STATISTICS IN MEDICINE
Statist. Med. 2006; **25**:3787–3797
Published online 28 December 2005 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/sim.2488

Comparison of misclassification rates of search partition analysis and other classification methods

Roger J. Marshall*,†

Section of Epidemiology and Biostatistics, School of Population Health, Faculty of Medical and Health Sciences, University of Auckland, New Zealand

SUMMARY

Search partition analysis (SPAN) is a method to develop classification rules based on Boolean expressions. The performance of SPAN is compared against the trials reported by Lim *et al.* of 33 other methods of classification, including tree, neural network and regression methods on 16 data sets, most of which were health related. Each data set was augmented with noise variables in further trials. Lim *et al.* assessed the performance of the methods by estimates of misclassification rate, either cross-validated or test sample based. In this paper, the same data sets are analysed by SPAN and misclassification rates of the SPAN classifiers are estimated. Comparison is made of the performance of SPAN against the other methods that were considered by Lim *et al.* In terms of average misclassification error rate, taken over all data sets, SPAN was among the best five methods. In terms of average ranking of misclassification, that is, for each data set ranking the misclassification rates from lowest to highest, SPAN was second only to polyclass logistic regression. Copyright © 2005 John Wiley & Sons, Ltd.

KEY WORDS: classification; SPAN; tree; neural network; misclassification error

INTRODUCTION

There are, in the statistical and machine learning literature, numerous methods for the problem of classification, but there are relatively few systematic evaluations and comparisons of the methods. However, one rigorous comparative study is the STATLOG project [1]. Another, which extended STATLOG in various ways, and in particular by including tree methods, is a study by Lim *et al.* [2, 3] (henceforth LLS). These authors compared the performance of 33 methods for classification on 16 data sets. In addition a further 16 pseudo-data sets, comprising the original data but each augmented with noise, were used in the trials. LLS considered tree based methods, 'statistical' methods (regression, spline or discriminant analysis) and neural

*Correspondence to: R. J. Marshall, Section of Epidemiology and Biostatistics, School of Population Health, Faculty of Medical and Health Sciences, Private Bag 92019, University of Auckland, Auckland, North Island, New Zealand.
†E-mail: rj.marshall@auckland.ac.nz

Received 25 March 2004
Accepted 18 November 2005

networks. Their study provides useful benchmarks against which to compare other methods, for despite its broad scope and thoroughness, there are other known methods which were not included.

One is a procedure known as search partition analysis (SPAN) [4–7]. It is a method to develop diagnostic and prognostic classification rules based on Boolean expressions. Like most classification methods it has potential applications other than medical. Tree methods also produce classifiers in terms of Boolean logic formulae, but SPAN generates classifiers non-hierarchically. The approach avoids some of the conceptual problems inherent in tree classifiers [8].

In this paper, SPAN is compared against the performance of the 33 methods reported by LLS for 12 of their 16 data sets. All but 3 of the 12 are in some sense 'medical'. The measure of performance used in the comparisons was misclassification error rate. LSS concluded that a spline-based 'polyclass' algorithm [9] performed best, followed by logistic regression, though the performance of the best algorithms differed only marginally. The best of the tree algorithms was one allowing splits based on linear regression [10].

METHODS

The LLS data sets

Five of the LLS data sets concerned the two-state classification problem and a further seven were for classification to three states. The remaining were for classification to four or more nominal states and are not used in this study, because the SPAN method is not applicable to multi-state nominal outcomes. For each data set, an additional one had been artificially created by LLS by adding a number of additional noise variables and analysis was done both on the original and on the noise-augmented pseudo-data sets. The data sets, both with and without the noise variables, were downloaded from http://www.stat.wisc.edu/∼loh/quest.html and are listed in Table I. Further details are:

1. Breast cancer from Wisconsin (`bcw`). Observations on breast tissue to determine either malignancy or not. There were nine predictor measures, each measured on a 1 to 10 scale with 1 referring to a normal state and 10 most abnormal.
2. BUPA liver disorders (`bld`). A data set with six numerical predictors of the presence, or not, of a liver disorder.
3. Boston housing (`bos`). To predict house values in three ordinal classes. Although LLS used 10-fold cross validation, here data split into training and test samples of 400 and 106, respectively.
4. Contraceptive method (`cmc`). To predict contraceptive method use (no use, long-term methods, short term methods) from demographic and social characteristics.
5. DNA data (`dna`). To predict three-states, the boundaries between exons and introns, in DNA sequences.
6. Heart diseases (`hea`). The problem is to predict presence, or not, of heart disease. There were seven numerical and six categorical predictors.
7. Pima Indian diabetes (`pid`). To predict whether a female Pima Indian is positive or negative for diabetes based on seven numerical attributes.

Table I. Main features of data sets used in the comparisons. num = numeric variable (either continuous or integer), cat = categorical, bin = binary. N(0, 1) is standard normal variable, U(0, 1) = uniform on 0 to 1, and UI(n, m) is uniform over integers n to m, e.g. $9 \times$ UI(1, 10) means that nine integer variables, uniform over 1 to 10, were added. CV is cross-validation.

Data	# classes	N	# predictors	With added noise	Validation
bcw	2	683	9: each num	$9 \times$ UI(1, 10)	10-fold CV
bld	2	345	6 num	$9 \times$ N(0, 1)	10-fold CV
bos	3(ordinal)	506	14: 13 num, 1 bin	$12 \times$ N(0, 1)	Training 400, test 106
cmc	3(nominal)	1473	9: 3 bin, 3 cat, 3 num	$6 \times$ N(0, 1)	Training 1000, test 473
dna	3(nominal)	3186	60, each 4-state cat	$20 \times$ UI(1, 4)	Training 2000, test 1186
hea	2	270	13: 3 bin, 3 cat, 7 num	$7 \times$ N(0, 1)	10-fold CV
pid	2	532	7 num	$8 \times$ N(0, 1)	10-fold CV
smo	3(ordinal)	2855	8: 3 bin, 1 cat, 4 num	$7 \times$ N(0, 1)	Training 1855, test 1000
tae	3(ordinal)	151	5: 2 bin, 2 cat, 1 num	$5 \times$ N(0, 1)	Training 100, test 51
thy	3(ordinal)	7200	21: 15 bin, 6 num	$4 \times$ U(0, 1), $10 \times$ UI(1, 2)	Training 3772, test 3428
vot	2	435	16 each 3-state cat	$14 \times$ UI(1, 3)	10-fold CV
wav	3(nominal)	3600	21 num	$19 \times$ N(0, 1)	Training 600, test 3000

8. Smoking restrictions (smo). A three-state problem to predict attitudes to work place smoking (prohibited, restricted and unrestricted) from eight predictors.
9. Teaching evaluations (tae). To predict three ordinal classes of teaching performance. Two of the five predictors are categorical: course instructor (25 categories) and course type (26 categories).
10. Thyroid disease (thy). To predict three-states of thyroid function from twenty-one predictors.
11. Voting US Congressional (vot). To predict two-states: Democrat or Republican on the basis of voting. 16 categorical predictors, each with three-states.
12. Wave form (wav). Artificial data with three (nominal) classes data of waveform from twenty-one predictors.

The LLS methods

As mentioned, LLS considered 33 classification methods. The reader is referred to the LLS paper for full details of each of them. Broadly, LLS made a division into three types: statistical (9 methods), tree or rule based (22 methods), and neural networks (2 methods). By 'statistical', they meant regression-type methods: logistic regression, discriminant analysis and variants (linear, quadratic, mixture, nearest neighbour), and regression splines. The tree algorithms were all based on fitting trees by successive sub-division of the data. LLS used a number of different tree programs, including the well-known CART and S-plus algorithms, with different pruning schemes. Some of the methods they employed are really hybrid statistical/tree-based. The QUEST algorithm [10], for example, uses splits based on linear regression, and the 'fast classification tree algorithm' (FACT [11]) uses discriminant analysis to determine splits at tree nodes. Only two neural networks were used: one using a radial basis function SAS macro [12] and another using S-plus library routines (olvq1, lvq1) [13].

Statistical approaches can be regarded as 'arithmetic', in that the classification rule is based on an arithmetic output score, or scores, for example, a linear combination. Neural networks also output scores and are really very complex regression functions, so that they too could be regarded as arithmetic. In distinction, classifiers of tree-based methods could be termed 'logical', since the classification rules can be represented by Boolean logical expressions.

In medical contexts, logical rule based methods are often considered to be closer to clinical reasoning in the way they demarcate clusters of people by Boolean expressions. For example, the necessary conjunction of symptoms that indicate a certain diagnosis is closer to diagnostic reasoning than averaging out clinical features in an arithmetic score. Given the very different nature of the algorithms that LLS considered, and the rules that they produce, there seems to be little point in comparing the classification rules themselves, and LLS did not attempt to do so. Following LLS, this study focuses on comparison of misclassification rates. However, the issue of interpretability of the rules that result from the different methods is clearly important. Some remarks on interpretability are made in the Discussion.

SPAN

SPAN is a rule-based 'logical' method, though rules are not generated hierarchically and they cannot usually be easily represented by a tree [8]. SPAN was developed for the two-state classification problem, say states S and $\hat{}S$ (here $\hat{}S$ denotes *not* S), and is based on classifiers that are Boolean combinations of *attributes.* Details of the method are presented elsewhere [3–6]; here just a sketch is given. An attribute is a feature derived from predictor data, for example, 'systolic blood pressure over 130 mmHg'. A Boolean combination of attributes, A say, defines a binary partition $(\hat{}A, A)$ of the predictor space. The partition is found by a search for Boolean combinations that best discriminate between the states S and $\hat{}S$. Since the number of potential Boolean combinations is vast, restrictions are imposed on the extent of the search. The analyses were done mostly using the default search restriction parameters, as indicated below.

All analyses were done using SPAN software, a program written by the author in Visual Fortran for Windows platforms. A SPAN analysis begins by creating a set of attributes from data. Each attribute has a corresponding complement. Obviously where a predictor is binary, only one attribute (and its complement) is possible, otherwise different ways of creating attribute(s) from a variable are required. Attributes in a SPAN analysis are user specified. For a continuous predictor, x, an attribute would typically be $X = \{x > C\}$ for some cut-off C. Most tree growing packages (and apparently all those used by LLS) consider each datum of a numeric variable as a potential cut-off for making splits. SPAN, however, requires cut-offs specified in advance by the user. When a predictor variable is categorical, attributes are created of the form $X = \{x \in C\}$ for a combination of categories C, where again C is user specified.

For continuous predictors in the LLS data sets, possible cut-offs were the 5 per cent, 10 per cent,..., 95 per cent empirical percentiles. Cut-offs for discrete integer predictors with values < 10, were the integers themselves. When a categorical variables had four or fewer categories all possible combinations of categories were tried for C. In one data set (`tae`, Table I) there were two variables with more than twenty categories each, making complete enumeration impractical. These were dealt with by specifying separate 'dummy' binary attributes; one for each category.

SPAN requires specifying a measure of effectiveness of a binary partition. A split into A and $^{\wedge}A$ can be assessed by the reduction of an impurity measure $i()$ [5]

$$G = i(P_S) - P_A i(P_{S|A}) - P_{^{\wedge}A} i(P_{S|^{\wedge}A})$$

where P_A and $P_{^{\wedge}A}$ are the data proportions in A and $^{\wedge}A$, and P_S, $P_{S|A}$ and $P_{S|^{\wedge}A}$ are proportions of state S in the data as a whole, in A and in $^{\wedge}A$, respectively. SPAN software allows two different impurity measures $i()$: the Gini index of diversity and an entropy index $i(P) = -P\log P - (1-P)\log(1-P)$. For the analyses reported here entropy was used.

The measure G is used initially to obtain a filtered 'attribute set', the elements of which are subsequently used to create Boolean classifiers. Filtering is done first by finding the best attribute for each individual variable, that is, the C which maximizes G. The set of best attributes for each variable is itself ranked and the best m of them forms the attribute set $T_m = \{X_1, X_2, \ldots, X_m\}$. Each element of T_m should be a 'positive attribute' with respect to S [5, 7], that is, an indication of S rather than $^{\wedge}S$. This is ensured by, if necessary, swapping an element X for its complement $^{\wedge}X$ when association (phi-coefficient) is negative.

SPAN generated Boolean expressions are of the (normal disjunctive) form

$$A = K_1 \cup K_2 \cup \cdots \cup K_q \tag{1}$$

where K_i is the conjunction of p_i elements of T_m. For given q and p_i all possible expressions of the form of equation (1) are exhaustively generated by a combinatoric 'lock-and-key' algorithm [6]. To make the search feasible m, q, and p_i must be small. Typically $m \leqslant 12$ and $q \leqslant 2$, $p_1 \leqslant 2$ and $p_2 \leqslant 2$ (the SPAN defaults). The complement of (1), $^{\wedge}A$, can also be expressed in normal disjunctive form of q' terms

$$^{\wedge}A = J_1 \cup J_2 \cup \cdots \cup J_{q'} \tag{2}$$

where J_i is a conjunction of the *complements* of elements of T_m. Switching the roles of the $\cup$ and $\cap$ operators generates an alternative complementary partition which, when A and $^{\wedge}A$ are written in disjunctive form, has q' terms in A and q in $^{\wedge}A$ [4].

More complex partitions (with greater q and p_i) are built in an iterative procedure by which an attribute is created from the best partition on an initial search and is added to T_m for subsequent searches [5]. In doing so the partitions generated may only have marginal increase in G. To offset G against increasing partition complexity, partitions are complexity penalized using $G - \beta c$ for some penalty parameter β and complexity, defined from (1) and (2), by $c = q + q' - 1$ [5, 7]. The rationale is that a partition with complexity $c+1$ is only better than one of complexity c if the increase in G exceeds β. SPAN software has a procedure for setting β as a fraction (default 0.05) of the best G for complexity $c = 1$ partitions.

Three-state classifiers

SPAN can be used for classification to k ordinal states by doing separate analyses of the $k-1$ binary outcomes formed by collapsing into 'above' and 'below' each of the categories. However, for the resulting k-way partition to be coherent the individual binary partitions must be nested and the searches can be constrained to ensure this [14]. The procedure is here only used for the three-state, $k = 3$, data sets as none of the $k > 3$ class LLS data sets are ordinal.

Four of the $k = 3$ state sets are obviously ordinal (bos, smo, thy, and tae). The method is, however, used on the other three (cmc, dna, and wav) taking the assigned category codes 1, 2 and 3 as the ordered sequence.

Estimates of error rates

Error rates for two-state data sets were estimated by 10-fold cross-validation (Table I), as was done by LLS. Precisely the same ten sub-samples that LLS formed were used in the cross-validations. For the three-state data sets dna, wav, thy, and smo, LLS reported analyses based on data split in training and test samples and precisely the same test-training samples are used here. For the remaining three-state data sets (cmc, tae, bos) LLS used 10-fold cross-validation. Implementing cross-validation to estimate error rates for the three state SPAN classification procedure described above cannot be easily done (at least with existing software). Instead the data sets cmc, tae and bos were randomly split into a training and a test sample (see Table I) for estimation of misclassification.

RESULTS

Table II gives estimated error rates for the SPAN analyses. It also reports, for reference, the range of the error rates reported by LLS for the 33 different methods that they considered. The re-substitution error rates for SPAN analysis of the full data set are also given. These, as expected, are typically less than the cross-validated and test data estimates. Also given in Table II is the 'plurality' rule. This is the term that is used by LLS for the (Bayes) rule for predicting in the absence of any predictors. That is, putting everyone (or everything) in the most common category.

The SPAN misclassification estimates are all within the range of the estimates by other methods. The SPAN rule for the smo data set was actually worse than the plurality rule and in this instance the plurality rule is reported, since it would be irrational to use a classifier that is worse. In sixteen methods reported by LLS on the smo data the error rate equalled the plurality rule, that is, the tree methods pruned to the root node or the maximum predicted class probability was always for the most common category. Only one method was reported to fare better, marginally, then the plurality rule (0.304 *versus* 0.305).

Table III reports on SPAN *versus* seven of the 33 LLS methods—three 'statistical' (logistic regression, linear discriminate analysis, and the polyclass algorithm), three tree methods (CART, QUEST and S-plus) and a neural network (radial basis function). It gives the rank of each in terms of misclassification rate ($1 = \text{best}, 33 = \text{worst}$). Also the corresponding position at which the SPAN misclassification rate appears on this ranking schedule is given. The average rank of these seven methods, over all data sets, is also given. In terms of average rank, SPAN, with average 11.1 is only exceeded by the polyclass method [9] with rank 9.3, the method LLS identified as having lowest average rank overall. Of all the tree methods the lowest average rank is for QU0 (QUEST [10]), which allows linear combination splits, and has rank 11.8. The best of the neural network methods (radial basis function, RBF [12]) had rank 17.1. Table III also gives the mean error rate over the data sets. On this measure SPAN is among the best five methods.

Table II. Validated estimated error rates (either cross-validation or test sample) by SPAN and other methods. Median and range of 33 other methods are shown, from data reported in LLS [2, Tables 10–24]. + indicates data set with added noise.

Data	# classes	SPAN validated error	SPAN resubstitution error	LLS Range of 33 methods	'Plurality' error
bcw	2	0.035	0.026	0.03–0.09	0.350
bcw+		0.035	0.026	0.03–0.08	
bld	2	0.337	0.223	0.28–0.43	0.419
bld+		0.328	0.223	0.29–0.44	
bos	3	0.236	0.217	0.221–0.314	0.657
bos+		0.236	0.217	0.225–0.422	
cmc	3	0.449	0.435	0.43–0.60	0.573
cmc+		0.444	0.462	0.43–0.58	
dna	3	0.075	0.078	0.05–0.38	0.492
dna+		0.075	0.078	0.04–0.38	
hea	2	0.170	0.131	0.14–0.34	0.444
hea+		0.170	0.131	0.15–0.31	
pid	2	0.227	0.204	0.22–0.31	0.333
pid+		0.227	0.213	0.22–0.32	
smo	3	0.305	0.305	0.30–0.45	0.305
smo+		0.305	0.305	0.31–0.45	
tae	3	0.510	0.400	0.325–0.693	0.656
tae+		0.450	0.404	0.445–0.696	
thy	3	0.0134	0.0130	0.005–0.89	0.073
thy+		0.0134	0.0130	0.01–0.88	
vot	2	0.0435	0.0437	0.04–0.06	0.386
vot+		0.0435	0.0437	0.04–0.07	
wav	3	0.261	0.200	0.151–0.477	0.667
wav+		0.261	0.200	0.160–0.446	

DISCUSSION

It is unreasonable to expect that one classification method will be consistently better than others; some may perform well in some situations, and not so well in others. The reported results give some indication of the performance of different methods, but whether they can be generalized to really make claims about one method being 'better' than another is questionable. The data sets may not be representative of typical data sets, though it is hard to see how a representative 'sample' of data sets could be obtained. Nevertheless, LLS found that the polyclass algorithm [9], based on polytomous logistic regression with splines and tensor products, seemed to do rather better overall, as is indicated also in Table III. The error rates of SPAN are consistently in the middle or lower end of the range of the error rates that LLS reported and the average error rate is among the best five methods. In terms of ranking of error rate it performs as well or better than all tree methods, logistic regression and discriminant analysis variants, and is out-done only by the POL method.

SPAN therefore emerges as a method that warrants consideration. Besides giving good misclassification error rates it also has an advantage that its decision rules are usually more interpretable than those of other methods. Interpretability of decision rules is important, at least in medical problems, for as Ripley [15] points out, no one cares how ZIP code classification

Table III. Reported rank in LLS of seven selected methods and, for SPAN, the rank to which the SPAN misclassification rate is closest. (LDA = linear discriminate analysis, LOG = logistic regression, POL = polyclass logistic regression, IC0 = CART with 0-SE rule, QU0 = QUEST with 0-SE rule, ST0 = S-Plus with 0-SE rule, RBF = Radial basis function network) Fractional ranks (e.g. 18.5) indicates ties. Mean overall error rate is also given.

Data	POL*	SPAN†	QU0†	LOG*	LDA*	IC0†	RBF‡	ST0†
bcw	19	7	2.5	5.5	12	23	5.5	30
bcw+	11	8	1.5	8	9	27	4	20
bld	3	24	7	9	18.5	20	24	10
bld+	1	9	26	17	15	7	18	6
bos	2.5	6	28	11.5	14.5	18.5	2.5	22.5
bos+	3	2	22.5	13	20	11	28	17.5
cmc	1	5	14	19	22	7.5	10	9
cmc+	1	4	17	18.5	22.5	5	15.5	9
dna	2	21	18	16	12	10	31	10
dna+	3	20	19	17	13.5	6	31	6
hea	9	7.5	4.5	6	1	18	11.5	30
hea+	16	6	6	9	2	16	8	23
pid	16.5	9	5	11	1.5	16.5	11	16.5
pid+	6	6	2.5	7	4	13.5	15	30
smo	9.5	9.5	9.5	9.5	9.5	26	18.5	9.5
smo+	7.5	9.5	7.5	16.5	21	7.5	23.3	7.5
tae	20	19	10	11	6	3	13	30.5
tae+	16	11	9	4	2	20	10	32
thy	14.5	14.5	17	26	28	8	23	5.5
thy+	15	15	17.5	25	27	10	29	7.5
vot	25.5	9	1	21	15	17.5	25.5	21
vot+	16	5	21	5	16	26	21	19
wav	5	19	8.5	2	10.5	29	1	26
wav+	6	20	9	3.5	7.5	27.5	31	26
Mean Rank	9.3	11.1	11.8	12.1	12.9	15.6	17.1	17.7
Mean error	0.210	0.219	0.219	0.215	0.216	0.223	0.249	0.247

*Statistical.
†Tree-based or rule algorithm.
‡Neural network.

is done, as long as it works, but rules for medical diagnosis and prognosis should make sense. Interpretability, in particular in the context of medical problems, has driven the SPAN project [4, 8]. Trees tend to give rules that are counter intuitive [8] and statistical regression models create rules based on arithmetic scores which seem to have little in common with clinical reasoning. Neural networks are very complex regression functions [16]—so complex that it has been seriously suggested that you need to do a tree analysis of the predicted values from the analysis to make sense of a neural network classifier [17, 18]. The performance of these methods is not good anyway, as the LLS study demonstrates.

Since publication of the LLS report, other methods, besides SPAN, have been proposed. The most lauded is the method using 'support vector machines' (e.g. see Reference [19] for a readable account). These encompass the radial basis function (RBF) and three layer neural nets as special cases (which perform badly in the LLS study). Also the issue of interpretability, for medical problems, remains.

LLS also considered interpretability of decision rules to be an important criterion. However, they did not discuss the issue beyond reporting size of trees, that is, number of leaves, which gives some indication of the complexity of decision rules. But, even small trees can yield complex rules and, conversely, simple rules often need large trees to represent them [8]. SPAN tends to produces classification rules that are generally fairly simple and often easier to interpret than tree derived rules. For example, the `bcw` data, for which *elevated* values of the predictors are positive attributes for malignancy, gives the SPAN rule for classification to 'malignant' as

$$X \cup (Y \cap W) \cup (Y \cap Z)$$

where $X = \{\text{clump thickness} > 6\}$, $Y = \{\text{uniformity of cell shape} > 3\}$, $W = \{\text{single epithelial cell size} > 2\}$ and $Z = \{\text{bare nuclei} > 6\}$. Tree grown rules for the `bcw` data give classifiers that are conjunctions of elevated and depressed values of the predictor variables and such rules, which mix positive and negative attributes, are generally counter-intuitive [8]. The SPAN rule for malignancy will necessarily only involve elevated positive attributes, because only those are elements of the attribute set. However, tree based rules may occasionally *simplify* to rules that do not mix positive and negative attributes. For example, suppose a tree analysis for the `bcw` data had produced a tree with the rule to 'malignant' nodes

$$X \cup (\hat{}X \cap Y \cap W) \cup (\hat{}X \cap Y \cap \hat{}W \cap Z)$$

that is, a split first on X, with pathways $\hat{}X \cap Y \cap W$ and $\hat{}X \cap Y \cap \hat{}W \cap Z$ from the $\hat{}X$ branch. This rule, on simplifying the Boolean algebra, is identical to the SPAN rule above. Such simplifications seldom occur however [8] and, without the simplification, which is not usually done, the importance of the curious 'interactions', $\hat{}X \cap Y \cap W$ and $\hat{}X \cap Y \cap \hat{}W \cap Z$, would be questionable.

LLS did not, apparently, use variable selection methods in logistic regression or any of the other regression methods. Therefore, the issue of model parsimony was not addressed, though it seems important. For example, for the `dna` data there were 60 variables (80 with the additional 20 noise variables); it seems to warrant some sort of model reduction. SPAN has two mechanisms which implicitly filter only the important variables: first by pre-filtering to specify the pool of important attributes in T_m and making m not too big (up to 12). Second by restricting the parameters q and $p_1\, p_2 \ldots p_q$ in conjunction with complexity penalizing.

LLS also considered computation time and reported substantial differences. The polyclass algorithm [9], while having lowest error rates overall, was one of the most computationally intensive, requiring about 4 h, against 4 min for ordinary logistic regression (it was 1998!). Computation times are not reported here because SPAN is done interactively and time spent clicking a mouse and making other user–computer interactions make the time aspect difficult to measure. Once the actual search algorithm is set running, CPU time depends on the constraints imposed to limit search size. By selecting the 'top twelve' ($m = 12$) attributes for a search and limiting the $p_1 \ldots p_q$ parameters to 2, 2 usually enables the analysis to be accomplished in a matter of less than a minute.

Although the measure of performance used in these trials has been misclassification error, it is not necessarily the best measure. It makes no distinction between whether misclassifications are false positive or false negative, or costs of misclassification. Some programs allow for costs and cost minimization. SPAN, at present, does not.

With the exception of the wav data, these 'trials' have all been done on real data sets. The estimates of misclassification are precisely that—estimates. An alternative procedure [20] is to test models on simulated data, repeating the analysis on many sets of data generated by the same model to obtain stable estimates of performance. With this approach, Reference [20] demonstrated that logistic regression again outperformed trees and neural nets. It would be interesting to test SPAN in this way too but, as each SPAN analysis is done by pointing and clicking, it will require isolating the SPAN algorithm in, for example, an R function embedded in a data generation loop. It is hoped that this may be possible in the future.

Although SPAN performs well, it does have limitations: first, using it for $k>2$ ordinal states does not allow cross-validation, at least, with its present program structure and the ordinal procedure outlined in Reference [14] is messy for $k>3$. The software was basically written with binary partitions in mind. Second, unlike logistic regression and other statistical approaches, where an optimal cut-off for a score function can be chosen to offset false positives, false negatives, and costs of misclassification, the optimal SPAN partition cannot be so tweaked. But neither can tree-based classifiers. A third disadvantage, is that SPAN probably requires the user to specify more parameters of the analysis in advance than other approaches, in particular, the specification of cut-off for attributes of numeric predictors is left up to the user. It could be argued though that this is preferable, given that established cut-offs may exist. Finally, the software is written entirely by the author and it perhaps lacks the functionality of much corporate statistical software. SPAN can be downloaded from http://www.auckland.ac.nz/mch/span.

ACKNOWLEDGEMENTS

This work was mostly done in late 2003 while visiting the Institut Catala D'Oncologia, Barcelona. I would like to thank the Institut for hosting me and making my stay so pleasant. Special thanks to Dr Victor Moreno for discussions and interest in SPAN. I would also like to thank Professor Wei-Yin Loh (one of the LLS authors) for some helpful correspondence.

REFERENCES

1. Michie D, Spiegelhalter DJ, Taylor CC (eds). *Machine Learning, Neural and Statistical Classification*. Ellis Horwood: London, 1994.
2. Lim TS, Loh WY, Shih YS. A comparison of prediction accuracy, complexity and training time of thirty-three old and new classification algorithms. *Machine Learning* 2000; **40**:203–229.
3. Lim TS, Loh WY, Shih YS. An empirical comparison of decision trees and other classification methods. *Technical Report 979*, Department of Statistics, University of Wisconsin, 1998.
4. Marshall RJ. Partitioning methods for classification and decision making in medicine. *Statistics in Medicine* 1986; **5**:517–526.
5. Marshall RJ. A program to implement a search method for identification of clinical subgroups. *Statistics in Medicine* 1995; **14**:2645–2659.
6. Marshall RJ. Generation of Boolean classification rules. In *Proceedings of Computational Statistics 2000*, Utrecht, The Netherlands, Bethlehem, van der Heijden PGM (eds). Springer: Heidelberg, 2000; 355–360.
7. Marshall RJ. Determining and visualising at risk groups in case-control data. *Journal of Epidemiology and Biostatistics* 2001; **6**:343–348.
8. Marshall RJ. The use of classification and regression trees in clinical epidemiology. *Journal of Clinical Epidemiology* 2001; **54**:603–609.
9. Kooperberg C, Bose S, Stone CJ. Polychotomous regression. *Journal of the American Statistical Association* 1997; **92**:117–127.
10. Loh WY, Shih YS. Split selection methods for classification trees. *Statistica Sinica* 1997; **7**:815–840.
11. Loh WY, Vanichsetakul N. Tree-structured classification via generalized discriminant analysis (with Discussion). *Journal of the American Statistical Association* 1988; **83**:715–728.

12. Sarle WS. Neural networks and statistical models. *Proceedings of the Nineteenth Annual SAS Users Groups International Conference*. SAS Institute Inc.: Cary, NC, 1994; 1538–1550.
13. Venables WN, Ripley BD. *Modern Applied Statistics with S-Plus*. Springer: New York, 1997.
14. Marshall RJ. Classification to ordinal categories using a search partition methodology with an application in diabetes screening. *Statistics in Medicine* 1999; **18**:2723–2735.
15. Ripley BD. *Pattern Recognition and Neural Networks*. Cambridge University Press: Cambridge, MA, 1996.
16. Schwarzer G, Vach W, Schumacher M. On the misuses of artificial neural networks for prognostic and diagnostic classification in oncology. *Statistics in Medicine* 2000; **19**:541–561.
17. Faraggi D, LeBlanc M, Crowley J. Understanding neural networks using regression trees: an application to multiple myeloma survival data. *Statistics in Medicine* 2001; **20**:2965–2976.
18. Marshall RJ. Letter to editor. Understanding neural networks using regression trees: an application to multiple myeloma survival data. *Statistics in Medicine* 2003; **22**:661–662.
19. Hearst M. Support vector machines. *IEEE Intelligent Systems* 1998; **13**:18–21.
20. Terrin N, Schmid CH *et al.* External validity of predictive models: a comparison of logistic regression, classification trees, and neural networks. *Journal of Clinical Epidemiology* 2003; **56**:721–729.

