

Why Do Nearest-Neighbour Algorithms Do So Well?

Brian D. Ripley

*Professor of Applied Statistics
University of Oxford*

ripley@stats.ox.ac.uk
<http://stats.ox.ac.uk/~ripley>

References

- Ripley, B. D. (1996) *Pattern Recognition and Neural Networks*. CUP. ISBN 0-521-48086-7.
- Devijver, P. A. & Kittler, J. V. (1982) *Pattern Recognition. A Statistical Approach*. Prentice-Hall.
- Fukunaga, K. (1990) *Introduction to Statistical Pattern Recognition*. Second edition. Academic Press.
- Michie, D., Spiegelhalter, D. J. & Taylor, C. C. (eds) (1994) *Machine Learning, Neural and Statistical Classification*. Ellis Horwood.
- Weiss, S. M. & Kulikowski, C. A. (1991) *Computer Systems that Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning and Expert Systems*. Morgan Kaufmann.

1

Pattern Recognition

Supervised vs unsupervised

Most pattern recognition problems have pre-assigned classes of patterns, and the task is to classify new observations.

Sometimes the task includes finding the patterns.

'Statistical' pattern recognition

so called by electrical engineers.

Very little is assumed about the classes of patterns: everything is 'learned from examples'.

Structural pattern recognition

Quite a lot is assumed about the classes of patterns.

Main example is *syntactic* pattern recognition using formal languages.

Very few working examples.

2

Examples of Pattern Recognition

- Railway crossing
- Types of galaxies (SKYCAT)
- Medical diagnosis
- Detecting abnormal cells in cervical smears
- Recognizing dangerous driving conditions
- Reading ZIP codes
- Reading hand-written symbols (on a penpad)
- 'Logicook' microwave oven
- Financial trading
- Spotting fraudulent use of credit cards
- Spotting fake 'antique' furniture
- Forensic studies of fingerprints, glass
- Fingerprints, retinal patterns for security
- Reading a vehicle number plate
- Reading circuit diagrams
- Industrial inspection
- Oil-well lithofacies
- Credit allocation rules

3

Machine Learning

‘Machine Learning is generally taken to encompass automatic learning procedures based on logical or binary operations, that learn a task from a series of examples.’

‘Machine Learning aims to generate classifying expressions simple enough to be understood easily by humans. They must mimic human reasoning sufficiently well to provide insight into the decision process. Like statistical approaches, background knowledge may be exploited in development, but operation is assumed without human intervention.’ (Michie *et al.*, 1994, p. 2)

This stresses the need for a comprehensible explanation, which is needed in some but not all pattern recognition tasks. We have already noted that we cannot explain our identification of faces, and to recognize Zip codes no explanation is needed, just speed and accuracy.

4

Overview of Approaches

to supervised statistical pattern recognition.

- Linear and logistic discrimination
- Nearest-neighbour approaches
- Partitioning, especially classification trees
- Rule-based expert systems
- Neural networks
- Bayesian belief nets

A few comprehensive comparisons, especially Michie *et al.* Nearest-neighbour approaches come at or near the top.

Why?

5

Nearest-Neighbour Approaches

Several variants. Main one, usually attributed to Fix & Hodges (1951), is k -NN. Given a pattern x to classify

- Find the k nearest patterns in the training set
- Decide the classification by a majority vote amongst these k

[A less-popular variant is to find the nearest ℓ patterns of each class, and choose the class with the nearest set.]

Begs many questions:

- What similarity measure?
- Should less relevant features be included?
- How do we choose k ?
- What about tied votes (possible for $k > 1$)?
- Should we weight votes by proximity?
- Do we want a minimum of consensus
((k, ℓ) -rules)

6

Theory

There is an elegant large-sample theory, mainly due to Cover & Hart (1967). This looks at an abundance of training cases, so we can assume that there are many examples in a region in which the prevalences of the K classes do not change significantly.

In this theory there is no advantage in weighting votes, and the similarity measure is irrelevant, except to define the ‘local’ region of constant prevalence.

Let E_k be the error rate of k -NN (averaged over training sets) and E^* the minimum possible error rate (Bayes’ risk). For K classes:

$$E^* \leq E_1 \leq E^* \left(2 - \frac{K}{K-1} E^* \right) \leq 2E^*$$

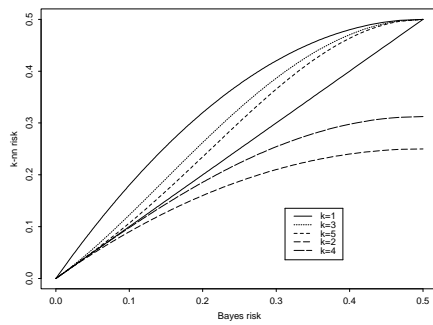
For $k = 2$

$$\begin{aligned} E'_2 &\leq E'_4 \leq \dots \leq E'_{2k} \nearrow E^* \searrow E_{2k} \\ &= E_{2k-1} \leq \dots \leq E_2 = E_1 = 2E'_2 \end{aligned}$$

where E' refers to not declaring an opinion if the vote is tied, and E to breaking ties at random.

7

Looking at the performance at particular x 's:



Large-sample risk r_k (k odd) or r'_k (k even) of k -NN rules against the Bayes risk r^* in a two-class problem.

There is also a theory of (k, ℓ) -NN rules, where a minimum vote of ℓ is needed.

Choice of Metric

This is often ignored: I have heard k -NN described as an automatic method (as against neural networks, say) without describing how k and the metric are to be chosen.

There *are* studies on choosing the metric automatically, almost all as a quadratic form in the variables (equivalently as Euclidean distance on linearly transformed variables), but Euclidean distance is the normal choice.

It can be crucial to down-weight less-relevant variables. Hastie & Tibshirani's DANN uses (locally) the best linear discriminator, but good features can be poor linearly. One neat idea is to use the importance of features as measured by a non-linear method (e.g. classification trees).

Where there are both continuous and categorical variables, Gower's general dissimilarity measure is the most common choice.

Data Editing

One common complaint about k -NN methods is that they can take too long to compute and need too much storage for the whole training set. The difficulties are sometimes exaggerated, as there are fast ways to find near neighbours.

However, in many problems it is only necessary to retain a small proportion of the training set to approximate very well the decision boundary of the k -NN classifier. This concept is known as *data editing*. It can also be used to improve the performance of the classifier by removing apparent outliers.

Multiedit

1. Put all patterns in the current set.
2. Divide the current set more or less equally into $V \geq 3$ sets. Use pairs cyclically as test and training sets.
3. For each pair classify the test set using the k -NN rule from the training set.
4. Delete from the current set all those patterns in the test set which were incorrectly classified.
5. If any patterns were deleted in the last I passes return to step 2.

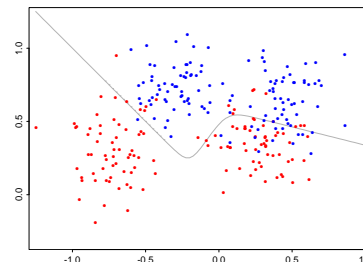
The edited set is then used with the 1-NN rule.

The *multiedit* algorithm aims to form homogeneous clusters. However, only the points on the boundaries of the clusters are really effective in defining the classifier boundaries. *Condensing* algorithms aim to retain only the crucial exterior points in the clusters.

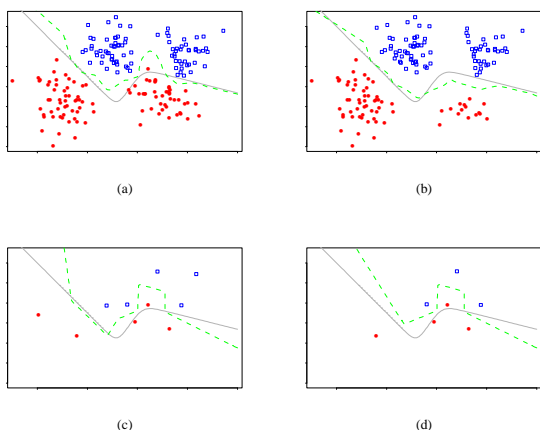
For example, Hart (1968) gives:

1. Divide the current patterns into a store and a grabbag. One possible partition is to put the first point in the store, the rest in the grabbag.
2. Classify each sample in the grabbag by the 1-NN rule using the store as training set. If the sample is incorrectly classified transfer it to the store.
3. Return to 2 unless no transfers occurred or the grabbag is empty.
4. Return the store.

A refinement, the *reduced nearest neighbour rule* of Gates (1972), is to go back over the condensed training set and drop any patterns (one at a time) which are not needed to correctly classify the rest of the (edited) training set.



A synthetic example of 250 points, 125 of each class. The best possible decision boundary is shown as a continuous curve: this is only known for synthetic problems.



Reduction algorithms. The known decision boundary of the Bayes rule is shown with a solid line; the decision boundary for the 1-NN rule is shown dashed.

- (a) *multiedit*.
- (b) The result of retaining only those points whose posterior probability of the actual class exceeds 90% when estimated from the remaining points.
- (c) *condense* after *multiedit*.
- (d) *reduced NN* applied after *condense* to (a).

Learning Vector Quantization

The refinements of the k -NN rule aim to choose a subset of the training set in such a way that the 1-NN rule based on this subset approximates the Bayes classifier. It is not necessary that the modified training set is a subset of the original.

The approach taken in Kohonen's LVQ is to construct a modified training set iteratively. Following Kohonen, we call the modified training set the *codebook*. This procedure tries to represent the decision boundaries rather than the class distributions.

The original procedure LVQ1 uses the following update rule. A example \mathbf{x} is presented. The nearest codebook vector to \mathbf{x} , \mathbf{m}_c , is updated by

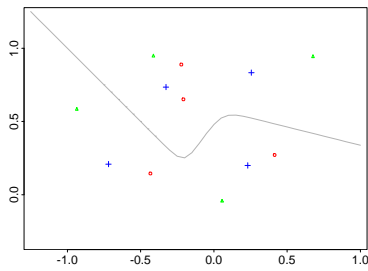
$$\mathbf{m}_c \leftarrow \mathbf{m}_c + \alpha(t)[\mathbf{x} - \mathbf{m}_c]$$

if \mathbf{x} is classified correctly by \mathbf{m}_c

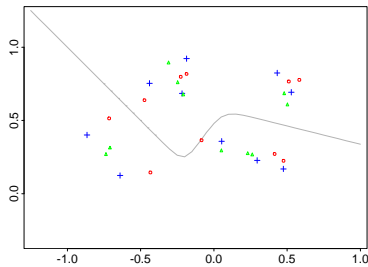
$$\mathbf{m}_c \leftarrow \mathbf{m}_c - \alpha(t)[\mathbf{x} - \mathbf{m}_c]$$

if \mathbf{x} is classified incorrectly

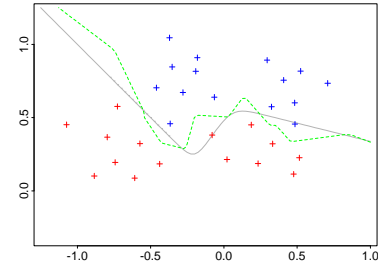
and all other codebook vectors are unchanged. Initially $\alpha(t)$ is chosen smaller than 0.1 and it is reduced linearly to zero during the fixed number of iterations.



Results of learning vector quantization. The initially chosen codebook is shown by small circles, the result of OLVQ1 by + and subsequently applying 25,000 passes of LVQ2.1 by triangles. The known decision boundary of the Bayes rule is also shown.



Further results of LVQ with a larger codebook. This time the triangles show the results from LVQ3.



Result of OLVQ1 with codebook of size 30. The decision boundary is shown as a dashed line.

Comparative Studies

Comparative studies often only show how well a particular class of naïve users might do with particular methods. The most comprehensive study is that published by Michie *et al.*, yet

In the trials reported in this book, we used the nearest neighbour ($k = 1$) classifier with no condensing. . . . Distances were scaled using the standard deviation for each attribute, with the calculation conditional on the class. [p. 36]

Although this method did very well on the whole, as expected it was the slowest of all for the very large datasets. However, it is known that substantial time saving can be effected, at the expense of some slight loss of accuracy, by using a condensed version of the training data. . . . It is clear that substantial improved accuracy can be obtained with careful choice of variables, but the current implementation is far too slow.

When scaling of attributes is not important, such as in object recognition databases, k -NN is first in the trials. Yet the explanatory power of k -NN might be said to be very small. [p. 216]

So, despite being the simplest of all of the methods in the comparison, it was implemented in a cursory way and still came out as one of the best algorithms, consistently beating neural networks by a large margin. In the hands of a less inexperienced user it would do better; in particular editing helps greatly with overlapping classes as in our synthetic example.

A Forensic Example

Data on 214 fragments of glass. Each has a measured refractive index and composition (weight percent of oxides of Na, Mg, Al, Si, K, Ca, Ba and Fe).

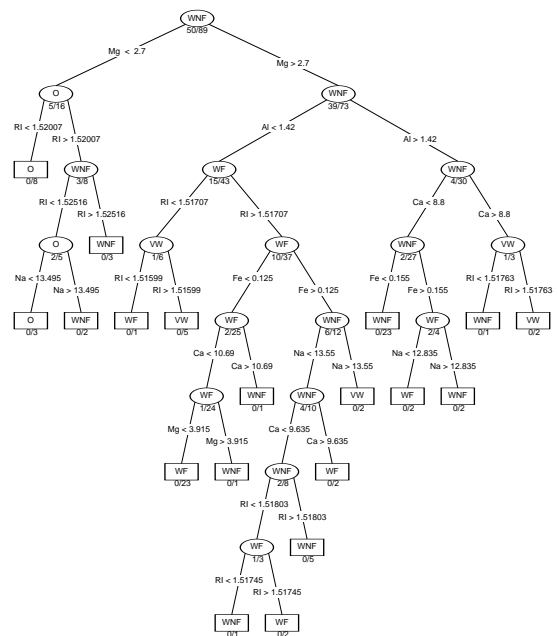
Grouped as window float glass (70), window non-float glass (76), vehicle window glass (17) and other (containers, tableware, headlamps) (22).

Used 89 to train, 96 to test, excluding headlamps.

methods	error rates %
linear discriminant	41 22
nearest neighbour	26 17
neural network with 2 hidden units	38 16
ditto, averaged over fits	38 14
neural network with 6 hidden units	33 15
ditto, averaged over fits	28 12
classification tree	28 15

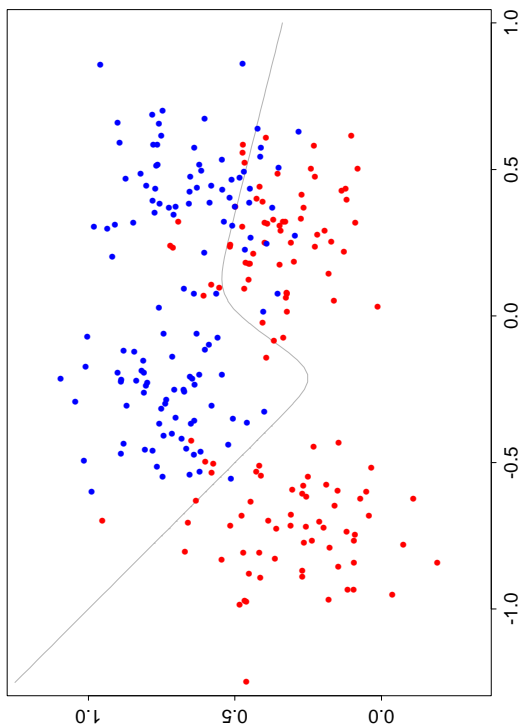
In the second column of error rates confusion between the two types of window glass is allowed.

20



Classification tree for the forensic glass example. At each node the majority classification is given, together with the error rate amongst training-set cases reaching that node.

21



A synthetic example of 250 points, 125 of each class. The best possible decision boundary is shown as a continuous curve: this is only known for synthetic problems.