Since, by (8)

$$p(\theta_i \mid \lambda_i) = \frac{p(X_i \mid \theta_i)}{p(X_i \mid \lambda_{i-1})} p(\theta_i \mid \lambda_{i-1}), \tag{11}$$

expression (10) is equivalent to the following:

$$p(\theta_k \mid \lambda_{k-1}) = \int \cdots \int \left[\prod_{i=1}^{M} p(\theta_{k-i} \mid \lambda_{k-i}) \right] \cdot p(\theta_k \mid \theta_{k-1}, \cdots, \theta_{k-M})\, d\theta_{k-1} \cdots d\theta_{k-M}. \tag{12}$$

But

$$\begin{aligned} p(\theta_k \mid \lambda_{k-1}) &= \int \cdots \int p(\theta_k, \cdots, \theta_{k-M} \mid \lambda_{k-1})\, d\theta_{k-1} \cdots d\theta_{k-M} \\ &= \int \cdots \int p(\theta_{k-1}, \cdots, \theta_{k-M} \mid \lambda_{k-1}) \\ &\quad \cdot p(\theta_k \mid \theta_{k-1}, \cdots, \theta_{k-M})\, d\theta_{k-1} \cdots d\theta_{k-M}. \end{aligned} \tag{13}$$

Thus, if (12) and therefore (10) is to hold independent of the shape of $p(\theta_k|\theta_{k-1}, \cdots, \theta_{k-M})$, one must have

$$p(\theta_{k-1}, \cdots, \theta_{k-M} \mid \lambda_{k-1}) = \prod_{i=1}^{M} p(\theta_{k-i} \mid \lambda_{k-i}), \tag{14}$$

which is a pathological situation and not true in general for $M > 1$. Note, however, that if $M = 1$, (14) is an identity and (12) does hold, but in this case, Fralick's expression (10) reduces to ours (4). Thus, the results (3) and (4) provide a correction to Fralick's expression for $M > 1$, and (5) and (6) give a similar recursive expression for $p(\theta_k \mid \lambda_k)$.

The question then arises of how actually to implement these iterative expressions, i.e., of how to store a function such as $p(\theta_k, \cdots, \theta_{k-M+1} \mid \lambda_{k-1})$. For their implementation, some finite parameterization of the equations must be found. Perhaps the simplest parameterization is to restrict θ_k to take on a finite set of values[1] (or be so approximated). The densities for θ_k, etc. are then replaced by probabilities, and the integrals by finite sums.

C. G. Hilborn, Jr.
D. G. Lainiotis
Dept. of Elec. Engrg.
The University of Texas
Austin, Tex.

References

[1] S. C. Fralick, "Learning to recognize patterns without a teacher," Stanford Electronics Laboratories, Stanford, Calif., Tech. Rept. 6103-10, March, 1965. (A brief version appears in *IEEE Trans. Information Theory*, vol. IT-13, pp. 57–64, January 1967.)
[2] C. G. Hilborn, Jr., and D. G. Lainiotis, "Optimal adaptive pattern recognition," *Proc. 1st Annual Princeton Conf. Information Sciences and Systems*, March 30–31, 1967.

[1] This technique is used in Hilborn and Lainiotis.[2]

The Condensed Nearest Neighbor Rule

The purpose of this note is to introduce the condensed nearest neighbor decision rule (CNN rule) and to pose some unsolved theoretical questions which it raises. The CNN rule, one of a class of *ad hoc* decision rules which have appeared in the literature in the past few years, was motivated by statistical considerations pertaining to the nearest neighbor decision rule (NN rule). We briefly review the NN rule and then describe the CNN rule.

The NN rule[1]–[4] assigns an unclassified sample to the same class as the nearest of n stored, correctly classified samples. In other words, given a collection of n reference points, each classified by some external source, a new point is assigned to the same class as its nearest neighbor. The most interesting theoretical property of the NN rule is that under very mild regularity assumptions on the underlying statistics, for any metric, and for a variety of loss functions, the large-sample risk incurred is less than twice the Bayes risk. (The Bayes decision rule achieves minimum risk but requires complete knowledge of the underlying statistics.) From a practical point of view, however, the NN rule is not a prime candidate for many applications because of the storage requirements it imposes. The CNN rule is suggested as a rule which retains the basic approach of the NN rule without imposing such stringent storage requirements.

Before describing the CNN rule we first define the notion of a consistent subset of a sample set. This is a subset which, when used as a stored reference set for the NN rule, correctly classifies all of the remaining points in the sample set. A minimal consistent subset is a consistent subset with a minimum number of elements. Every set has a consistent subset, since every set is trivially a consistent subset of itself. Obviously, every finite set has a minimal consistent subset, although the minimum size is not, in general, achieved uniquely. The CNN rule uses the following algorithm to determine a consistent subset of the original sample set. In general, however, the algorithm will not find a minimal consistent subset. We assume that the original sample set is arranged in some order; then we set up bins called STORE and GRABBAG and proceed as follows.

1) The first sample is placed in STORE.

2) The second sample is classified by the NN rule, using as a reference set the current contents of STORE. (Since STORE has only one point, the classification is trivial at this stage.) If the second sample is classified correctly it is placed in GRABBAG; otherwise it is placed in STORE.

3) Proceeding inductively, the ith sample is classified by the current contents of STORE. If classified correctly it is placed in GRABBAG; otherwise it is placed in STORE.

4) After one pass through the original sample set, the procedure continues to loop through GRABBAG until termination, which can occur in one of two ways:

a) The GRABBAG is exhausted, with all its members now transferred to STORE (in which case, the consistent subset found is the entire original set), or

b) One complete pass is made through GRABBAG with no transfers to STORE. (If this happens, all subsequent passes through GRABBAG will result in no transfers, since the underlying decision surface has not been changed.)

5) The final contents of STORE are used as reference points for the NN rule; the contents of GRABBAG are discarded.

Qualitatively, the rule behaves as follows: If the Bayes risk is small, i.e., if the underlying densities of the various classes have small overlap, then the algorithm will tend to pick out points near the (perhaps fuzzy) boundary between the classes. Typically, points deeply imbedded within a class will not be transferred to STORE, since they will be correctly classified. If the Bayes risk is high, then STORE will contain essentially all the points in the original sample set, and no important reduction in sample size will have been achieved. No theoretical properties of the CNN rule have been established.

The CNN rule has been tried on a number of problems, both real and artificial. In order to investigate the behavior of the rule when the classes are (essentially) disjoint—the case in which the CNN rule is of greatest interest—several experiments similar to the following were run. The underlying probability structure for a two-class problem was assumed to consist of two probability densities, each a uniform distribution on the supports shown in Fig. 1. The set of all vectors with integer components lying within each

Manuscript received November 24, 1966; revised October 5, 1967.

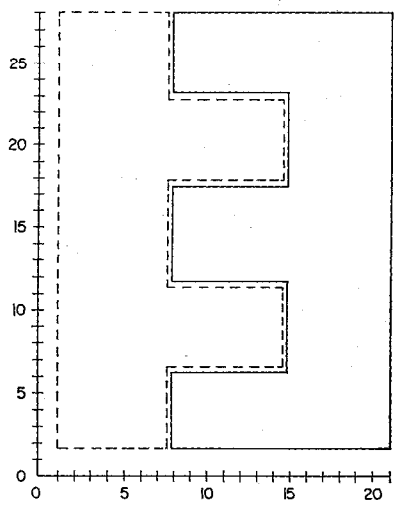

Fig. 1. Class boundaries.

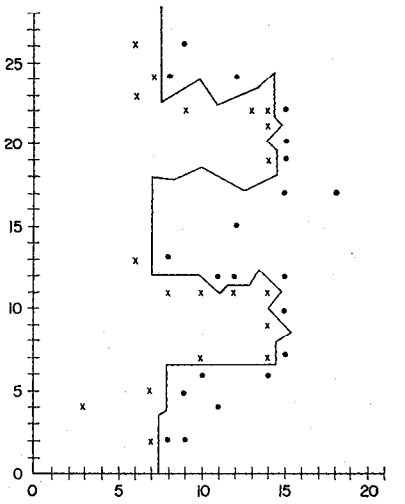

Fig. 2. Samples selected and induced decision surface.

support was taken to simulate a random sampling from each population. The 482 points thus obtained were ordered by a random mechanism and processed using the algorithm described above.

The algorithm terminated after four interations through GRABBAG, at which time STORE contained 40 samples. Fig. 2 shows the final 40 samples and the decision surface induced by the NN rule using these 40 samples as a stored reference set.

Since all samples had integer-valued components, ties occurred with nonzero probability, and these were broken arbitrarily. This accounts for the fact that occasionally the decision surface lies properly within one or the other of the supports rather than between them. The points most deeply imbedded within each class were the first two points in the random ordering.

A more realistic experiment was performed using data supplied by Nagy of IBM.[5] This data consisted of approximately 12 000 96-dimensional binary vectors drawn from 25 different statistical populations. (The data represent upper-case typewritten characters, excluding "I," typed with nine different styles of fonts.) The 12 000 samples were divided into a training set and a testing set of approximately equal size, and the CNN algorithm was used on the training set. The algorithm terminated after four iterations through GRABBAG, at which time STORE contained 197 of the original 6295 samples. An error rate of 1.28 percent was obtained on the independent test set. This was somewhat disappointing in view of the fact that a number of simpler classifiers (the ternary reference classifier,[5] linear machine,[6] and piecewise-linear machine[6]), using considerably less computer time, achieved error rates on the order of 0.3-0.5 percent.[7],[8] It was also a little surprising, since (necessarily) the 197 stored points correctly classified all the 6295 samples in the training set.

These and similar experiments have persuaded us that the CNN rule offers interesting possibilities, but that a great deal more work of both a theoretical and experimental nature will be needed before the rule is thoroughly understood. For example, under suitably restrictive assumptions on the underlying statistics:

1) What is the expected number of iterations before termination?
2) What is the expected reduction in the size of the stored sample set?
3) What is the expected increase in CNN risk over NN risk for a sample set of given size?

In view of the desirable theoretical properties of the k-NN rule,[1],[2]—the rule that makes a decision on the basis of votes cast by each of the k nearest neighbors—we pose a final obvious question which should, perhaps, be answered experimentally. How would the CNN rule perform if the vote of, say, the three nearest neighbors was substituted for the decision of the single nearest neighbor everywhere in the algorithm?

PETER E. HART
Applied Physics Lab.
Stanford Research Institute
Menlo Park, Calif. 94025

REFERENCES

[1] P. E. Hart, "An asymptotic analysis of the nearest-neighbor decision rule," Stanford Electronics Labs., Stanford, Calif., Tech. Rept. 1828-2 (SEL-66-016), May 1966.
[2] T. M. Cover and P. E. Hart, "Nearest-neighbor pattern classification," *IEEE Trans. Information Theory*, vol. IT-13, pp. 21–27, January 1967.
[3] T. M. Cover, "Estimation by the nearest-neighbor rule," *IEEE Trans. Information Theory*, vol. IT-14, pp. 50–55, January 1968.
[4] A. W. Whitney and S. J. Dwyer, III, "Performance and implementation of the k-nearest neighbor decision rule with incorrectly identified training samples," *1966 Proc. 4th Allerton Conf. Circuit and System Theory.*
[5] C. N. Liu and G. L. Shelton, Jr., "An experimental investigation of a mixed-font print recognition system," *IEEE Trans. Electronic Computers*, vol. EC-15, pp. 916-925, December 1966.
[6] N. J. Nilsson, *Learning Machines-Foundations of Trainable Pattern Classifying Systems.* New York: McGraw-Hill, 1965.
[7] R. G. Casey *et al.*, "An experimental comparison of several design algorithms used in pattern recognition," IBM Corp., Research Rept. RC 1500, November 1965.
[8] D. S. Nee, "Multifont character-recognition experiments using trainable classifiers," Stanford Research Institute, Menlo Park, Calif., Tech. Note 1, Contract AF 30(602)-3945, August 1966.

Uncertainty and the Probability of Error

Let X and Y be discrete random variables which can be thought of as the input and output, respectively, of a communication channel. Let X and Y take on the values $\{x_i: i = 1, \cdots, m\}$ and $\{y_i: i = 1, \cdots, n\}$, respectively, where $n \geq m$. A decision rule for X in terms of Y can be considered as a partition $\{A_i: i = 1, \cdots, m\}$ such that $A_i \cap A_j = \phi$, $i \neq j$, and $\cup_{i=1}^{m} A_i = \{y_j: j = 1, \cdots, n\}$ where the decision is x_i if $Y \in A_i$. This also defines a "post-decision" random variable Z, where Z is defined by $Z = z_i$ if $Y \in A_i$, $i = 1, \cdots, m$.

Two putative measures of the efficiency of this system are uncertainty (or equivocation) and probability of error. It is desirable to determine the relationship between these two measures. In particular, we can compare $H(X|Y)$ with the minimum probability of error $P_0(e)$ if we want to evaluate the channel independent of the decision rule. Otherwise we can compare, given a particular decision rule, $H(X|Z)$ with the probability of error $P(e)$. The purpose of the paper is to demonstrate the exact relationship between $H(X|Y)$ and $P_0(e)$.

First, we relate $H(X|y_k)$ to $P_0(e|y_k)$ for each k. Now $P_0(e|y_k) = 1 - \max_i P(x_i|y_k)\}$, and letting y_k be fixed, we donete $P_i = P(x_i|y_k)$, $i = 1, \cdots, m$, such that $P_1 \geq P_i$, $i = 2, \cdots, m$. Then $P_0(e|y_k) =$

Manuscript received September 13, 1966; revised October 18, 1967. The work of D. L. Tebbe was supported by NASA traineeship; the work of S. J. Dwyer, III, was supported in part by the Missouri Regional Medical Program grant of U. S. Public Health.