ELSEVIER

Expert Systems with Applications xx (2005) 1–7

Expert Systems with Applications

www.elsevier.com/locate/eswa

Generating rules from examples of human multiattribute decision making should be simple

Arie Ben-David *, Leon Sterling

[a] *Management Information Systems, Department of Technology Management, Holon Academic Institute of Technology, 52, Golomb St., P.O. Box 305, Holon 58102, Israel*

[b] *Department of Computer Science and Software Engineering, University of Melbourne, Melbourne, Vic., Australia*

Abstract

How many prototypes or clusters are needed to predict real world human multiattribute subjective decision making? Although subjective decision making problems occur daily in our life, they have received relatively little attention in artificial intelligence, machine learning and data mining communities. We claim that for most problems, a simple set of rules derived by a nearest neighbor algorithm is the appropriate approach. A simple version of a nearest neighbor model is tested and compared with two other well-established classification methods: neural networks and classifications and regression trees (CART). The results of the experiments show that the simple nearest neighbor method provides very accurate predictions while using very few prototypes or clusters. Although not always the best in accuracy, the differences are sufficiently slight to not warrant greater complexity in deriving rules. Our research on the effectiveness of parsimonious rule sets suggests that decision trees with more than 7–10 branches are not needed for capturing most human multiattribute decision-making problems, and minimal time or memory resources should be used to generate decision making rules.

Keywords: Neural networks; Classification and regression trees; Nearest neighbor; Exemplar-based learning; Data mining; Clustering; Human multiattribute decision-making

1. Introduction

Finding prototypes, clusters, or classification rules from large data sets has been the subject of extensive research in various fields such as artificial intelligence, pattern recognition, psychology, statistics, machine learning, and data mining over a long period of time. Besides its academic importance, finding classification rules has significant economic impact. Commercial companies are continually trying to employ more effective computerized decision-making tools in an increasingly competitive business environment.

It has been long been established that, given two models that classify a set of data equally accurately, the simpler is preferable to the more complex. For example, given two decision trees, which classify the data set they have been constructed from with equal accuracy, the decision tree with the fewer branches is preferable in terms of its generalization capabilities, that is on data taken from the same domain which it has never 'seen' before. Similarly, rule bases with fewer rules are likely to better generalize the underlying phenomenon than larger, seemingly more elaborate, rule bases. As yet another example, given two neural networks which classify a given data set equally well, the more complex network (i.e. the one with more layers and more processing elements) is likely to perform poorly on a previously unseen data set compared with the simper one (Pao, 1989).

The above phenomenon is referred to as *overfitting*. Overfitting occurs when a model which is generated from noisy learning examples 'learns' the noise rather than just the underlying patterns within the data. There is a potential for overfitting regardless of the nature of the approach being taken for building the model (Last & Maimon, 2004; Mitchell, 1997; Weiss & Kulikowski, 1991; Witten & Frank, 2000).

There are many more reasons why one would prefer simpler structures. For example, the rationale behind small decision trees or compact sets of decision-making rules is easier to explain to employees, their integration into company manuals and culture is simpler, they consume less space and they classify quicker if implemented in a computer algorithm. The 'smaller is better' principle is usually attributed to William of

* Corresponding author. Tel.: +972 3 5026744/6; fax: +972 3 5026650.
E-mail address: hol_abendav@bezeqint.net (A. Ben-David).

0957-4174/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2005.09.066

Occam. A full discussion of Occam's Razor is outside the scope of this paper, and the interested reader is referred to interesting discussions in (Domingos, 1999; Mitchell, 1997).

Beyond the basic well-understood principle of 'smaller is better', it would be useful for researchers and application developers in the context of machine learning and data mining to have more quantitative information on the expected size of rule sets. At least orders of magnitudes should be given as guidelines to various types of problems and problem-solving techniques. For example, assume that one has a large database of past decisions about credit cards, life insurance, university admission, or other decision-making applications. A typical task is to extract a decision tree or a set of rules (sometimes referred to as *prototypes*) which, if applied to a previously unseen example, will yield as small an error as possible. There are many well known algorithms which do exactly that. For a partial list and an extensive benchmark results see (Lim, Loh, & Shih, 2000).

Many of the available models include a predefined limit on the size of the generated tree or an upper limit on the number of generated prototypes. These parameters can be modified by designers to suit their needs. However, there are typically no indications, neither in the literature nor in user manuals, what values of these parameters are reasonable to begin with. An analogous question should be asked even when a tree-generating algorithm determines the 'optimal' size of the tree without any user intervention. For example, does an 'optimal' decision tree with 10,000 branches for granting credit line limits according to, say, ten features really make sense? What about a tree with 1000 branches? Maybe 100? Or 10?… This question underlies the experiments reported in this paper within the scope of a very common and important type of daily human multiattribute subjective decision-making problems (MDP for short) to be discussed in the next section.

To address this question, a modified version of the nearest neighbor algorithm was applied to four real world MDP data sets in different domains. The data sets all included past human decisions. Prototype examples were iteratively tested and added one at a time in a hill climbing search. The generalization accuracy of the resulting rule set was tested after each addition and the results were recorded. Two different techniques were also applied to the same data sets and used as benchmarks: Classification and Regression Trees, known as CART (Breiman, Friedman, Olshen, & Stone, 1998), which represented a well-established statistical method, and back propagation Neural Networks (Rumelhart & McLelland, 1986) which represented the (currently large) family of neural network algorithms. These particular algorithms were chosen since each of them is accepted by a wide variety of researchers from various disciplines, and is well documented in the literature. Ordinal Machine Learning Algorithms (i.e. those which just use order information), on the other hand, are quite rare and they have not been included so far in any major benchmark (Cao-Van and De Baetes (2000) and Kotsiantis and Pintelas (2004)). Since no error-cost function was known for any data set, which was used in this experiment, identical distance among all attribute and class values was assumed throughout. The approach used in this experiment enabled to incrementally generate rules from ordinal data sets, and to compare the results with those obtained via well-reputed machine learning models using identical error definition.

Similar to other reports (Lim et al., 2000; Kramer, Widmer, Pfahringer, & Groeve, 2000), none of the three methods was a clear-cut winner with respect to generalization accuracy, which usually serves as the most important criterion for selecting the 'best' model out of several candidates. However, the seemingly unsophisticated modified version of the nearest neighbor algorithm that was used in this experiment proved to be very competitive when compared with its more mathematically oriented counterparts. More importantly, after selecting just 3–7 prototypes from the data sets using a hill climbing (i.e. not exhaustive) search, the generalization capabilities of this simple model matched and even sometimes outperformed those of the other models, which served as benchmarks.

2. Background and related work

It might be instructive to begin with a short example of human multiattribute subjective decision-making problems (MDP): while evaluating this manuscript the reviewers have most likely subjectively solved a MDP of the type we are dealing with here. The reviewers quantitatively ranked several important attributes such as novelty, importance, interest to the readers and so on. Based on the values they have assigned to these attributes each has made his or her own assessment: accept, revise, reject, etc. Later on the editor has solved a similar MDP, based on his/her impression and the recommendations of the reviewers. Even the reader of this paper has probably solved an implicit MDP while deciding whether this paper is worth his or her precious time and attention to read. The topic, novelty, writing style, and other obligations, etc. have played a role in a decision: skip, browse, read carefully and so on. Many other problems in our daily life such as consumer preferences, credit rating, and employee assessment share many characteristics of our reviewer/editor problem. The reader can clearly add more MDPs to these typical examples without any difficulty.

It is quite surprising to note that while MDPs are so abundant in our every day life, they have received relatively little attention from artificial intelligence, machine learning and data mining communities until very recently. Most of the research in this area has been conducted by Cognitive Psychologists who were mainly interested in psychological aspects of human decision-making rather in how machines can mimic human behavior. Perhaps one of the main reasons for this phenomenon stems from the fact that in many cases of MDPs there is no clear-cut notion of a 'correct' class. In our running example, each reviewer is clearly entitled to his/her own point of view. Furthermore, he or she might be influenced by factors which are difficult to measure—a good mood or lack of time can sometimes affect a decision considerably. Also, even if two reviewers agree on the values of every single attribute of a particular manuscript, each can rightfully arrive at a different conclusion. In many cases many attribute values

(such as 'credit history'—a very important consideration in credit rating, or 'personal impression' during an interview of a candidate) are highly subjective and may differ not only from one evaluator to another, but within an individual's decisions over time. As a result MDP data sets, such as those which where used in this experiment are very noisy.

Unfortunately, human subjective MDPs have not been in the focus of machine learning and data mining research until very recently. As an example: 123 data sets are currently documented at the UCI Machine Learning Repository, which is the major source of data for Machine Learning research. Only three or four of the UCI data sets (about 3%) can be regarded as MDPs: credit card application, car evaluation, nursery, and (possibly) Balance. Consequently, it is rare to read in the literature results of machine learning or data mining models which were applied to MDPs. Consider, for instance, the very interesting and comprehensive comparative study reported by (Lim et al., 2000). Out of the 16 databases that were used in this study, only one (attitude towards smoking restrictions) is a MDP.

One way of recognizing a MDP is by looking at the original scales of both the attributes and the class. We as humans tend to use ordinal (i.e. ordered) scales. This does not necessarily mean that all the attribute and class values of a MDP must all be ordinal. There may be some numeric attributes as well ('account balance' for example), binary values (e.g. 'internal/external' candidate for a position), or categorical values. However, if most of the attributes and the class values are numeric one may rightfully suspect whether the problem at hand is a typical human MDP since we humans tend to generalize and simplify our reasoning by mapping numeric values into ordinal symbols. For instance, a credit officer does not usually make any use of the exact numeric values of an applicant income. Instead he or she tends to think in ordinal terms such as 'very high', 'high', 'average', etc. income.

The extensive use of ordinal symbols in human MDPs has not gone entirely unnoticed, and the topic did receive some attention over the years. Several versions of Ordinal Logistic Regression were proposed in the field of Statistics (e.g. McCullagh & Ndeler, 1983). Larichev, Moshkovich, and Furems (1986) have built a decision support system called CLASS, which helped to generate consistent and irredundent ordinal rule-bases. CLASS was assisting knowledge engineers, but was not a machine learning model as it entirely relied on its users choices. Only later MDPs have attracted the attention of AI researchers. A framework for ordinal learning reasoning was proposed by (Ben-David, Sterling, & Pao, 1989) in order to avoid monotonic inconsistencies in rule-bases. Later an approach for maintaining consistency in decision trees was proposed (Ben-David, 1995). Kramer et al. (2000) have developed a version of CART, called S-CART, which is a modified version of the well known CART algorithm, capable of working with ordinal classes. The S-CART was tested on four data sets (of which one was ordinal but a human MDP) and has shown good predictive accuracy. Cao-Van and De Baetes (2000), Kotsiantis and Pintelas (2004), Makino, Suda, Ono, and Ibaraki (1999) and Potharst and Bioch (2000), studied various aspect of ordinal classifications. However, there has been no report in these publications regarding the size of the resulting decision trees or other types of concepts they generated—which is the main topic of this paper.

An interesting work which might have been relevant to our work is Holte's (1993) 1R program which showed that classifying according to a single attribute may provide surprisingly accurate predictions when compared with more complex models. However, despite of the thorough check of 1R on sixteen data sets (taken from the UCI Repository), not a single data set in the experiment was based on human MDP, so it is impossible to assess whether the very interesting findings regarding 1R are pertinent to these problems or not.

3. The data sets

The four data sets which were used in this experiment came from actual human decision-making. They were not collected by us. Rather they were originally used in research into psychological decision-making[1]. We preferred working with our data sets since those very few MLPs in the UCI Machine Learning Repository are not well documented.

A characteristic of all four of the data sets is that they are all qualitative in nature. Also each data set contained decisions, which were taken by many individuals. The data sets included only ordinal (i.e. ordered) values for input properties and for the output. All the data sets were originally encoded as integers, {1, 2, 3,…} where 1 represented the 'worst' or lowest possible value, 2 the second 'worst', etc.

One can easily think of more complex decision-making tasks, for example using more features or using numeric values. However, it is not clear whether problems with significantly higher dimensionality than those which were used here actually reflect how we, as human beings, do our problem-solving (Tversky, 1969). We refer to this point again in the Discussion section.

3.1. Social workers decision (SWD)

The SWD data set contains real-world assessments of qualified social workers regarding the risk facing children if they stayed with their families at home. This evaluation of risk assessment is often presented to judicial courts to help decide what is in the best interest of an alleged abused or neglected child.

The total number of examples in the SWD data file was 1000, each having 10 inputs, such as the economic situation at home and the quality of the child–parent relationship, and one output, reflecting the assessed risk to the child.

[1] The authors are thankful to Prof. Yoav Ganzach of Tel Aviv University for contributing the data sets we used in this research.

Table 1
Quantitative characteristics of the data

	SWD	LEV	ESL	ERA
No. of examples	1000	1000	488	1000
No. of input attributes	10	4	4	4
No. of possible values (each attribute)	4	5	10	15
No. of possible values (output)	7	5	10	10

3.2. Lecturers evaluation (LEV)

The LEV data set contains 1000 examples of anonymous lecturer evaluations, taken at the end of MBA courses. Before receiving the final grades, students were asked to score their lecturers according to four attributes such as oral skills and contribution to their professional/general knowledge. The single output was a total evaluation of the lecturer's performance.

3.3. Employee selection (ESL)

The ESL data set contains 488 profiles of applicants for certain industrial jobs. Expert psychologists of a recruiting company, based upon psychometric test results and interviews with the candidates, determined the values of the input attributes. The same experts also predicted an overall score corresponding to the degree of fitness of the candidate to this type of job (the output).

3.4. Employee rejection–acceptance (ERA)

The ERA data set was originally gathered during an academic decision-making experiment aiming at determining which are the most important qualities of candidates for a certain type of jobs. Unlike ESL data set which was collected from expert recruiters, this data set was collected during a MBA academic course.

The input in the data set are features of a candidates such as past experience, verbal skills, etc., and the output is the subjective judgment of a decision-maker to which degree he or she tends to accept the applicant to the job or to reject him altogether (the lowest score means total tendency to reject an applicant and vice versa). The number of example decisions in ERA was 1000.

Table 1 shows quantitative characteristics of the three data sets.

4. The models

4.1. A variant of nearest neighbor

Nearest Neighbor is one of the oldest and most well known family of classification algorithms. The algorithms in the family all require some metric to determine how close two patterns are with respect to each other. Many distance metrics have been suggested over the years. Euclidean and Hamming distances are two notable examples (Pao, 1989). A description of the particular algorithm which we used for this experiment follows.

Since all the values in all the data sets were ordinal, a simple distance metric has been defined for this experiment. Let $\underline{\boldsymbol{k}}$ and $\underline{\boldsymbol{l}}$ denote two example vectors taken from the same problem domain, each with $\boldsymbol{n}$ attributes and one output value, the distance between their jth attribute value has been defined as:

$$\mathrm{Dist}_j^{\overline{kl}} = |\underline{\boldsymbol{k}}_j - \underline{\boldsymbol{l}}_j| \tag{1}$$

where $\boldsymbol{k}_j$ and $\boldsymbol{l}_j$ are integers representing the jth ordinal value in example vectors $\underline{\boldsymbol{k}}$ and $\underline{\boldsymbol{l}}$, respectively.

The total distance between two example attributes (i.e. input) is the summation of the distances over all their n attributes:

$$\mathrm{Dist}^{\overline{kl}} = \sum_j \mathrm{Dist}_j^{\overline{kl}} \tag{2}$$

The rationale behind this interpretation of 'distance' is to avoid favoring smaller distances over larger as in Euclidean distance, and to avoid just counting the number of disagreements as in Hamming distance while ignoring their magnitude.

Given an example vector $\underline{\boldsymbol{e}}$ taken from a data set $\boldsymbol{E}$ and a set of already selected prototypes $\boldsymbol{P}$, $\boldsymbol{P} = \{\underline{\boldsymbol{p}}_1, \underline{\boldsymbol{p}}_2, \underline{\boldsymbol{p}}_3, \ldots, \underline{\boldsymbol{p}}_{|p|}\}$, classification requires finding which prototype, $\underline{\boldsymbol{p}}' \varepsilon \boldsymbol{P}$, is the closest to the example input attributes and assigning it the output value of $\underline{\boldsymbol{p}}'$. The classification error of an example $\underline{\boldsymbol{e}}$ with respect to a set of prototypes $\boldsymbol{P}$, denoted $\mathrm{Err}^{\underline{\boldsymbol{e}}}\boldsymbol{P}$, has thus been defined as in (1) for the reasons discussed above, where $j = n+1$:

$$\mathrm{Err}^{\underline{\boldsymbol{e}}}\boldsymbol{P} = \mathrm{Err}^{\overline{ep'}} = \mathrm{Dist}_{n+1}^{\overline{ep'}} = |\underline{\boldsymbol{e}}_{n+1} - \underline{\boldsymbol{p}}'_{n+1}| \tag{3}$$

where $\underline{\boldsymbol{p}}'$ is the closest prototype in $\boldsymbol{P}$ to the input attributes of the example $\underline{\boldsymbol{e}}$.

The algorithm shown in Fig. 1 classifies a single example according to a set of already selected prototypes by choosing the closest one according to the defined metric.

Classify (an example $\underline{e}$; prototype set P)

{

1. **Select a prototype $\underline{p}'$ ε P which is the closest to $\underline{e}$'s input attributes**
2. **Return the output of $\underline{p}'$** // i.e., Classify $\underline{e}$ to the output of $\underline{p}'$;

}

Fig. 1. Nearest neighbor classification.

```
Create_Prototype_Set (a set of training examples E,
                      a set of testing examples T,
                                          max_no_of_rules)
{
  P = {}   // The prototype set is initially empty
  Do while (E ≠ {} and |P| < max_no_of_rules)
  {
    1. Select among all current example candidate set E, a single example, e',
       which, if added to the current prototype set P, minimizes the average
       error, AvErr^ET (i.e., the error over the testing set T.)
    2. ADD e' to the set of already selected prototypes, P.
    3. DELETE e' from the set of candidate examples E.
  }
  Return P
}
```

Fig. 2. Exemplar-based learning.

Finding the average classification error of a set $\boldsymbol{E}$ with $|\boldsymbol{E}|$ examples with respect to a set $\boldsymbol{P}$ of prototypes, designated $\mathrm{AvErr}^{\boldsymbol{EP}}$, involves picking the closest prototype for each example:

$$\mathrm{AvErr}^{\boldsymbol{EP}} = 1/|\mathbf{E}| * \Sigma \mathrm{Err}^{\underline{\boldsymbol{e}}}\boldsymbol{P} \tag{4}$$

where $\mathrm{Err}^{\underline{\boldsymbol{e}}}\boldsymbol{P}$ is the classification error of a single example with respect to a prototype set $\boldsymbol{P}$, defined in (3) and $|\mathbf{E}|$ is the number of examples.

During the learning phase the exemplar-based algorithm selected those examples in the training set, which would become prototypes. Fig. 2 gives an algorithm for generating the set of prototypes.

As can be seen from Fig. 2, the search for the best prototype set is not exhaustive as not all possible rule sets are tested. Rather it is a hill-climbing search for a locally optimized set of (hopefully) reasonably good prototypes. As we will shortly see, this simplified search strategy performed quite satisfactorily in all the experiments.

4.2. Neural networks (NN)

Neural networks (Minsky & Papert, 1969) have gained acceptance in recent years as an accurate classification tool. Back propagation neural networks (Rumelhart & McLelland, 1986) with several hidden layers and sigmoid transfer function were used in this experiment. These networks are known as being capable of approximating any nonlinear function to an arbitrary precision (Leshno, Ya Lin, Pinkus, & Schocken, 1993; Wary & Green, 1995). Since the model is well documented, it will not be discussed here in more detail.

4.3. Classification and regression trees (CART)

Classification and Regression Trees (Breiman et al., 1995), CART for short, have also gained wide recognition and acceptance in recent years. The CART algorithm is also well documented in the literature and will not be discussed here any further.

5. The experiment

Each data set was randomly partitioned into two mutually exclusive subsets: a learning set (three quarters of the examples) and a hold-out or validation set (one quarter of the examples). Testing examples were randomly selected from the respective learning sets (one third of the learning set). Each experiment was repeated ten times to allow ten-fold 25% holdout estimate of the prediction accuracies over the previously unseen examples.

The version for the Nearest Neighbor algorithm outlined above was written in Matlab. SPSSs Clementine version 7.2 package was used for Neural Networks and CART. Several Neural Networks topologies and CART parameter settings were tested for each set, but they generally did not outperform the default values with one notable exception: In Neural Networks, the number of hidden layers and the number of processing elements in each of them had to be determined manually for each data set by a trail and error process.

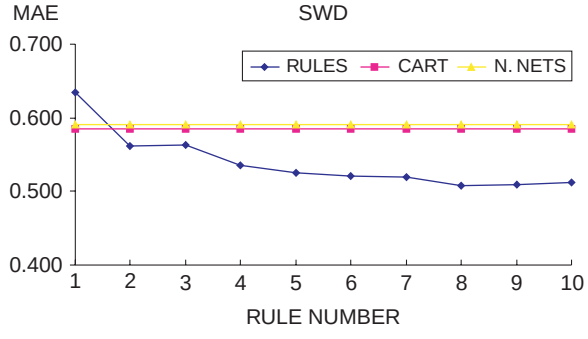

Fig. 3. Prediction errors (SWD).

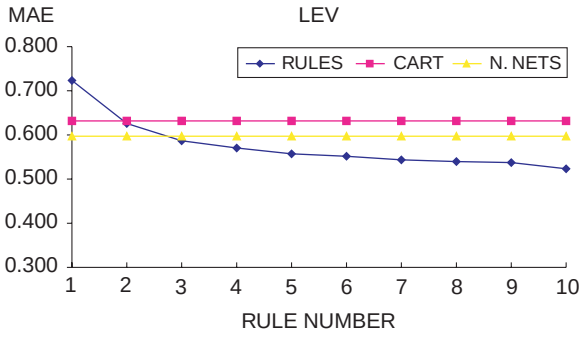

Fig. 4. Prediction errors (LEV).

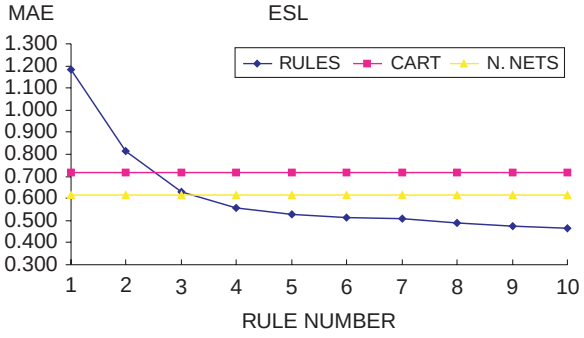

Fig. 5. Prediction errors (ESL).

6. Results

The results of the experiment are shown in Figs. 3–6. Each of these graphs shows the Mean Absolute Error (MAE) over the ten validation runs on the vertical axis versus the number of selected classification rules of the Nearest Neighbor algorithm (designated 'rules' in the legends). The errors were not normalized and they are as in the original data sets (see above). The MAE for CART and Neural Networks are shown in horizontal lines for comparison for their best performance only. These are the average MAE results of all the validation runs for each model. Again, note that unlike the Nearest Neighbor results which are shown incrementally, those which are shown for the two benchmark models reflect only their best and final results.

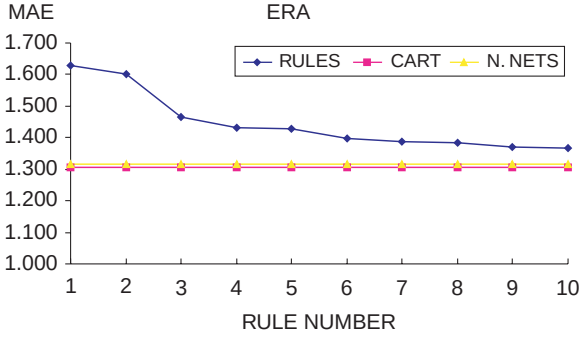

Fig. 6. Prediction errors (ERA).

Table 2
Summary of results

	SWD	LEV	ESL	ERA
MAE: nearest neighbor (seven rules)	0.5192	0.5436	0.5074	1.3876
MAE: neural networks	0.5972	0.5972	0.6147	1.3153
MAE: C&R trees	0.6318	0.6318	0.7187	1.3072
95% Confidence interval: nearest neighbor (half width)	0.0536	0.0304	0.0261	0.0619
95% Confidence interval: neural networks (half width)	0.0358	0.0358	0.0417	0.0464
95% Confidence interval: C&R trees (half width)	0.0523	0.0523	0.0419	0.0295

Table 2 shows the main results of the ten validation runs for each data set. The first three rows show the Mean Average Errors for each model, while the next three show the value of half the width of a 95% confidence interval around the MAE. The numbers of the Nearest Neighbor MAEs and confidence intervals were calculated after finding seven prototypes or classification rules in each data set.

For comparison, CARTs decision trees, for which the best results are shown in Figs. 3–6, were with 14 leaves (SWD), 19 leaves (LEV), 15 leaves (ELS), and 9 leaves (ERA). CART trees are also quite compact but still significantly larger than those, which were obtained by the Nearest Neighbor algorithm. The Neural Network configurations, which are shown in these figures, had three hidden layers with 10, 20, and 10 processing elements respectively for SWD, LEV and ERA and only one internal hidden layer with 7 processing elements for ESL.

Two observations are quite evident from the results:

First, it is clear from Figs. 3–6 that there has not been a clear-cut winner accuracy-wise. This observation is perhaps important and interesting by itself but it will not be discussed here any further simply because it reproduces known results. Several benchmarks, notably the extensive one reported in (Lim et al., 2000) have shown that not a single model can claim universal superiority in solving all classification problems; Methods which excel in some problem domains frequently score poorly in others. However, note that this is the first time these results are confirmed with respect to human MDPs. Even within this restricted class of problems, it was impossible to identify any winning Machine Learning algorithm.

The second observation is central to our research; It is quite evident that very few classification rules which were selected by a not-so-sophisticated and certainly not statistically oriented Nearest Neighbor algorithm were quite successful in generalizing previously unseen examples. Furthermore, there were cases where three classification rules (SDW) or six (LEV) were sufficient to beat their well established counterparts.

7. Conclusions

Certainly it has not been shown here (nor has it been intended) that the particular version of the Nearest Neighbor algorithm which is detailed above is superior to other classification models when applied to human MDPs. What has been shown here indeed is that an order of magnitude of ten prototypes, if found amongst the training set, can generalize very well, sometimes doing even better than other, more sophisticated models.

Our main conclusion from this experiment is that while applying classification techniques to the many problem domains which are similar in nature to those which are reported here, it is pointless to allow decision trees or rule bases to grow beyond an order of magnitude of ten branches or rules. Not 10,000, not 1000, not 100 but an order of magnitude of ten.

Some support for our findings comes from the field of Cognitive Psychology: Miller (1956) has shown what human short-term memory can hold only about 7 ± 2 'chunks of information' (later known as Miller's 'Magical Numbers'). Each 'chunk' is roughly equivalent to one multiattribute decision-making rule. This phenomenon was later confirmed by Ganzach (1993), Simon (1978) and Tversky (1969) as well as by others. We suspect that human MDPs utilize our short-term memory due to similar storage requirements, but we leave the burden of proof to Cognitive Psychology. However, form Machine Leaning point of view, if we human beings can do quite well at generalizing from previous experience using such a limited capacity of short-term memory, there is no reason to expect a contradictory phenomenon while building decision trees or rule bases which are intended to solve very similar problems.

But what about those many application domains which are known to have classification rules in numbers which by far exceed an order of magnitude of ten? Problems which require significantly more branches or rules in their (machine) learned concepts are apparently not MDPs: Chess playing, image understanding, some medical and complex mechanical diagnosis, etc., require additional human skills (and 'computational' resources) such as processing of sensual data, pattern recognition, etc.. Those problems are quite distinct from human MDPs which were studied here. Nevertheless, MDPs are so very common and important in our daily life, hence a better understanding how to efficiently learn then by computers can be very interesting and rewarding.

References

Ben-David, A. (1995). Monotonicity maintenance in information-theoretic machine learning algorithms. *Machine Learning*, *9*, 29–43.

Ben-David, A., Sterling, L., & Pao, Y. H. (1989). Learning and classification of monotonic ordinal concepts. *Computational Intelligence*, *5*(1), 45–49.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1998). *Classification and regression trees*. London/Boca Raton: Chapman & Hall/CRC.

Cao-Van, K., & De Baetes, B. (2002). On the definition and representation of a ranking. *Lecture Notes in Computer Science*, *2561*, 291–299.

Domingos, P. (1999). The role of Occam's razor in knowledge discovery. *Data Mining and Knowledge Discovery*, *3*, 409–425.

Ganzach, Y. (1993). Goals as determinants of nonlinear noncompensatory judgment strategies. *Organizational Behavior and Human Decision Processes*, *56*, 422–440.

Holte, R. C. (1993). Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, *11*, 65–91.

Kotsiantis, S. B. & Pintelas, P. E., (2004). A cost sensitive technique for ordinal classification problems. *Proceedings of SETN 2004*. Berlin: Springer (pp. 220–229).

Kramer, S., Widmer, G., Pfahringer, B., & Groeve, M. D. (2000). Prediction of ordinal classes using regression trees. *Fundamenta Informaticae*, *34*, 1–15.

Larichev, O. I., Moshkovich, H. M., & Furems, E. M. (1986). Decision support system class. In H. Brehmer, et al. (Ed.), *New directions in research on decision making* (pp. 303–315). Amsterdam: Elsevier.

Last, M., & Maimon, O. (2004). A compact and accurate model for classification. *IEEE Transactions on Knowledge and Data Engineering*, *16*(2).

Leshno, M., Ya Lin, Y., Pinkus, A., & Schocken, S. (1993). Multi-layer feed-forward networks with a non-polynomial activation function can approximate any function. *Neural Networks*, *6*, 861–867.

Lim, T. S., Loh, W. Y., & Shih, Y. S. (2000). A comparison of prediction accuracy, complexity and training time of thirty-three old and new classification algorithms. *Machine Learning*, *40*, 203–229.

Makino, K., Suda, T., Ono, H., & Ibaraki, T. (1999). Data analysis by positive decison trees. *Fundamenta Informaticae*, *47*, 1–13.

McCullagh, P., & Nedler, J. A. (1983). *Generalized linear models*. London: Chapman & Hall.

Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, *63*(2), 81–97.

Minsky, M. L., & Papert, S. A. (1969). *Perceptrons*. Cambridge, MA: MIT Press.

Mitchell, T. M. (1997). *Machine learning*. New York, NY: McGraw-Hill.

Pao, Y. H. (1989). *Adaptive pattern recognition and neural networks*. Reading, MA: Addison-Wesley.

Potharst, R., & Bioch, J. (2000). Decision trees for ordinal classification. *Intelligent Data Analysis*, *4*, 97–112.

Rumelhart, D. E., & McLelland, J. L. (1986). *Parallel distributed processing learning: Internal representation by error propagation* (Vol. 1). Cambridge, MA: The MIT Press.

Simon, H. A. (1978). Information-processing theory of human problem solving. In W. K. Estes, *Handbook of learning and cognitive processing* (Vol. 5) (pp. 271–295). Hillsdale, NJ: Lawrence Erlbaum Associates.

Tversky, A. (1969). Intransivity of preferences. *Psychological Review*, *76*, 31–48.

Wary, J., & Green, G. R. (1995). Networks, approximation theory and finite precision computation. *Neural Networks*, *8*(1), 31–35.

Weiss, S. M., & Kulikowski, C. A. (1991). *Computer systems that learn*. Los Altos, CA: Morgan-Kaufmann.

Witten, I. H., & Frank, E. (2000). *Data mining*. London: Academic Press.