set B'. In that case, we declare both B^* and S^* to be empty sets.

We now denote the set S^* by S_{max} to emphasize that the derivation of each separating vector involved maximization of the number of nonzeros.

We repeat the above process, but change the objective function of (6.7) by defining $c'(\mathit{True}) = 1$ and $c'(\mathit{False}) = 0$. The revised objective function produces separating vectors with minimum number of nonzero entries. Correspondingly, we label the separating set S^* obtained that way as S_{min}.

The reader should note that we have described the computation of S_{max} and S_{min} as two independent processes. Actually, they can be combined since both must solve the same instances of (6.5). This also implies that they terminate with the same B^*.

Before going on, we motivate the choice of the two objective functions for (6.6). Minimization of the two objective functions produces separating vectors with as many or as few nonzero entries as possible. Recall that we classify a record r using the condition (6.3) with $b = r$. That condition is harder (resp. easier) to satisfy when s has more (resp. fewer) nonzeros. Thus, when additional data beyond those of A and B must be classified, then the separating vectors with maximum (resp. minimum) number of nonzeros tend to avoid type $\mathcal{A}$ errors (resp. type $\mathcal{B}$ errors).

We continue the description of the algorithm. We initialize a list $S_{com/max}$ (resp. $S_{com/min}$) by inserting the just found S_{max} (resp. S_{min}) as first item.

If $B^* = B$, we do not go further. So suppose that B^* is a proper subset of B. We replace B by $B - B^*$, reverse the roles of A and B, and apply the above process to find two separating sets S_{max} and S_{min} that separate a maximum cardinality subset A^* of A from the current B. When the two separating sets S_{max} and S_{min} are at hand, we add S_{max} (resp. S_{min}) to the list $S_{com/min}$ (resp. $S_{com/max}$). Note that we have reversed the rule according to which separating sets are added to the two lists. The reversal is due to the fact that we have reversed the roles played by A and B.

If $A^* = A$, we stop. Otherwise, we reduce A to $A - A^*$ and carry out the above process so that we obtain two separating sets S_{max} and S_{min} that separate a maximum cardinality subset B^* of the current B from the current A. We add S_{max} as third item to $S_{com/max}$ and S_{min} to $S_{com/min}$.

Proceeding in the above described alternating fashion, we are guaranteed by (6.1) to complete an iteration eventually with $A^* = A$ or $B^* = B$. At that time, either one of the lists $S_{com/max}$ and $S_{com/min}$ may be used to classify the records of A and B correctly. Details follow momentarily. On additional data, $S_{com/max}$ (resp. $S_{com/min}$) tends to avoid type $\mathcal{A}$ errors (resp. type $\mathcal{B}$ errors).

Let r be a record of $\mathcal{A}$ or $\mathcal{B}$. We want to use the list $S_{com/max}$ or $S_{com/min}$ to classify r. We take the first separating set of the list. If some separating vector s of that separating set satisfies (6.3) with $b = r$, we declare that the separating set places r into $\mathcal{B}$, and stop. Otherwise, we take the second separating set of the list and apply it to r. If r is declared to be in $\mathcal{A}$, we stop. Otherwise, we proceed to the third separating set and treat it like the first one. If the result is not placement into $\mathcal{B}$, we go to the fourth separating set and treat it like the second one. Proceeding in such alternating fashion, we either declare r to be in $\mathcal{A}$ or $\mathcal{B}$ at some point, or we use up all separating sets without a decision. Assume the latter case. If the total number of separating sets in the list is odd (resp. even), then the last separating set of $S_{com/max}$ or $S_{com/min}$ failed to classify the record as being in $\mathcal{B}$ (resp. $\mathcal{A}$); accordingly, we declare r to be in $\mathcal{A}$ (resp. $\mathcal{B}$).

We add yet another layer to the process. We started the construction of the two lists by first separating a subset of B from A. We record that fact by relabeling the two lists $S_{com/max}$ and $S_{com/min}$ as $S_{com/max}(B, A)$ and $S_{com/min}(B, A)$, respectively. We could have started the process by first separating a subset of A from B. When we do this, we obtain two additional lists labeled $S_{com/max}(A, B)$ and $S_{com/min}(A, B)$. On additional records, the list $S_{com/max}(A, B)$

(resp. $S_{com/min}(A,B)$) tends to avoid type $\mathcal{B}$ errors (resp. type $\mathcal{A}$ errors).

At this point, we have a total of four separating lists: $S_{com/max}(A,B)$, $S_{com/min}(A,B)$, $S_{com/max}(B,A)$, and $S_{com/min}(B,A)$. Each one of them correctly classifies the records of A and B. Two of the lists tend to avoid type $\mathcal{A}$ errors, while the remaining two lists tend to avoid type $\mathcal{B}$ errors.

The four lists constitute one classification member C for the family $\mathcal{C}$. The member C votes on a $\{0, \pm 1\}$ record of length n as follows. We apply each one of the four lists to the record. If a list declares the record to be in $\mathcal{A}$ (resp. $\mathcal{B}$), we say that the list gives a vote of 1 (resp. -1). We add these votes to get the vote of C. Thus, the vote of C ranges from -4 to $+4$ and is even. For records of A (resp. B), the vote of C is $+4$ (resp. -4).

We have covered the entire algorithm producing a classification member C of $\mathcal{C}$ except for the solution method for the logic minimization problems (6.6) and (6.7). We employ the Leibniz System (1996) for that task. That system analyzes each problem and compiles a solution algorithm. We execute that algorithm to obtain the desired solution. The mathematics underlying the Leibniz System (1996) is described in Truemper (1998).

We should mention that the above logic-based approach has uses beyond those summarized above. For example, the approach can account for classification costs in settings where getting information about any entry of a record entails a certain cost. As another example, the approach can produce propositional formulas that can be added as axioms to other logic formulas instead of being evaluated via vote totals as done here. For details, see Felici and Truemper (1997).

7. References

Boros, E., Hammer, P. L., Ibaraki, T., Kogan, A., Mayoraz, E., and Muchnik, I. (1996), An implementation of logical analysis of data, RUTCOR Research Report 22-96, Rutgers University, NJ.

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

Breiman, L. (1996b), Stacked regressions, *Machine Learning*, 24, 49-64.

Breiman, L. (1997), Arcing classifiers, Technical Report, Statistics Department, University of California, Berkeley.

Dietterich, T. G. (1998), An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization, *Machine Learning*, to appear.

Drucker, H., and Cortes, C. (1996), Boosting decision trees, in *Advances in Neural Information Processing Systems* (D. Touretzky, M. Mozer, and M. Hasselmo, eds.) (pp. 478-485), volume 8, MIT Press, Cambridge, MA.

Duda, R. O., and Hart, P. E. (1973), *Pattern Classification and Scene Analysis*, Wiley, New York.

Efron, B. (1993), *An Introduction to the Bootstrap*, Chapman & Hall, New York.

Felici, G. (1995), *Il Problema di Riconoscimento Automatico: Proprietà ed Algoritmi di Soluzione*, Tesi di Dottorato, Biblioteca Nazionale di Roma, Italy.

Felici, G., and Truemper, K. (1997), Learning logic, Technical Report No. 450, CNR-IASI, Rome, Italy.

Freund, Y., and Schapire, R. (1997), A decision-theoretic generalization of on-line learning and an application to boosting, *Journal of Computer and System Sciences*, 55, 119-139.

Gallant, S. I. (1993), *Neural Network Learning and Expert Systems*, MIT Press, Cambridge, MA.

Harrison, D., and Rubinfeld, D. L. (1978), Hedonic prices and the demand of clean air, *Journal of Environment Economics and Management*, 5, 111-143.

Hoerl, A. E., and Kennard, R. W. (1970a), Ridge regression: Biased estimation for nonorthogonal problems, *Technometrics*, 12, 55-67.

Hoerl, A. E., and Kennard, R. W. (1970b), Ridge regression: Applications to nonorthogonal problems, *Technometrics*, 12, 69-82.

Kamath, A. P., Karmarkar, N. K., Ramakrishnan, K. G., and Resende, M. G. C. (1992), A continuous approach to inductive inference, *Mathematical Programming*, 57, 215-238.

Kittler, J., Hatef, M., Duin, R. P. W., and Matas, J. (1998), On combining classifiers, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20, 226-239.

Leibniz System (1996), Version 4.2, Plano, Texas.

Maclin, R., and Optiz, D. (1997), An empirical evaluation of bagging and boosting, *Proceedings of the Fourteenth National Conference on Artificial Intelligence* (pp. 546-551), AAAI Press/MIT Press, Cambridge, MA.

Mangasarian, O. L., Setiono, R., and Wolberg, W. H. (1990), Pattern recognition via linear programming: Theory and application to medical diagnosis, in: *Large-scale numerical optimization* (pp. 22-30), SIAM Publications, Philadelphia.

Mangasarian, O. L., Street, W. N., and Wolberg, W. H. (1995), Breast cancer diagnosis and prognosis via linear programming, *Operations Research*, 43, 570-577.

McLachlan, G. J. (1992), *Discriminant Analysis and Statistical Pattern Recognition*, Wiley-Interscience, New York.

Michie, D., Spiegelhalter, D. J., and Taylor, C. C. (1994), *Machine Learning, Neural and Statistical Classification*, Ellis Horwood, New York.

Nadler, M., and Smith, P. E. (1993), *Pattern Recognition Engineering*, Wiley, New York.

Quinlan, J. R. (1992), *C4.5: Programs for Machine Learning*, Morgan Kaufmann, San Francisco, CA.

Quinlan, J. R. (1996), Bagging, boosting, and C4.5, *Proceedings of the Thirteenth National Conference on Artificial Intelligence* (pp. 725-730), AAAI Press/MIT Press, Cambridge, MA.

Schlimmer, J. C. (1987), Concept acquisition through representational adjustment, thesis, University of California at Irvine, Irvine, CA.

Sun, F. S. (1998), *Error Prediction in Data Mining*, thesis, University of Texas at Dallas, Richardson, Texas.

Triantaphyllou, E., Allen, L., Soyster, L., and Kumara, S. R. T. (1994), Generating logical expressions from positive and negative examples via a branch-and-bound approach, *Computers and Operations Research*, 21, 185-197.

Truemper, K. (1998), *Effective Logic Computation*, Wiley, New York.

White, H. (1992), *Artificial Neural Networks: Approximation and Learning Theory*, Blackwell Publishers, Cambridge, MA.

Wolpert, D. H. (1992), Stacked generalization, *Neural Networks*, 5, 241-259.