

## *Technical Note*

# First Nearest Neighbor Classification on Frey and Slate's Letter Recognition Problem

TERENCE C. FOGARTY  
*Bristol Polytechnic, Bristol BS16 1QY, U.K.*

TC\_FOGAR@UK.AC.BRISTOL-POLY.CSD

**Editor:** Tom Dietterich

At the end of the paper entitled "Letter Recognition Using Holland-Style Adaptive Classifiers" (Frey & Slate, 1991) the authors state that it would be of interest to determine the effectiveness of some other approaches to their letter recognition problem. Access is provided to the data used in their experiments via the UCI repository of machine learning databases.<sup>1</sup> This is a note of results obtained on that data using a first nearest neighbor classification algorithm equivalent to IBI (Aha, Kibler & Albert, 1991).

First nearest neighbor classification entails retaining all members of the training set in memory and using them to classify each member of the testing set. To determine the class of a member of the testing set, its Euclidean distance from each member in memory is calculated. It is then assigned the same classification as the classification of the member it is nearest to. Conflicts are resolved simply by choosing the classification of the first member in memory with the shortest distance.

Results obtained using this method give over 95.4% accuracy compared with Frey and Slate's best result of 82.7%. These results were obtained by using the first 16,000 examples after some starting point in the database as the "training" or memory set and the next 4,000 as the testing set. Ten experiments were conducted from randomly generated starting points in the database and an average accuracy of 95.67% with a standard deviation of 0.12% was obtained.

It can be argued that effectiveness should not be the only basis for comparison. If efficiency is to be taken into consideration the first nearest neighbor algorithm will require more time and space than Frey and Slate's method, because it does not compress the data at all. However, with regard to speed of computation, Waltz (1990) shows that simple approaches of the nearest neighbor type are well suited to implementation on massively parallel systems. With regard to size of memory, good results can be obtained on this problem using the first nearest neighbor algorithm with a much smaller memory set.

Using only the first 1,600 examples in the database, after some starting point, as the memory set and the next 4,000 as the testing set, as before, results giving over 81.5% accuracy were obtained. Ten experiments were conducted from randomly generated starting points in the database and an average accuracy of 82.57% with a standard deviation of 0.66% were obtained. These results were obtained using "training" or memory sets of only 10% of the size of the Frey and Slate training set yet are comparable to their best result.

This demonstrates that first nearest neighbor classification substantially outperforms Holland-style adaptive classifiers (HSAC) on this letter recognition problem. No comparison of the two approaches can be made on other data sets before Frey and Slate's algorithm is made publicly available. It should be noted that Frey and Slate's best results were obtained using an exemplar-based rule generation technique which owes much to the instance-based learning paradigm and little to the genetic learning paradigm. Results obtained relying purely on the random generation of rules, crossover, and mutation produced classification accuracy of no more than 67.8% using a training set of 16,000 examples.

To summarize, results using 16,000 examples for training and 4,000 for testing were:

1stNN 95.7%

HSAC 82.7% (with exemplar-based induction)

HSAC 67.8% (without exemplar-based induction)

Only 26 classes of data are identified, corresponding to the letters of the alphabet, but each class is composed of examples from 20 different fonts, some Gothic. Thus the data is in fact a conglomeration of 520 sub-classes in which difference in font may well be as important as difference in letter in some cases. Keeping all examples of each sub-class available for comparison is obviously a better strategy than trying to generalize. The fact that each of the 16 attributes for each letter has been normalized and that the features are of about the same importance means that simple nearest neighbor methods work best.

The main lesson to learn from this is the importance of comparative studies when introducing new algorithms in case studies. To facilitate these, a repository of machine learning algorithms would be the ideal complement to the existing machine learning databases repository.

## Notes

1. Send email to [ml-databases@ics.uci.edu](mailto:ml-databases@ics.uci.edu) for information on accessing the contents of the machine learning databases repository.

## References

- Aha, D.W., Kibler, D., & Albert, M.K. (1991). Instance-based learning algorithms. *Machine Learning*, 6, 37-66.
- Frey, P.W., & Slate, D.J. (1991). Letter recognition using holland-style adaptive classifiers. *Machine Learning*, 6, 161-182.
- Waltz, D. (1990). Massively parallel AI. *Proceedings of the Eighth National Conference on Artificial Intelligence* (pp. 1117-1122). San Mateo, CA: Morgan Kaufmann.