Pergamon

Computers in Human Behavior, Vol. 11, No. 2, pp. 273–288, 1995

0747-5632/95 $9.50 + .00

0747-5632(94)00036-0

Classification by Similarity: An Overview of Statistical Methods of Case-Based Reasoning

Dean P. McKenzie

Department of Psychological Medicine, Monash University

Richard S. Forsyth

Department of Mathematical Sciences, University of the West of England

Abstract — *There has recently been a great deal of interest in case-based reasoning, the generation of solutions to new problems using methods which have served for similar problems in the past. Much of the commonly available computer software is however concerned with "case-retrieval." The latter involves the matching of an observation for which the outcome is not known, to a database of examples for which the outcome is known. Various types of case retrieval, or "classification by similarity" (CBS), algorithms are discussed. Several CBS algorithms, as well as various other techniques, were applied to two small datasets. Although more comparisons are required, the CBS algorithms were found to perform significantly better than a linear discriminant analysis on a predominantly binary dataset. A single-nearest-neighbor technique, first developed in the 1950s, performed particularly well on this dataset. A more sophisticated CBS algorithm, based upon a type of neural network, performed consistently well on both datasets. As CBS techniques generally encourage the*

Requests for reprints should be addressed to: Dean McKenzie, Biostatistics and Psychometrics Unit, Mental Health Research Institute, Royal Park Hospital, Private Bag 3, P.O. Parkville, Victoria, Australia 3052. E-mail: deanm@vaxc.cc.monash.edu.au

researcher to work closely with databases, they should be developed further. Progress needs to be made in the identification of "good" subsets of classifier variables, as well as in bridging the gap between statistical techniques and artificial intelligence.

On every great occurrence I endeavoured to discover in past history the event that most nearly resembled it. I procured, whenever it was possible, the contemporary historians, memorialists, and pamphleteers. Then fairly subtracting the points of difference from those of likeness, as the balance favoured the former or the latter, I conjectured that the result would be the same or different. As, for instance, in the series of essays entitled *A Comparison of France under Napoleon with Rome under the first Caesars*, and in those which followed, "on the probable final restoration of the Bourbons." The same plan I pursued at the commencement of the Spanish Revolution, and with the same success.

Samuel Taylor Coleridge, *Biographia Literaria*, p. 214.

When confronted with a new problem, for which the solution is not known, humans generally search for similar problems that they have been able to solve in the past (Rich & Knight, 1991). This process, known to the artificial intelligence/cognitive science community as case-based reasoning (e.g., Riesbeck & Schank, 1989; Slade, 1991), has recently been the focus of much attention (e.g., Barletta, 1991; Eliot, 1992; Lang, 1990).

This article is concerned with established and novel nonparametric statistical techniques for classifying observations. Classification is made on the basis of similarity to other observations, for whom the outcome is known. The techniques are then applied to two small datasets and compared with a variety of other algorithms, including conventional discriminant function analysis. The emphasis will be on the techniques themselves, as well as their application to behavioral data. Specific software implementations will, however, be referred to when necessary. Finally, a list of vendors of software will be provided, in order to facilitate the use of relevant programs by interested researchers.

CASE-BASED REASONING

Case-based reasoning (CBR) involves case storage, indexing, and adaptation (the modifying of existing information in order to be able to deal with new problems), as well as case retrieval. Slade (1991) sees case storage or case memory as being "the database of experience" (p. 45), and relates this type of storage to Tulving's (1972) concept of episodic memory. Riesbeck and Schank (1989) compare case retrieval with looking up a table of logarithms (in case storage). Case adaptation is compared with the interpolation of table entries when an exact match is not possible.

CLASSIFICATION BY SIMILARITY

As well as being an integral part of CBR algorithms, case retrieval forms the basis for both established and novel nonparametric statistical methods, to be discussed

below. We shall use the term "classification by similarity" (CBS), to encompass statistical methods as well as CBR.

With regard to practical applications, Lang (1990) mentions that a commercial CBS system known as The Outcome Advisor (Patrick, 1990) has been used with burns, choking, and heart disease victims. The program has also been used in the identification of low reading ability students and illiterate adults. Rissland and Skalak (1989) describe TAX-HYPO, a system used with taxation laws. As TAX-HYPO is able to perform case adaptation and learn from experience, it is an example of a true CBR algorithm.

CBS techniques can be applied using existing statistical packages, or special purpose programs such as those reviewed by Dwinnell (1994). Both established and novel CBS algorithms are grounded in statistical techniques, which have been widely known for several decades.

COMPUTERIZED CLASSIFICATION SYSTEMS

Computerized classification systems (Hale & Glassman, 1986; McKenzie & Low, 1992) can easily be built around CBS algorithms. A synthesis between human experience, as codified in an expert system (e.g., Rich & Knight, 1991), and CBS algorithms, is embodied in Rissland and Skalak's (1989) TAX-HYPO mentioned earlier. TAX-HYPO is able to employ CBR if the necessary conditions of expert system rules are not met.

Unlike inductive methods such as decision rules and trees, which present information in a readily understood form, CBS algorithms may not give very much insight into relationships between variables (Breiman, Freidman, Olshen, & Stone, 1984). It could, however, be argued that decision rules and trees are abstractions that allow the researcher to remain distant from the data (Panel on Discriminant Analysis, Classification & Clustering, 1989). CBS methods on the other hand, cannot function independently of a database. If the researcher is familiar with the information in the database, the identification of patients, or students, or legal cases similar to the target case may generate alternative hypotheses. The researcher may remember information about the retrieved cases that was not explicitly codified in the database.

Although the constant need for searching of databases may require a great deal of computer processing and storage space, various methods of indexing databases and removing redundant cases have been proposed. CBS methods may be of use when databases are constantly being added to, as is often the case in educational, medical, and other clinical contexts. It is easier to update a CBS system, than it is to update a tree or rule-based system, as new information is added to a database (Aha, 1992). Finally, CBS methods may be of use when variables, required by an inductive decision tree or rule, are missing for a particular observation.

Inductive algorithms are able to handle missing data to some extent. Disjunction (the logical ORing of similar variables, Breiman et al., 1984; Weiss & Kulikowski, 1991), or "surrogate" variables (Breiman et al., 1984) may be able to be used. Information may still be missing for these alternative variables, however. CBS methods will try to classify on the basis of the information that is available (Teranet IA, 1992).

NEAREST-NEIGHBOR METHODS

An algorithm for finding matches to a target observation was first described by Fix and Hodges (1951). The simplest embodiment of the nearest neighbor (NN) technique is known as the 1-NN rule. The outcome of a target record is taken to be the outcome for the single closest case or observation in a database. Closeness is often assessed in terms of the straight-line or Euclidean distance.

An extension of the 1-NN technique is known as the K-NN rule (McLachlan, 1992). The latter involves assigning a target to the outcome for which a majority of its K (generally an odd number) nearest neighbors belong to. The BASIC source code for a simple 1-NN algorithm is provided by James (1985), while 1-NN and K-NN rules are available in IMSL (IMSL, 1987) and SAS (SAS Institute, 1989).

K-NN algorithms usually require the value of K to be chosen by the researcher. However, a procedure described by Luk and MacLeod (1986) automatically chooses a value of K, subject to two conditions defined by the researcher. Nearest neighbors are chosen until the number of members of the most typical group exceeds membership of other groups by a specified majority, or until a specified maximum value of K is reached.

A similar, although more sophisticated, algorithm is used by the ModelWare microcomputer program (Teranet IA, 1992). ModelWare automatically chooses the value of K for each target observation to be matched. Cases are then weighted on the basis of their similarity to the target.

An extended version of the program, ModelWare Professional, allows the researcher to choose various methods of weighting observations. A minimum level of similarity between target observations and cases is also able to be defined.

DISTANCE METRICS

Distances other than the Euclidean distance, which can be used with numeric and symbolic information, have been discussed by several authors (Rothman, 1992; Cost & Salzberg, 1993). The ModelWare program described above incorporates a proprietary distance measure. This measure is supposedly more resistant to outliers than is the Euclidean distance (Teranet IA, 1992).

NEURAL NETWORKS

Dwinnell (1994) views CBS algorithms such as ModelWare as being competitors to neural networks. The latter have been examined within a cognitive psychological framework by Martindale (1991), and within a clinical psychological framework by Tyron (1993). One of the most popular neural network algorithms is known as back-propagation (BP). BP neural networks iteratively adjust the activation of the "neurons," "nodes," or "neurodes" (Caudill & Butler, 1992) in the network. This process is repeated until classification performance converges (Nelson & Illingworth, 1991). BP networks are generally not accepted as being biologically plausible (Forsyth, 1990; Nelson & Illingworth, 1991) and can be difficult to interpret (Carson, 1991). Moreover, BP can take weeks, months, or even years to converge upon a solution (Weiss & Kulikowski, 1991).

PROBABILISTIC NEURAL NETWORKS

Specht (1990) describes a type of neural network known as the probabilistic neural network (PNN). Although PNN's are not intended to be biologically plausible, they have been found to be several orders of magnitude faster than BP networks (Specht & Shapiro, 1991). The architecture of a PNN is shown in Figure 1. The first, or input layer of the network simply contains the standardized values of the classifier variables for the target observation to be classified. There are as many neurodes in the input layer as there are classifier variables.

The second, or pattern layer consists of the summed standardized values for each case in a reference library or training set. These values are weighted by an exponential activation function. The latter utilizes a smoothing parameter that is generally determined by trial and error. There are as many neurodes in the pattern layer as there are cases.

The third, or summation layer of the network consists of the sums of the values in the pattern layer, for each outcome or group separately. The third and fourth layers consist of one neurode for each outcome. Each neurode in the fourth, or output layer represents the probability of membership for each outcome, based upon the values of the summation layer. The target observation is assigned to the outcome that has the highest probability of membership. Near neighbors of a target will have high weights in the pattern layer.

PNN algorithms are implemented in commercial neural network packages such as NeuroWindows (Ward Systems Group, 1992). Caudill and Butler (1992)

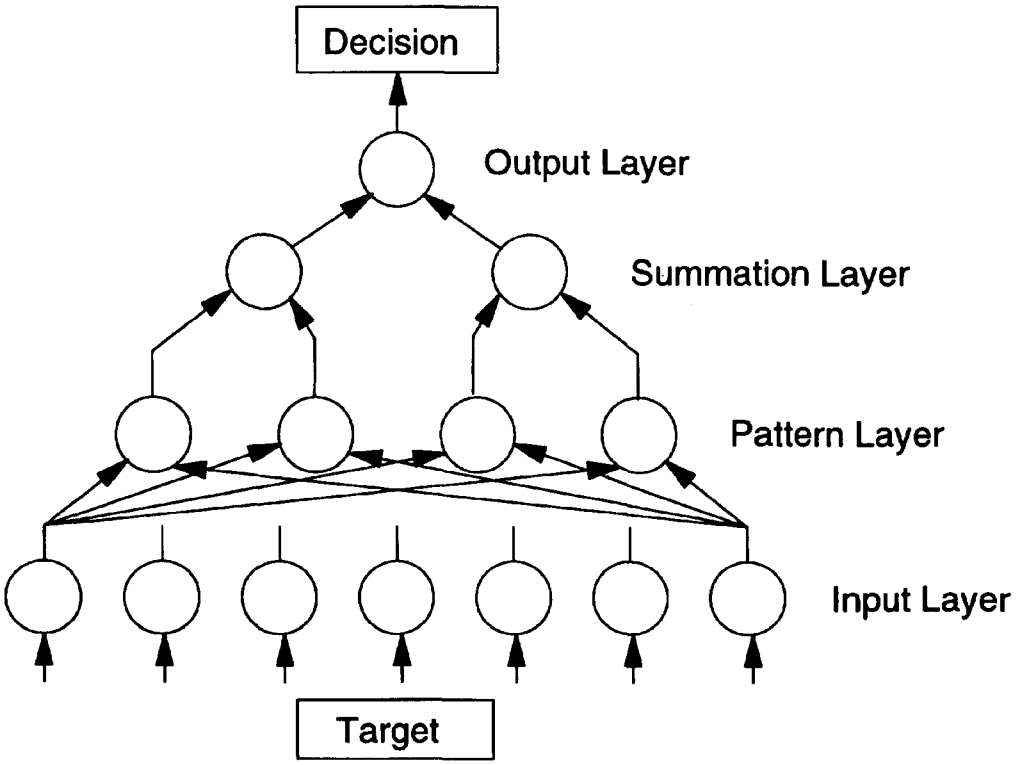

Figure 1. A probabilistic neural network showing connections between neurodes.

demonstrate how PNN's can be created manually or with the use of spreadsheet software.

ACCURACY OF CBS METHODS

Forsyth (1990) found that a simple 1-NN rule gave better results, when applied to a validation sample, than did BP neural networks. However, Weiss and Kulikowski (1991) observed that back propagation neural networks and CART, a popular decision tree generator (Breiman et al., 1984) gave better results than 1-NN rules. The performance of K-NN rules seemed to improve as the value of K was increased however.

Bounds, Lloyd, and Mathew (1990) noted that a K-NN rule, with the value of K being determined empirically, performed nearly as well as various neural network algorithms at diagnosing lower back pain disorders. The K-NN rule performed as well as, or better than, clinicians at diagnosing most types of lower back pain. Finally, Ripley (1994) found that nearest neighbor algorithms performed about as well as various neural networks, which in turn performed slightly better than conventional linear discriminant analysis.

With regards to more sophisticated nearest neighbor algorithms, ModelWare (Teranet IA, 1992) has been found to equal or exceed the accuracy of back propagation neural networks (Hess Consulting, 1992), as have the procedures of Aha (1992) and Cost and Salzberg (1993). The classification performance of the latter two procedures also matched or exceeded that of decision trees. Cost and Salzberg (1993) argue that no machine learning algorithm has been shown to consistently outperform nearest neighbor algorithms.

CBS methods have much faster training times than do BP neural networks. Hess Consulting (1992) found that a 5-NN (5 nearest neighbors) algorithm, and ModelWare, were generally 10 to 100 times faster than a BP neural network. As mentioned earlier PNN's are many times faster than BP networks, although their classification accuracy is comparable (Specht & Shapiro, 1991). Specht (1990) discusses an analysis that took a PNN 0.7 s on an IBM AT microcomputer, with slightly higher accuracy than a BP network that had been trained over an entire weekend.

EXPEDITING SEARCH AND RETRIEVAL

CBS methods require the case library or database to be held on-line. Classification of new cases may therefore be a slow process, especially for large databases (Devijer & Kittler, 1982; Weiss & Kulikwoski, 1991). Redundant cases can, however, be removed altogether (Aha, 1992; Devijer & Kittler, 1982; Teranet IA, 1992), or merged into clusters (Kaufman & Rousseeuw, 1990; Panel On Discriminant Analysis, Classification & Clustering, 1989; Specht, 1992).

A clustering procedure based on the Learning Vector Quantizer (LVQ) technique (Kohonen, 1988), was used in conjunction with the PNN algorithm, by Burrascano (1991). An implementation of the LVQ technique is available in NeuroWindows (Ward Systems Group, 1992). Although most clustering algorithms require the number of clusters to be decided upon by the researcher, an automatic clustering procedure was described by Patrick (1991). Finally, algorithms that quickly search the database for nearest neighbors are discussed in Weiss and Kulikowski (1991) and McLachlan (1992).

ESTIMATION OF ERROR RATE

If sample sizes were infinite, the 1-NN rule would give an error rate that is twice that of the true or Bayesian error rate (Cover & Hart, 1967). As K approaches infinity, the error rate given by a K-NN rule approaches that of the Bayes error rate. Most nearest neighbor implementations use a leave-one-out error estimate (the estimate of error does not include matching a case with itself).

A more sophisticated algorithm for error estimation, based on a modification of the bootstrap technique (Efron, 1983) was described by Weiss (1991).

SELECTION OF CLASSIFIER VARIABLES

The performance of CBS algorithms deteriorates when there are many poor classifier variables (Aha, 1992; Weiss & Kulikwoski, 1991). Unfortunately, there are few CBS algorithms that are specifically concerned with the selection of "good" variables.

Barletta (1991) and Ripley (1994) suggest that a tree-building algorithm such as CART (see also Forsyth, 1992; McKenzie & Low, 1992; McKenzie, McGorry, Wallace, Low, Copolov, & Singh, 1993) could be used to find salient classifier variables. A CBS system could be built around a decision tree or rule. Cases that meet the criteria represented in the branches of a tree, or the segments of a rule, could be identified by the system.

Although this is not to undermine the usefulness of inductive methods, the information needed by a decision tree or rule to make a classification may be missing for a particular observation. Provided that a fast search algorithm was used, a sequential search or stepwise technique (McLachlan, 1992) could be used in conjunction with the CBS algorithm to find a subset of salient variables.

The professional version of ModelWare (Teranet IA, 1992) is able to select subsets of classifier variables using a form of backward elimination, the latter being discussed by McLachlan (1992). At each iteration of the procedure, variables are eliminated if they lead to a decrease in average error.

An algorithm for fast variable selection for nearest neighbor classification, built around a genetic algorithm (e.g., Forsyth, 1989), was described by Kelly and Davis (1991). Aha (1992) presents an algorithm that is able to weight the relevance of each classifier variable automatically. Finally, Specht (1992) describes a method of variable selection for PNN. The latter involves the iterative selection of a separate smoothing parameter for each variable. Whichever method of variable selection is used, care must be taken to discard only variables with poor performance. Discarding variables that are as good as the ones retained is a common problem in the selection of variables by computer programs (Hauck & Miike, 1991).

A CBS EXAMPLE

A variety of CBS algorithms was applied to a zoological and a medical dataset. The first dataset was described by Forsyth (1990) and contains information on 101 animal species, randomly divided into learning (n = 57) and validation (n = 44) samples. There were 16 classifier variables, most of which were binary (such as the presence or absence of feathers, fins, and hair). There were seven outcome categories:

amphibian, bird, fish, insect, mammal, reptile, and other. This dataset was selected because the cases should have clear meaning for most people.

The second dataset was described by Afifi and Azen (1979) and contains information on 113 patients who had experienced cardiac failure. The dataset was randomly divided into a learning sample of 68 cases and a validation sample of 45 cases. There were 18 classifier variables, most of which were measured on interval or ratio scales, such as red cell counts, heart rate, and hemaglobin levels. There were two outcome categories, concerned with whether or not the patients survived. This dataset was chosen as an example of a practical application of CBS methods.

The CBS algorithms studied consisted of the simple 1-NN algorithm of James (1985), ModelWare Professional 2.0 (MW) (Teranet IA, 1992) and the PNN algorithm of Specht (1990). The latter algorithm is implemented in the NeuroWindows 3.1 (Ward Systems Group, 1992) program.

For purposes of comparison, a BP neural network generator, a conventional stepwise linear discriminant analysis (LDF) program, and two decision tree generators were chosen.

Brainmaker, a commercial BP program (California Scientific Software, 1989), was used. The median accuracy of five runs was reported. LDF was performed using the 7M subprogram of the BMDP statistical package (Jennrich & Sampson, 1988). Stepwise variable selection was used, with F values required to be at least as high as 4.0 (the default), in order for a variable to be entered into the discriminant function. LDF is generally used with classifier variables that are normally distributed (Tabachnick & Fidell, 1989). In the case of the medical data, LDF was therefore applied to both the full set of variables, as well as a subset of variables. The latter variables were not measured on a nominal or categorical scale.

The KnowledgeSEEKER (KS) program (FirstMark Technologies, 1990) was used with a .05 (KS_1) and a more conservative .01 (KS_2) stopping rule. In other words, the procedure terminated if the significance of a variable chosen to split the data into subgroups was not significant at the .05 or .01 levels. These levels of significance were automatically adjusted for the number of comparisons between category values. More details of the KS algorithm can be found in McKenzie and Low (1992).

CART (Breiman et al., 1984) was used with default values. The learning sample was divided into ten equally sized groups. Each group in turn was excluded from the analysis. A tree was then constructed using the remaining nine groups, and the procedure repeated. The tree selected was the smallest one with a classification accuracy that was within one standard error of the accuracy of the largest tree grown. More details can be found in Breiman et al. (1984). CART was not able to generate a decision tree for the medical dataset. A similar tree-building program, described by Forsyth (1992) was therefore used for this data. This program, Treemin, is based upon information theory (e.g. Attneave, 1959), and is currently restricted to binary outcomes. Treemin could, therefore, not be used with the zoological dataset.

MW was used in both 1-NN (MW_1) and K-NN (MW_2) modes. In the latter mode the value of K was automatically chosen by the program. "Drivers," a variable selection procedure available in ModelWare Professional, was also used, using the default settings (MWD). Three arbitarily chosen smoothing parameters were used with PNN, these being .0090 (PNN_1), .50 (PNN_2) and .99 (PNN_3).

RESULTS

As can be seen in Table 1, all algorithms apart from 1-NN, KS_1 , LDF, and MWD were able to classify all members of the zoological learning sample correctly. In the case of the zoological validation sample, 1-NN exhibited the best performance, closely followed by PNN_2 and PNN_3, while PNN_1 showed the worst performance. This illustrates the importance of choosing a satisfactory smoothing parameter. There was no difference in classification accuracy between MW_1, MW_2 and MWD.

With regards to the number of classifier variables selected, KS_2 chose four variables, KS_1 and CART chose five, although MWD and LDF selected six variables. The latter technique performed more poorly than all the other algorithms apart from PNN_1. MW_1, MW_2, MWD, PNN_2 and PNN_3 performed significantly better than LDF at the .05 level (1-tailed binomial test used throughout), while 1-NN performed significantly better than LDF at the .01 level.

There was no significant difference between KS_2 and the CBS algorithms apart from 1-NN, which performed significantly better than the former algorithm ($p < .05$). There were no significant differences between KS_1 and the CBS algorithms. 1-NN performed significantly ($p < .05$) better than KS_2 on the zoological validation sample, but did not perform significantly better than KS_1. There were no significant differences between KS_2 and the other CBS algorithms.

None of the methods were able to classify the reptiles in the zoological validation sample correctly, as no reptiles appeared in the learning sample. Reptiles tended to be misclassified as amphibians and fish.

In the case of the medical learning sample, only PNN_1 and BP were able to achieve perfect accuracy, as can be seen in Table 2. The simple 1-NN algorithm was able to classify less than half of the targets correctly. When the latter algorithm was applied to the z scores of the classifier variables however, the classification accuracy rose to 62%.

With regards to the medical validation dataset there was a significant ($p < .05$) difference between the performance of PNN_3 and LDF. LDF only chose one variable, however, whereas KS_2 chose two variables and KS_1 and MWD chose three. The Treemin tree-pruning procedure of Forsyth (1992), used because CART was not able to generate a tree, selected four variables.

Table 1. Classification Accuracy of CBS Algorithms Applied to Zoological Data

Method	Number of Variables	Learning Sample (n = 57) % Accuracy	Validation Sample (n = 44) % Accuracy
1-NN	16	95	89
BP	16	100	80
CART	5	100	80
KS_1	5	97	82
KS_2	4	100	73
LDF	6	95	71
MW_1	16	100	82
MW_2	16	88	82
MWD	6	100	82
PNN_1	16	100	61
PNN_2	16	100	82
PNN_3	16	100	82

Table 2. Classification Accuracy of CBS Algorithms Applied to Medical Data

Method	Number of Variables	Learning Sample (n = 68) % Accuracy	Validation Sample (n = 45) % Accuracy
1-NN*	18	47	62
BP	18	100	62
Treemin†	4	84	62
KS_1	3	84	58
KS_2	2	82	69
LDF‡	1	65	67
MW_1	18	63	58
MW_2	18	72	64
MWD	3	72	58
PNN_1	18	100	51
PNN_2	18	90	73
PNN_3	18	69	80

*When 1-NN was applied to the z scores of the learning and validation samples, the classification accuracy was 62% and 60%, respectively.
†As CART was not able to construct a tree using the default criteria, the backward pruning Treemin algorithm of Forsyth (1992) was used.
‡When LDF was used with the 14 noncategorical variables, classification performance increased to 67% and 78% for the learning and validation samples, respectively.

When LDF was allowed to choose from all 18 variables, the difference between PNN_3 and LDF was still significant. However, when only the 14 noncategorical variables were used, the classification performance of LDF rose to 78%. This performance was significantly higher ($p < .05$) than that of 1-NN, MW_1, and MWD.

PNN_3 performed significantly ($p < .05$) better than 1-NN, BP, KS_1, MW_1, and MWD. Use of the IMSL (IMSL, 1987) NNBRD subroutine indicated that 3-NN, 5-NN, 7-NN, and 9-NN rules performed no differently than a 1-NN rule, with regards to classification of the medical validation sample.

ANALYSIS OF CLOSEST MATCHES

Table 3 shows the closest matches to several observations in the zoological validation sample. These matches were found in the zoological learning or reference sample by MW_2. Proprietary similarity ("Similarity") measures and weighting coefficients ("Weight"), as well as the raw number of matching cases, are also given. MW_2 correctly classified a porpoise as a mammal, with the closest matches being a dolphin (an identical match) and a mink. The weighting coefficient for the latter is zero however, indicating that it is not needed in the classification.

None of the algorithms were able to classify reptiles such as the pitviper correctly, as there were no reptiles in the learning sample. The prediction formula used by MW_2 generated a value of four ("fish") for both the pitviper and the scorpion. A better strategy may have been to assign the targets to the most common species amongst the relevant matches. This would mean that the pitviper would be classified as an amphibian and a scorpion as an insect, which is surely a closer species than is a fish! A version of ModelWare that has separate modules for categorical and dimensional outcomes is currently under development (Teranet IA, personal communication; August 1994).

Table 3. Closest Matches Selected by ModelWare Algorithm

Closest Matches	Similarity	Weight	Matches	Species
Target = Pitviper (reptile); Cases selected = 4; Similarity = 0.864				
1 = Newt	0.917	–0.077	13	Amphibian
2 = Frog	0.917	0.500	13	Amphibian
3 = Kiwi	0.894	0.373	12	Bird
4 = Catfish	0.875	0.875	12	Fish
Target incorrectly classified as a Fish				
Target = Porpoise (mammal); Cases selected = 2; Similarity = 1.000				
1 = Dolphin	1.000	1.000	16	Mammal
2 = Mink	0.917	0.000	13	Mammal
Target incorrectly classified as a Mammal				
Target = Scorpion (other); Cases selected = 6; Similarity = 0.787				
1 = Ladybird	0.863	0.188	11	Insect
2 = Flea	0.863	0.188	11	Insect
3 = Kiwi	0.848	0.155	11	Bird
4 = Mole	0.823	0.273	10	Mammal
5 = Frog	0.823	0.053	10	Amphibian
6 = Octopus	0.812	0.143	11	Other
Target incorrectly classified as a Fish				

The ModelWare manual (Teranet IA, 1992) suggests using a single nearest neighbor approach in situations where the predicted value may not be meaningful with relation to the outcome categories. MW_1 matched the pitviper with a frog (amphibian) and the scorpion with a flea (insect). The scorpion was able to be correctly classified by the simple 1-NN algorithm, however, although MW_1 fared no better than MW_2 with regards to the classification ofthe zoological validation sample.

The matches described above were also found to have the strongest activation in the pattern layer of the PNN_3 neural network.

DISCUSSION AND CONCLUSIONS

Forsyth (1990) points out that nearest neighbor methods often serve as placeboes with regards to comparisons of classification algorithms. The simple 1-NN algorithm, first developed in the 1950s (Fix & Hodges, 1951), showed poor classification performance with relation to the hospital learning sample. However, with regards to the two validation datasets, the 1-NN algorithm showed comparable or superior performance to the other algorithms.

The performance of 1-NN was only significantly exceeded by PNN_3, and by LDF using the 14 noncategorical variables. These results indicate that care should be exercised in selecting variables for analysis by LDF. There was no significant difference in performance between the simple 1-NN and the current version of the more sophisticated ModelWare program.

Although far from conclusive, the present results indicate that use of the ModelWare "Drivers" variable selection algorithm may lead only to increased performance for the learning sample, and not for the validation sample. The computer-intensive error estimation procedure of Weiss (1991), or one of the cross-validation

procedures discussed by Weiss and Kulikowski (1991), could be incorporated into "Drivers." Alternatively, a more expedient method based on information theory (Patrick, 1991; Forsyth, 1992) could be utilized. Decreases in classification error for each set of variables could also be ascertained using McNemar's test. A similar strategy was suggested by Tabachnick and Fidell (1989), with relation to stepwise LDF.

Provided that a suitable smoothing parameter was chosen, PNN performed consistently well. A procedure for choosing smoothing parameters for PNN, based upon the use of a validation dataset, is included in the NeuroShell 2 microcomputer program (Ward Systems Group, personal communication; April 1993). Alternatively, a procedure such as the one described by Jain and Ramaswami (1988) could be utilized to help choose a smoothing parameter.

A more sophisticated version of PNN, which does not require the use of a smoothing parameter, was described by Musavi, Kalantri, Ahmed, and Chan (1993). This version has greater storage and processing requirements than the standard algorithm, but appears to exhibit higher classification accuracy. Similarly, a PNN algorithm that chooses a separate smoothing parameter for each variable (Specht, 1992), has greater processing requirements than the standard algorithm. Whichever version of PNN is used, a variable selection procedure (e.g., McLachlan, 1992) could also be incorporated.

The above suggestions could also be incorporated in an extended version of a simple K-NN algorithm. A value of K that is constant across the entire sample could be selected using Jain and Ramaswami's (1988) procedure. K could also be allowed to vary across targets using the algorithm of Luk and MacLeod (1986).

The latter algorithm could be extended to choose only those cases whose similarity to the target is equal to or greater than a specified value, although experimentation may be required in order to choose such a value. Distance measures, case clustering and database searching methods such as the ones mentioned above should also be included. Although not used in the present study, the nearest neighbor algorithms of Aha (1992) and Cost and Salzberg (1993) seem worthy of further attention.

With regard to the non-CBS procedures used, the performance of the BP algorithm on the present datasets does not seem to justify the amount of computer time needed to construct the networks. This finding lends support to the results of other studies (e.g., Forsyth, 1990; Specht & Shapiro, 1990; 1991).

The tree-building algorithms performed similarly to the other methods, and so may be able to be used to construct indices for CBS systems, as was suggested by Barletta (1991). Although decision trees themselves are easily understood, they may encourage the distancing of the researcher from the raw data. Case retrieval could, therefore, be used in conjunction with decision trees.

There may well be situations when the information required by decision trees and rules is missing for one or more observations. In such situations, case retrieval techniques would be required to search the database for close matches of a target observation, using available information.

Rather than being based upon abstract formulae, rules, or trees that are determined at a single point in time, CBS systems can easily be updated as new information is added to a database. As mentioned earlier, constant updating of information is common in medical and other clinical applications. However, CBS algorithms need to incorporate variable selection techniques. The researcher can then determine which classifier variables are more important than others. Furthermore, simple, as well as sophisticated, CBS techniques should be able to incorporate codified

human knowledge and experience. Generic case adaptation rules (e.g., Rich & Knight, 1991) should also be able to be incorporated.

CBS systems encourage, if not force, the researcher to closely examine the dataset. The researcher can then bring his or her practical experience and theoretical knowledge to bear on the particular problem. New, as well as unique, knowledge can then be codified and added to the CBS system. This allows for a constant "dialogue" between data and theory.

A list of known suppliers of commercial CBS software is given in the appendix.

Acknowledgment — The authors would like to thank Paul Low and Maria Posterino for their assistance with the preparation of the manuscript. The authors also acknowledge the constructive comments of the two anonymous referees.

REFERENCES

Afifi, A. A., & Azen, S. P. (1979). *Statistical analysis: A computer oriented approach* (2nd ed.). New York: Academic Press.

Aha, D. W. (1992). Tolerating noisy, irrelevant and novel attributes in instance-based learning algorithms. *International Journal of Man-Machine Studies*, **36**, 267–287.

Attneave, F. (1959). Applications of information theory to psychology. New York: Holt.

Barletta, R. (1991). An introduction to case-based reasoning. *AI Expert*, **6**, 43–49.

Bounds, D. G., Lloyd, P. J., & Mathew, B. G. (1990). A comparison of neural network and other pattern recognition approaches to the diagnosis of low back disorders. *Neural Networks*, **3**, 583–591.

Breiman, L., Freidman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees. Belmont*, CA: Wadsworth.

Burrascano, P. (1991). Learning vector quantization for the probabilistic neural network. *IEEE Transactions on Neural Networks*, **4**, 458–461.

California Scientific Software (1989). *BrainMaker user guide and manual* [computer program] (3rd ed.). Sierra Madre, CA: Author.

Carson, G. D. (1991). Interpreting neural network connection weights. *AI Expert*, **6**, 47–51.

Caudill, M., & Butler, C. (1992). *Understanding neural networks: Computer explorations* (Vol. 2). Cambridge, MA: MIT Press.

Coleridge, S. T. (1906). *Biographia literaria* (Vol. 1). London: Dent. [first edition 1817].

Cost, S., & Salzberg, S. (1993). A weighted nearest neighbor algorithm for learning with symbolic features. *Machine Learning*, **10**, 57–78.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, **IT13**, 21–27.

Devijer, P. A., & Kittler, J. (1982). *Pattern recognition: A statistical approach.* Englewood Cliffs, NJ: Prentice-Hall.

Dwinnell, W. (1994). Advanced modelling systems. *AI Expert*, **9**, 38–41.

Efron, B. (1983). Estimating the error rate of a prediction rule: Improvement on cross-validation. *Journal of the American Statistical Association*, **78**, 316–331.

Eliot, L. B. (1992). Appraising case-based reasoning. *AI Expert*, **7**, 9–10.

FirstMark Technologies. (1990). *KnowledgeSEEKER user's guide* [computer program]. Ottawa, Canada: Author.

Fix, E., & Hodges, J. L. (1951). *Discriminatory analysis, nonparametric discrimination: Consistency properties, Report 4.* Randolph Field, TX: USAF School of Aviation Medicine.

Forsyth, R. S. (1989). The evolution of intelligence. In R. S. Forsyth (Ed.), *Machine learning: Principles and techniques* (pp. 65–82). London: Chapman and Hall.

Forsyth, R. S. (1990). *Neural learning algorithms: Some empirical trials.* Proceedings of the Third Conference on Neural Nets and their Applications (pp. 301–317). Nanterre, France.

Forsyth, R. S. (1992). Ockham's razor as a gardening tool: Simplifying discrimination trees by entropy minimax. In M. Bramer & R. W. Milne (Eds.), *Research and development in expert systems IX* (pp. 183–195). Cambridge: Cambridge University Press.

Hale, R. L., & Glassman, S. S. (1986). Using computers to construct, evaluate, and apply actuarial systems of classification: A methodology and example. *Computers in Human Behavior*, **2**, 195–213.

Hauck, W. W., & Miike, R. (1991). A proposal for examining and reporting stepwise regressions. *Statistics in Medicine,* **10**, 711–715.

Hess Consulting. (1992). ModelWare: Applications of the universal process modeling (UPM) algorithm. Manassas, VA: Author.

IMSL. (1987). *IMSL stat/library user's manual.* Houston, TX: Author.

Jain, A. K., & Ramaswami, M. D. (1988). Classifier design with Parzen windows. In E. S. Gelsema & L. N. Kanal (Eds.), *Pattern recognition and artificial intelligence* (pp. 211–228). Amsterdam: North-Holland.

James, M. (1985). *Classification algorithms.* London: Collins.

Jennrich, R., & Sampson, P. (1988). Stepwise discriminant analysis. In W. J. Dixon (Ed.), *BMDP statistical software manual* (Vol. 1) (pp. 347–356). Berkeley, CA: University of California Press.

Kaufman, L., & Rousseeuw, P. J. (1990). *Finding groups in data: An introduction to cluster analysis.* New York: Wiley.

Kelly, J. D., & Davis, L. (1991). *Hybridizing the genetic algorithm and the K nearest neighbors classification algorithm.* Proceedings of the Fourth International Conference on Genetic Algorithms (pp. 377–383). San Mateo, CA: Morgan Kaufmann Publishers.

Kohonen, T. (1988). *Self-organization and associative memory* (2nd ed.). Berlin: Springer-Verlag.

Lang, T. (1990, December 20). *Computer in illiteracy fight.* The Cincinnati Enquirer, A19.

Luk, A., & MacLeod, J. E. S. (1986). An alternative nearest neighbour classification scheme. *Pattern Recognition Letters*, **4**, 375–381.

Martindale, C. (1991). *Cognitive psychology: A neural network approach.* Pacific Grove, CA: Brooks/Cole.

McKenzie, D. P., & Low, L. H. (1992). The construction of computerized classification systems using machine learning algorithms: An overview. *Computers in Human Behavior*, **8**, 155–167.

McKenzie, D. P., McGorry, P. D., Wallace, C. S., Low, L. H., Copolov, D. L., & Singh, B. S. (1993). Constructing a minimal diagnostic decision tree. *Methods of Information in Medicine*, **32**, 161–166.

McLachlan, G. (1992). *Discriminant analysis and statistical pattern recognition.* New York: Wiley.

Musavi, M. T., Kalantri K., Ahmed, W., & Chan, K. H. (1993). A minimum error neural network. *Neural Networks*, **6**, 397–407.

Nelson, M. M., & Illingworth, W. T. (1991). *A practical guide to neural nets.* Reading, MA: Addison-Wesley.

Panel on Discriminant Analysis, Classification & Clustering. (1989). Discriminant analysis and clustering. *Statistical Science*, **4**, 34–69.

Patrick, E. A. (1990). The outcome advisor. *Pattern Recognition*, **23**, 1427–1439.

Patrick, J. (1991). *SNOB: A program for discriminating between classes* (Tech. Rep. 91/151). Melbourne, Australia: Monash University.

Rich, E., & Knight, K. (1991). *Artificial intelligence* (2nd ed.). New York: McGraw-Hill.

Riesbeck, C., & Schank, R. (1989). *Inside case-based reasoning.* Hillsdale, NJ: Lawrence Erlbaum.

Ripley, B. D. (1994). Neural networks and related methods for classification (with discussion). *Journal of the Royal Statistical Society: Series B*, **56**, 409–456.

Rissland, E. L., & Skalak, D. B. (1989). *Case-based reasoning in a rule-governed domain.* Proceedings of the Fifth Conference on Artificial Intelligence Applications (pp. 45–53). Washington, DC: IEEE Computer Society Press.

Rothman, P. (1992, October). *Syntactic pattern recognition. AI Expert*, **7**, 40–51.

SAS Institute. (1989). *SAS/STAT user's guide, version 6* (4th ed.). Cary, NC: Author.

Slade, S. (1991). Case-based reasoning: A research paradigm. *AI Magazine*, **12**, 42–55.

Specht, D. F. (1990). Probabilistic neural networks. *Neural Networks*, **3**, 109–118.

Specht, D. F. (1992). *Enhancements to probabilistic neural networks.* Proceedings of the International Joint Conference on Neural Networks, vol. 1 (pp. 761–768). Piscataway, NJ: IEEE.

Specht, D. F., & Shapiro, P. D. (1990, July). Training speed comparison of probabilistic neural networks with back-propagation networks. *Proceedings of the International Neural Network Conference* (pp. 440–444). Paris.

Specht, D. F., & Shapiro, P. D. (1991). Generalized accuracy of probabilistic neural networks compared with back-propagation networks. *Proceedings of the International Joint Conference on Neural Networks* (pp. 887–892). New York: IEEE.

Tabachnick, B. G., & Fidell, L. S. (1989). *Using multivariate statistics* (2nd ed.). New York: HarperCollins.

Teranet IA, Inc. (1992). *ModelWare and ModelWare Professional user's manual* [computer program]. Nanaimo, Canada: Author.

Tulving, E. (1972). Episodic and semantic memory. In E. Tulving & W. Donaldson (Eds.), *Organization of memory* (pp. 381–403). New York: Academic Press.

Tyron, W. W. (1993). Neural networks: I. Theoretical unification through connectionism. *Clinical Psychology Review*, **13**, 341–352.

Ward Systems Group. (1992). *NeuroWindows: Neural network dynamic link library* [computer program]. Frederick, MD: Author.

Weiss, S. M. (1991). Small sample error rate estimation for k-NN classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **13**, 285–289.

Weiss, S. M., & Kulikowski, C. A. (1991). *Computer systems that learn: Classification and prediction methods from statistics, neural nets, machine learning, and expert systems.* San Mateo, CA: Morgan Kauffman.

APPENDIX 1. AVAILABILITY OF COMMERCIAL CBS SOFTWARE

CBR EXPRESS
Inference Corp.
550 N. Continental Blvd.
El Segundo, CA 90245.

THE EASY REASONER
The Haley Enterprise
413 Orchard Street
Sewickley, PA 15143

ESTEEM
Esteem Software, Inc.
302 E. Main Street
Cambridge City, IN 47327

INDUCE-IT
Inductive Solutions, Inc.
380 Rector Place, Suite 4A
New York, NY 10280-1443

IMSL
IMSL
2500 ParkWest Tower One
2500 CityWest Blvd
Houston, TX 77042-3020

MEM-1
Cecase
2291 Irving Hill Rd
Lawrence, KS 66045-2969

NEURALWORKS PROFESSIONAL
NeuralWare, Inc.
Penn Center West
Building IV, Suite 227
Pittsburgh, PA 15276

NEUROSMARTS
Cognition Technology Corp.
1000 Massachusetts Avenue
Cambridge, MA 02138

NEUROWINDOWS, NEUROSHELL 2
Ward Systems Group, Inc.
Executive Park West
5 HillCrest Drive,
Frederick, MD 21702

OUTCOME ADVISOR
Patrick Consulting, Inc.
P.O. Box 14444
Cincinnati, OH 45214

P-Logic
Knowledge Systems
3rd floor
2133 Hawthorne Blvd
Torrance, CA 90505

REMIND
Cognitive Systems, Inc.
220-230 Commercial Street
Boston, MA 02019

MODELWARE/MODELWARE PROFESSIONAL
Teranet IA, Inc.
#119-3721 Delbrook Ave
North Vancouver, B.C.
Canada, V7N 4L9

SAS-PC
SAS Institute, Inc.
SAS Circle, Box 8000,
Cary, NC 27512-8000.

VisSIM/NEURAL-NET
Visual Solutions
487 Groton Rd
Westford, MA 01886