Benchmarking the n-tuple Classifier with StatLog datasets

Richard Rohwer and Michał Morciniec
Dept. of Computer Science and Applied Mathematics
Aston University
Birmingham, UK B4 7ET

Abstract

The n-tuple recognition method was tested on 11 large real-world data sets and its performance compared to 23 other classification algorithms. On 7 of these, the results show no systematic performance gap between the n-tuple method and the others. Evidence was found to support a possible explanation for why the n-tuple method yields poor results for certain datasets. Preliminary empirical results of a study of the confidence interval (the difference between the two highest scores) are also reported. These suggest a counter-intuitive correlation between the confidence interval distribution and the overall classification performance of the system.

1 Introduction

The n-tuple classification system is one of the oldest neural network pattern recognition methods [5], and there have been many reports of its successful application in various domains [10, 4, 12, 11, 6]. The major advantage of the method is its lightning speed. Learning is accomplished by recording features of patterns in a random-access memory, which requires just one presentation of the training set to the system. Similarly, recognition of a pattern is achevied by checking memory contents at addresses given by the pattern.

It is prudent to suspect that relatively poor performance will accompany the speed and simplicity of the n-tuple algorithm. We therefore carried out a large-scale experiment [7] in which the n-tuple method was tested on 11 real-world datasets previously used by the European Community ESPRIT StatLog project [8] in a comparison of 23 other classification algorithms including the most popular neural network methods. The results, reviewed below, show the n-tuple method to be a strong performer, except in a few cases for which we can offer explanations.

Statistics were also recorded on the confidence intervals (the differences between the two highest scores). Preliminary results suggest that two types of distribution occur, and performance is correlated to the distribution type.

2 Selection and pre-processing of StatLog data sets

The StatLog project was designed to carry out comparative testing and evaluation of classification algorithms on large scale applications. About 20 data sets were used to estimate the performance of 23 procedures. These are described in detail in [8]. This study used 11 large data sets, selected as described in [7]. A specific random division into training and test sets was supplied for each data set.

The attributes of the patterns in the StatLog data sets are mostly real numbers or integers. Therefore each attribute was rescaled into an integer interval, quantised, and converted into a bit string by the method of Kolcz and Allinson [2], [3] based on CMAC and Gray coding techniques.

The prescription for encoding integer x is to concatenate K bit strings, the jth of which (counting from 1) is $\frac{x+j-1}{K}$, rounded down and expressed as a Gray code. The Gray code of an integer i can be obtained as the bitwise exclusive-or of i (expressed as an ordinary base 2 number) with $i/2$ (rounded down). This provides a representation in aK bits of the integers between 0 and $(2^a - 1)K$ inclusive, such that if integers x and y differ arithmetically by K or less, their codes differ by Hamming distance $|x - y|$, and if their arithmetic distance is K or more, their corresponding Hamming distance is at least K. The resulting bit strings are concatenated together, producing an input vector of length $L = aKA$, where A is the number of attributes.

3 Benchmarking results

The benchmark tests used preprocessor parameters $a = 5$ and $K = 8$, so each attribute was scaled to the interval $[0, 248]$ and coded in 40 bits. (Test set attributes falling outside this range under the scaling based on the training set were truncated.) The recogniser used 1000 n-tuples of size $n = 8$, with 1 bit of memory at each address. In the event of a 'tie', meaning that the highest-scoring class had the same score as one or more other classes, the class among these with the highest *a priori* probability was selected.

Results from the benchmarking exercise are shown together with the Statlog results for other algorithms in Figure 1. Table 1 is a key to the symbols representing the various algorithms in this figure. Classification performance is normalised to the probability of the most common class, which equals the success rate obtainable by the trivial algorithm of always guessing that class.

The results show no systematic bias against the n-tuple method, except on the 4 data sets which tended to be the most problematic for the other algorithms. The geometric properties of the n-tuple method provide insight into the problematic cases.

RAMnets	
(●) n-tuple recogniser	
Discriminators	
(♣) 1-hidden-layer MLP.	(♠) Radial Basis Functions.
(♡) Cascade Correlation.	(⊕) SMART (Projection pursuit).
(⊗) Dipol92 (Pairwise linear discrim.).	(⊖) Logistic discriminant.
(⊘) Quadratic discriminant.	(⊙) Linear discriminant.
Methods related to density estimation	
(α) CASTLE (Prob. decision tree).	(β) k-NN (k nearest neighbours).
(γ) LVQ (Kohonen).	(δ) Kohonen Topo. map.
(ε) NaiveBayes (Indep. attributes).	(ζ) ALLOC80 (Kernel functions)
Decision trees	
(a) NewID (Decision Tree)	(b) AC^2 (Decision Tree)
(c) Cal5 (Decision Tree)	(d) CN2 (Decision Tree)
(e) C4.5 (Decision Tree)	(f) CART (Decision Tree)
(g) IndCART (CART variation)	(h) BayesTree (Decision Tree)
(i) ITrule (Decision Tree)	

Table 1: Synopsis of Algorithms with symbols used in Figure 1.

4 Counting Hypercubes

It is well established that the tuple distance between two patterns (the expected number of tuples on which they differ) decays exponentially with Hamming distance according to

$$\rho(H) \approx N\left(1 - e^{-\frac{n}{L}H}\right) \tag{1}$$

to a good approximation [1, 9]. Therefore each training pattern defines an A-dimensional hypercube with edges of Hamming length roughly aK/n in each attribute, over which it can contribute to the score of a test pattern. The preprocessor gives Hamming distances linear in arithmetic differences up to Hamming distance K of a possible aK per attribute, corresponding to arithmetic difference K of a possible $(2^a - 1)K$. Demanding this linearity throughout the region of near tuple distance gives $K > Ka/n$, or

$$a < n. \tag{2}$$

The input region corresponding to a training pattern's hypercube consists of scalar differences up to $\pm K$ in each attribute, which is the fraction $\frac{2K}{(2^a-1)K}$ or about 2^{1-a} of the maximum separation $(2^a - 1)K$. Therefore each training pattern generalises to an input space hypercube whose volume is fraction $2^{-(a-1)A}$ of the total input space. Assuming a Gaussian distribution for the data, it is then possible to estimate the number of hypercubes need to cover the input region where data is likely to occur. The estimate is the product of the eigenvalues of the training data covariance matrix, in units of hypercube edge length, rounded up.[1] The results are shown in figure 2.

[1]Only the eigenvalues smaller than the edge length were rounded up; ie., they were dropped from the product. Neglecting to round the others does not affect the order of magnitude of the result.

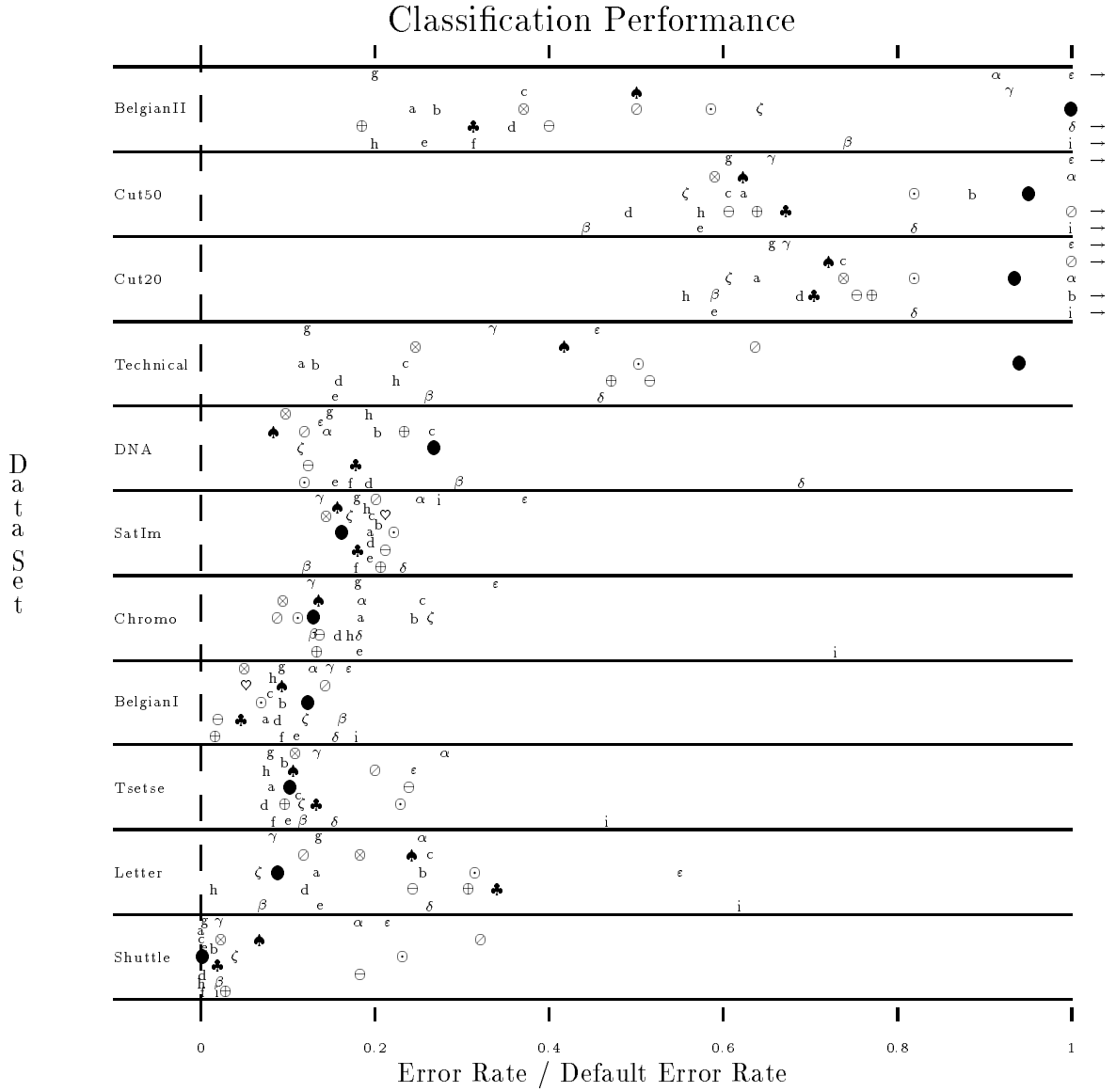

Figure 1: Results for N-tuple (●) and other algorithms. Classification error rates increase from left to right, and are scaled separately for each data set, so that they equal 1 at the error rate of the trivial method of always guessing the class with the highest prior probability, ignoring the input pattern. The arrows indicate the few cases in which performance was worse than this.

Evidently the data sets on which the n-tuple method fails are those requiring astronomical numbers of hypercubes to cover the data. The exceptions are "Technical" which is covered by just 1 hypercube, and "DNA" which is unusual in that originally Boolean attributes were treated as integers.

It is not easy to fix this problem by tuning parameters. The "generalisation length" aK/n can be increased by increasing a or K, or decreasing n. These alterations on a and n run into difficulty with (2). Increasing K gives longer representations of the patterns, which may in turn require the use of more N-tuples to adequately sample them.

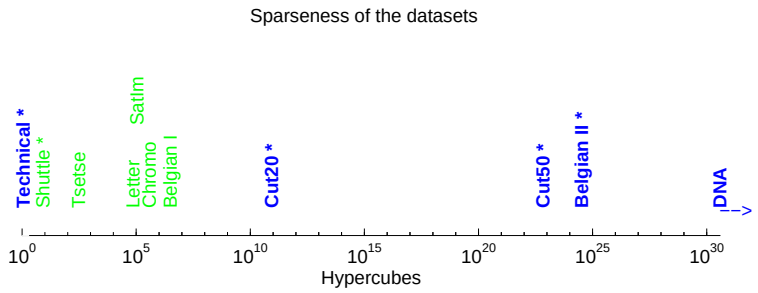

Figure 2: The number of hypercubes required to cover the space occupied by data. The datasets on which n-tuple classifier performed poorly are printed in bold face. A star denotes the existence of skewed priors. The DNA dataset had a highly redundant representation of its attributes and most of the data for Technical was concentrated in one hypercube.

5 Confidence intervals

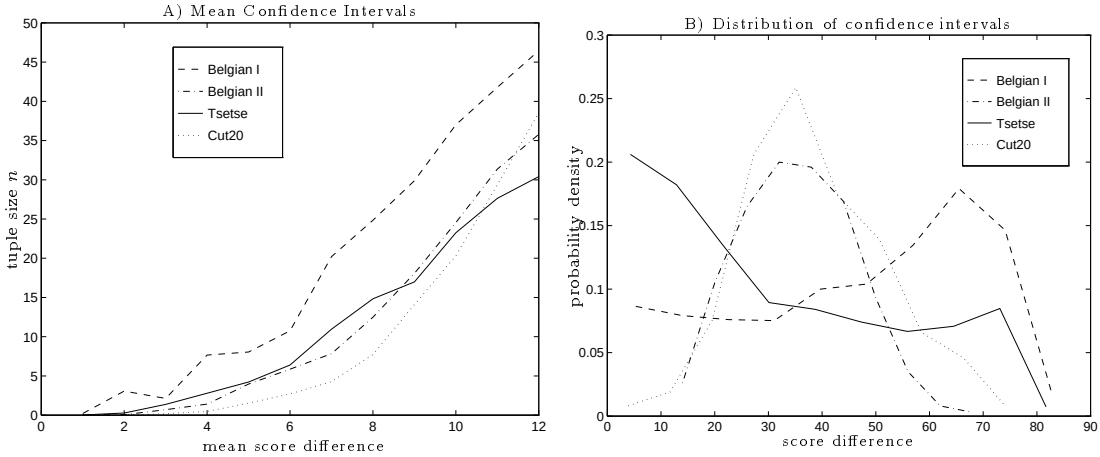

Figure 3: A) Average Confidence intervals as a function of tuple size n for several StatLog datasets. B) distribution of relative confidence intervals for tuple size $n = 12$.

The n-tuple system classifies a pattern to the class c that yields maximal tuple proximity (score) with discriminator D_c. The difference of maximal and next maximal score (the confidence interval) varies with the pattern. The mean confidence interval as a function of tuple size n is plotted in Figure 3A. The n-tuple system with 100 tuples was applied to the classification of several StatLog datasets. Two of them, Belgian II and Cut20, pose problems to the classifier (see section 3). The confidence interval increases with n, as higher-order correlations become available to the classifier. However, there seems to be no correlation between the size of the confidence interval and the percentage of correct classifications made.

Figure 3B presents the distribution of confidence intervals for the tuple size $n = 12$. It appears that the distributions tend to follow two forms: one approximately symmetrical, with a very low count of small confidence intervals, the other asymmetric with a considerable number of small score differences. The datasets on which the n-tuple classifier scores poorly seem to possess the symmetrical distribution.

This preliminary data seems to suggest, oddly, that the n-tuple classifier gives correct classifications with small confidence intervals, whereas mistakes are made "confidently".

6 Conclusions

A large set of comparative experiments shows that the n-tuple method is highly competitive with other popular methods, and other neural network methods in particular, except on data sets of high volume relative to the volumes naturally associated with the n-tuple method. Preliminary results suggest that confidence interval distributions fall into two categories, and that these are correlated with classification performance.

7 Acknowledgement

The authors are grateful to Louis Wehenkel of Universite de Liege for useful correspondence and permission to report results on the BelgianI and BelgianII data sets, Trevor Booth of the Australian CSIRO Division of Forestry for permission to report results on the Tsetse data set, and Reza Nakhaeizadeh of Daimler-Benz, Ulm, Germany for permission to report on the Technical, Cut20 and Cut50 data sets.

References

[1] N.M. Allinson and A. Kołcz. Distance relationships in the n-tuple mapping. submitted to Pattern Recognition.

[2] N.M. Allinson and A. Kołcz. Enhanced n-tuple approximators. *Weightless Neural Network Workshop*, pages 38–45, 1993.

[3] N.M. Allinson and A. Kołcz. Application of the cmac input encoding scheme in the n-tuple approximation network. *IEE Proceedings on Comput. Digit. Tech.*, 141(3):177–183, 1994.

[4] W.W. Bledsoe and C.L. Bisson. Improved memory matrices for the n-tuple recognition method. *IRE Joint Computer Conference*, 11:414–415, 1962.

[5] W.W. Bledsoe and I. Browning. Pattern recognition and reading by machine. *Proceedings of the Eastern Joint Computer Conference*, pages 225–232, 1959.

[6] R. Rohwer and D. Cressy. Phoneme classification by boolean networks. *Proceedings of the European Conference on Speech Communication and Technology*, 2:557–560, 1989.

[7] R. Rohwer and M. Morciniec. The theoretical and experimental status of the n-tuple classifier. Technical Report NCGR/4347, Aston University Neural Computing Research Group, Aston Triangle Brimingham B4 7ET UK, 1995.

[8] D. Michie D.J. Spiegelhalter and C.C. Taylor, editors. *Machine Learning, Neural and Statistical Classification.* Prentice–Hall, 1994.

[9] G. D. Tattersall and R. D. Johnson. Speech recognisers based on n-tuple sampling. In *Proc. Institute of Acoustics-Spring Conference*, pages 405–413, 1984.

[10] I. Aleksander W.V. Thomas and P.A. Bowden. Wisard a radical step forward in image recognition. *Sensor Review*, pages 120–124, 1984.

[11] J.R. Ullmann. Experiments with the n-tuple method of pattern recognition. *IEEE Transactions on Computers*, 18(12):1135–1137, 1969.

[12] J.R. Ullmann and P.A. Kidd. Recognition experiments with typed numerals from envelopes in the mail. *Pattern Recognition*, 1:273–289, 1969.