A Fuzzy *K*-Nearest Neighbor Algorithm

JAMES M. KELLER, MICHAEL R. GRAY, AND JAMES A. GIVENS, JR.

Abstract—**Classification of objects is an important area of research and application in a variety of fields. In the presence of full knowledge of the underlying probabilities, Bayes decision theory gives optimal error rates. In those cases where this information is not present, many algorithms make use of distance or similarity among samples as a means of classification. The *K*-nearest neighbor decision rule has often been used in these pattern recognition problems. One of the difficulties that arises when utilizing this technique is that each of the labeled samples is given equal importance in deciding the class memberships of the pattern to be classified, regardless of their "typicalness." The theory of fuzzy sets is introduced into the *K*-nearest neighbor technique to develop a fuzzy version of the algorithm. Three methods of assigning fuzzy memberships to the labeled samples are proposed, and experimental results and comparisons to the crisp version are presented. In fact, not only does the fuzzy algorithm dominate its counterpart in terms of a lower error rate, the resulting memberships give a confidence measure of the classification. The fuzzy *K*-NN rule is also shown to compare well against other standard, more-sophisticated pattern recognition procedures in these experiments. A fuzzy analog of the nearest prototype algorithm is also developed.**

I. Introduction

Classification of objects is an important area of research and of practical applications in a variety of fields, including pattern recognition and artificial intelligence, statistics, cognitive psychology, vision analysis, and medicine [1]–[10]. Considered as a pattern recognition problem, there have been numerous techniques investigated for classification. Clearly, the more *a priori* information that is known about the problem domain, the more the classification algorithm can be made to reflect the actual situation. For example, if the *a priori* probabilities and the state conditional densities of all classes are known, then Bayes decision theory produces optimal results in the sense that it minimizes the expected misclassification rate [3]. However, in many pattern recognition problems, the classification of an input pattern is based on data where the respective sample sizes of each class are small and possibly not representative of the actual probability distributions, even if they are known. In these cases, many techniques rely on some notion of similarity or distance in feature space, for instance, clustering and discriminant analysis [2], [3]. Under many circumstances, the *K*-nearest neighbor (*K*-NN) algorithm [3], [11] is used to perform the classification. This decision rule provides a simple nonparametric procedure for the assignment of a class label to the input pattern based on the class labels represented by the *K*-closest (say, for example, in the Euclidean sense) neighbors of the vector.

The *K*-NN rule is a suboptimal procedure. However, it has been shown that in the infinite sample situation, the error rate for the 1-NN rule is bounded above by no more than twice the optimal Bayes error rate and, that as *K* increases, this error rate approaches the optimal rate asymptotically [11], [12]. Since its introduction, the *K*-NN rule has been studied and improved upon by numerous researchers [13]–[18]. But it is not this asymptotic behavior in the limit that has maintained interest in this family of decision rules, but rather their computational simplicity and the perhaps surprising good results obtained by their use in many problems of small sample size [19]–[22]. It has been found for example that *K*-NN classification is well suited to those problem domains characterized by data that is only partially exposed to the system prior to employment [21], [22].

One of the problems encountered in using the *K*-NN classifier is that normally each of the sample vectors is considered equally important in the assignment of the class label to the input vector. This frequently causes difficulty in those places where the sample sets overlap. Atypical vectors are given as much weight as those that are truly representative of the clusters. Another difficulty is that once an input vector is assigned to a class, there is no indication of its "strength" of membership in that class. It is these two problems in the *K*-NN algorithm that we address by incorporating fuzzy set theory into the *K*-NN rule.

Fuzzy sets were introduced by Zadeh in 1965 [23]. Since that time researchers have found numerous ways to utilize this theory to generalize existing techniques and to develop new algorithms in pattern recognition and decision analysis [24]–[27]. In [24] Bezdek suggests that interesting and useful algorithms could result from the allocation of fuzzy class membership to the input vector, thus affording fuzzy decisions based on fuzzy labels. This work is concerned with incorporating fuzzy set methods into the classical *K*-NN decision rule. In particular, a "fuzzy *K*-NN" algorithm is developed utilizing fuzzy class memberships of the sample sets and thus producing a fuzzy classification rule. Three methods of assigning fuzzy membership for the training sets are proposed, and their advantages and disadvantages are discussed. Results of both the "crisp" (that based on traditional set theory) and fuzzy *K*-NN rule are compared on two data sets, and the fuzzy algorithm is shown to dominate its crisp counterpart by having lower error rates and by producing membership values that serve as a confidence measure in the classification.

Finally, a simple variant of the *K*-NN rule, the nearest prototype technique, is considered. In this decision scheme, a typical pattern of each class is chosen, and the unknown vector is assigned to the class of its closest prototype. A fuzzy analog to this procedure is developed and the results of the two versions are compared.

II. Fuzzy Sets

Given a universe U of objects, a conventional crisp subset A of U is commonly defined by specifying the objects of the universe that are members of A. An equivalent way of defining A is to specify the characteristic function of A, u_A: $U \rightarrow \{0,1\}$ where for all $x \in U$

$$u_A(x) = \begin{cases} 1, & x \in A \\ 0, & x \notin A. \end{cases}$$

Fuzzy sets are derived by generalizing the concept of a characteristic function to a membership function u: $U \rightarrow [0,1]$. An example of a fuzzy set is the set of real numbers much larger than zero, which can be defined with a membership function as follows:

$$u(x) = \begin{cases} x^2/(x^2+1), & x \geqslant 0 \\ 0, & x < 0. \end{cases}$$

Numbers that are not at all larger than zero are not in the set ($u = 0$), while numbers which are larger than zero are partially in the set based on how much larger than zero they are. Thus the impetus behind the introduction of fuzzy set theory was to provide a means of defining categories that are inherently imprecise [24]. Since the introduction of fuzzy set theory the terms *hard* and *crisp* have been used to describe sets conforming to traditional set theory.

Most crisp set operations (such as union and intersection) and set properties have analogs in fuzzy set theory. (See [28] for a more detailed presentation of fuzzy set theory.)

The advantage provided by fuzzy sets is that the degree of membership in a set can be specified, rather than just the binary

Manuscript received September 1, 1984; revised February 26, 1985.

J. Keller and J. Givens are with the Department of Electrical and Computer Engineering, University of Missouri, Columbia, MO 65211, USA.

M. Gray was in the Department of Electrical and Computer Engineering, University of Missouri. He is now with Wright-Patterson Air Force Base, OH 45433, USA.

0018-9472/85/0700-0580$01.00

is or isn't a member. This can be especially advantageous in pattern recognition, where frequently objects are not clearly members of one class or another. Using crisp techniques an ambiguous object will be assigned to one class only, lending an aura of precision and definiteness to the assignment that is not warranted. On the other hand, fuzzy techniques will specify to what degree the object belongs to each class, which is information that frequently is useful.

Given a set of sample vectors, $\{x_1, \cdots, x_n\}$, a fuzzy c partition of these vectors specifies the degree of membership of each vector in each of c classes. It is denoted by the c by n matrix U, where $u_{ik} = u_i(x_k)$ for $i = 1, \cdots, c$, and $k = 1, \cdots, n$ is the degree of membership of x_k in class i. The following properties must be true for U to be a fuzzy c partition

$$\sum_{i=1}^{c} u_{ik} = 1,$$

$$0 < \sum_{k=1}^{m} u_{ik} < n,$$

$$u_{ik} \epsilon [0,1].$$

The fact that a vector's memberships in the c classes must sum to one is for mathematical tractability. In the two class case for example, memberships near 0.5 indicate that the vector has a high degree of membership in both classes; i.e., the "bounding region" separates one class from another.

III. The K-Nearest Neighbor Algorithms

The nearest neighbor classifiers require no preprocessing of the labeled sample set prior to their use. The crisp nearest-neighbor classification rule assigns an input sample vector y, which is of unknown classification, to the class of its nearest neighbor [11]. This idea can be extended to the K-nearest neighbors with the vector y being assigned to the class that is represented by a majority amongst the K-nearest neighbors. Of course, when more than one neighbor is considered, the possibility that there will be a tie among classes with a maximum number of neighbors in the group of K-nearest neighbor exists. One simple way of handling this problem is to restrict the possible values of K. For example, given a two-class problem, if we restrict K to odd values only no tie will be possible. Of course, when more than two classes are possible, this technique is not useful. A means of handling the occurrence of a tie is as follows. The sample vector is assigned to the class, of those classes that tied, for which the sum of distances from the sample to each neighbor in the class is a minimum. Of course, this could still lead to a tie, in which case the assignment is to the last class encountered amongst those which tied, an arbitrary assignment. Clearly, there will be cases where a vector's classification becomes an arbitrary assignment, no matter what additional procedures are included in the algorithm.

A. The Crisp K-NN Algorithm

Let $W = \{x_1, x_2, \cdots, x_n\}$ be a set of n labeled samples. The algorithm is as follows:

```
BEGIN
  Input y, of unknown classification.
  Set K, 1 ≤ K ≤ n.
  Initialize i = 1.
  DO UNTIL (K-nearest neighbors found)
    Compute distance from y to x_i.
    IF (i ≤ K) THEN
      Include x_i in the set of K-nearest neighbors
    ELSE IF (x_i is closer to y than any previous nearest
      neighbor) THEN
      Delete farthest in the set of K-nearest neighbors
      Include x_i in the set of K-nearest neighbors.
    END IF
    Increment i.
  END DO UNTIL
  Determine the majority class represented in the set of K-
  nearest neighbors.
  IF (a tie exists) THEN
    Compute sum of distances of neighbors in each class
    which tied.
    IF (no tie occurs) THEN
      Classify y in the class of minimum sum
    ELSE
      Classify y in the class of last minimum found.
    END IF
  ELSE
    Classify y in the majority class.
  END IF
END
```

B. Fuzzy K-NN Classifier

While the fuzzy K-nearest neighbor procedure is also a classification algorithm the form of its results differ from the crisp version. The fuzzy K-nearest neighbor algorithm assigns class membership to a sample vector rather than assigning the vector to a particular class. The advantage is that no arbitrary assignments are made by the algorithm. In addition, the vector's membership values should provide a level of assurance to accompany the resultant classification. For example, if a vector is assigned 0.9 membership in one class and 0.05 membership in two other classes we can be reasonably sure the class of 0.9 membership is the class to which the vector belongs. On the other hand, if a vector is assigned 0.55 membership in class one, 0.44 membership in class two, and 0.01 membership in class three, then we should be hesitant to assign the vector based on these results. However, we can feel confident that it does not belong to class three. In such a case the vector might be examined further to determine its classification, because the vector exhibits a high degree of membership in both classes one and two. Clearly the membership assignments produced by the algorithm can be useful in the classification process.

The basis of the algorithm is to assign membership as a function of the vector's distance from its K-nearest neighbors and those neighbors' memberships in the possible classes. The fuzzy algorithm is similar to the crisp version in the sense that it must also search the labeled sample set for the K-nearest neighbors. Beyond obtaining these K samples, the procedures differ considerably.

Let $W = \{x_1, x_2, \cdots, x_n\}$ be the set of n labeled samples. Also let $u_i(x)$ be the assigned membership of the vector x (to be computed), and u_{ij} be the membership in the ith class of the jth vector of the labeled sample set. The algorithm is as follows:

```
BEGIN
  Input x, of unknown classification.
  Set K, 1 ≤ K ≤ n.
  Initialize i = 1.
  DO UNTIL (K-nearest neighbors to x found)
    Compute distance from x to x_i.
    IF (i ≤ K) THEN
      Include x_i in the set of K-nearest neighbors
    ELSE IF (x_i closer to x than any previous nearest neigh-
    bor) THEN
      Delete the farthest of the K-nearest neighbors
      Include x_i in the set of K-nearest neighbors.
    END IF
  END DO UNTIL
  Initialize i = 1.
  DO UNTIL (x assigned membership in all classes)
    Compute u_i(x) using (1).
    Increment i.
  END DO UNTIL
END
```

where

$$u_i(x) = \frac{\sum_{j=1}^{K} u_{ij}\left(1/\|x - x_j\|^{2/(m-1)}\right)}{\sum_{j=1}^{K}\left(1/\|x - x_j\|^{2/(m-1)}\right)}. \quad (1)$$

As seen by (1), the assigned memberships of x are influenced by the inverse of the distances from the nearest neighbors and their class memberships. The inverse distance serves to weight a vector's membership more if it is closer and less if it is farther from the vector under consideration. The labeled samples can be assigned class memberships in several ways. First, they can be given complete membership in their known class and nonmembership in all other classes. Other alternatives are to assign the samples' membership based on distance from their class mean or based on the distance from labeled samples of their own class and those of the other class or classes, and then to use the resulting memberships in the classifier. Both of these techniques have been used in this study and the results are reported. It is noted that in [29] an alternate scheme for assigning initial memberships based on a learning scheme was considered.

The variable m determines how heavily the distance is weighted when calculating each neighbor's contribution to the membership value. If m is two, then the contribution of each neighboring point is weighted by the reciprocal of its distance from the point being classified. As m increases, the neighbors are more evenly weighted, and their relative distances from the point being classified have less effect. As m approaches one, the closer neighbors are weighted far more heavily than those farther away, which has the effect of reducing the number of points that contribute to the membership value of the point being classified. In the results presented in Section V we used $m = 2$, but note that almost equal error rates have been obtained on these data over a wide range of values of m.

IV. Nearest Prototype Classifiers

These classifiers bear a marked resemblance to the one-nearest neighbor classifier. Actually, the only difference is that for the nearest prototype classifier the labeled samples are a set of class prototypes, whereas in the nearest neighbor classifier we use a set of labeled samples that are not necessarily prototypical. Of course, the nearest prototype classifier could be extended to multiple prototypes representing each class, similar to the K-nearest neighbor routine. Nevertheless, this study considers only the nearest prototype classifier in both a crisp and fuzzy version. The prototypes used for these routines are taken as the class means of the labeled sample set.

A. The Crisp Nearest Prototype Classifier

Let $W = \{Z_1, Z_2, \cdots, Z_c\}$ be the set of c prototype vectors representing the c classes. The algorithm is as follows:

```
BEGIN
  Input x, vector to be classified.
  Initialize i = 1
  DO UNTIL (distance from each prototype to x computed)
    Compute distance from Z_i to x.
    Increment i.
  END DO UNTIL
  Determine minimum distance to any class prototype.
  IF (tie exists) THEN
    Classify x as last class found of minimum distance
  ELSE
    Classify x as class of closest prototype.
  END IF
END
```

B. Fuzzy Nearest Prototype Algorithm

As above, let $W = \{Z_1, Z_2, \cdots, Z_c\}$ be the set of c prototypes representing the c classes. The algorithm is as follows:

```
BEGIN
  Input x, vector to be classified.
  Initialize i = 1.
  DO UNTIL (distance from each prototype to x computed)
    Compute distance from Z_i to x.
    Increment i.
  END DO UNTIL
  Initialize i = 1.
  DO UNTIL (x assigned membership in all classes)
    Compute u_i(x) using (2)
    Increment i
  END DO UNTIL
END
```

where

$$u_i(x) = \frac{1/\|x - Z_i\|^{2/(m-1)}}{\sum_{j=1}^{c}\left(1/\|x - Z_j\|^{2/(m-1)}\right)}. \quad (2)$$

The difference between (2) and (1) is that membership in each class is assigned based only on the distance from the prototype of the class. This is because the prototypes should naturally be assigned complete membership in the class that they represent.

V. Results

The results presented in this section were produced by software implementation of the algorithms described above. The software was developed using Fortran 77 on a Perkin-Elmer 3220 in the Image Analysis Laboratory at the University of Missouri-Columbia.

Three labeled data sets were utilized to test the algorithms. The data sets and their attributes are as follows:

Data Set Name	Number of Classes	Number of Vectors	Number of Features per Vector
IRIS	3	150	4
IRIS23	2	100	4
TWOCLASS	2	242	4

The data set IRIS is that of Anderson. This particular data set has been utilized extensively by researchers in the area of cluster analysis since 1936, when R. A. Fisher first used it to illustrate the concept of linear discriminant analysis [30]. The data represents three subspecies of irises, with the four feature measurements being sepal length, sepal width, petal length, and petal width, all in centimeters. There are fifty vectors per class in this data set. The IRIS23 data set is a subset of the IRIS data. It includes classes two and three, the nonseparable classes, of the IRIS data.

The TWOCLASS data set is an artificially generated normally distributed set of vectors. This data set was included because classification results from a Bayes classifier were available to use in the comparison. This data set contains 121 samples per class.

The results of the fuzzy classifications are reported in terms of the simplest crisp partition, where a sample vector is assigned to the class of maximum membership. The classifications are obtained using the "leave one out" technique. The procedure is to leave one sample out of the data set and classify it using the remaining samples as the labeled data set. This technique is repeated until all samples in the data set have been classified. In addition, in order to evaluate one technique used to initialize memberships of the labeled samples used in the classifier the IRIS23 set was created by using only class two and three of the

TABLE I
RESULTS OF K-NEAREST NEIGHBOR CLASSIFIERS
NUMBER OF MISCLASSIFIED VECTORS[1]

	Crisp			Fuzzy-(1)			Fuzzy-(2)		Fuzzy-(3)		
K	I	T	I'	I	T	I'	T	I'	I	T	I'
1	6	26	6	6	26	6	26	6	6	26	6
2	7	26	7	6	26	6	21	6	6	21	6
3	6	21	6	6	22	6	21	7	5	19	6
4	5	20	5	6	19	6	20	7	5	20	5
5	5	20	5	5	21	5	20	7	4	19	4
6	6	19	6	5	18	5	20	6	4	20	4
7	5	19	5	5	21	5	18	6	4	19	4
8	7	21	7	6	18	6	20	6	4	20	4
9	6	21	6	4	21	4	18	5	4	18	4
Average	5.9	21.4	5.9	5.4	21.3	5.4	20.4	6.2	4.7	20.2	4.8

[1] K number of neighbors used
I IRIS data (four features)
T TWOCLASS data (four features)
I' IRIS23 data (four features)
(1) crisp initialization
(2) exponential initialization
(3) fuzzy 3-nearest neighbor initialization.

IRIS data set. This was necessary because the initialization technique will only work on two class classification problems.

Before comparing the results produced by the nearest neighbor algorithms, the types of labeling techniques used for the fuzzy classifier are explained. Three different techniques of membership assignment for the labeled data are considered. The first method, a crisp labeling, is to assign each labeled sample complete membership in its known class and zero membership in all other classes. The second technique assigns membership based on the procedure presented in [31]. This technique works only on two class data sets. The procedure assigns a sample membership in its known class based on its distance from the mean of the labeled sample class. These memberships range from one to one half with an exponential rate of change between these limits. The sample's membership in the other class is assigned such that the sum of the memberships of the vector equals one. A more detailed explanation of this technique is given in [31]. The third method assigns memberships to the labeled samples according to a K-nearest neighbor rule. The K (not K of the classifier)-nearest neighbors to each sample x (say x in class i) are found, and then membership in each class is assigned according to the following equation:

$$u_j(x) = \begin{cases} 0.51 + (n_j/K) * 0.49, & \text{if } j = i \\ (n_j/K) * 0.49, & \text{if } j \neq i. \end{cases}$$

The value n_j is the number of the neighbors found which belong to the jth class. This method attempts to "fuzzify" the memberships of the labeled samples, which are in the class regions intersecting in the sample space, and leaves the samples that are well away from this area with complete membership in the known class. As a result, an unknown sample lying in this intersecting region will be influenced to a lesser extent by the labeled samples that are in the "fuzzy" area of the class boundary.

Thus with these three initialization techniques, three sets of results of the fuzzy K-nearest neighbor classifier are produced. These results are presented in Tables I and II. Upon comparison of the results of the crisp classifier and the fuzzy classifier with crisp initialization, we can see that on the average the fuzzy classifier has slightly lower error rates. In addition, the fuzzy classifier, which uses the second initialization technique, produced nearly equal results. Although not reported in the tables, the results of this fuzzy classifier using the membership assignment rule described in [31] did not produce memberships for the misclassified vectors that suggest that they actually belong to a different class. Instead this second initialization technique causes an overall reduction in the values of memberships assigned with most of the samples given majority memberships less than 0.7. But the nearest neighbor initialization technique does produce membership assignments that give an indication of degree of correctness of classification.

TABLE II
COMPARISON OF CRISP AND FUZZY K-NN CLASSIFIERS ON TWOCLASS DATA AND ON IRIS DATA WITH FUZZY KINIT-NN INITIALIZATION

	Number of Misclassified Vectors (out of 242)					
		Fuzzy KINIT				
K	Crisp	1	3	5	7	9
	TWOCLASS Data					
1	26	26	26	26	26	26
2	26	23	21	23	22	22
3	21	20	19	21	21	23
4	20	17	20	19	19	19
5	20	16	19	19	20	19
6	19	20	20	20	21	20
7	19	17	19	20	20	20
8	21	17	20	20	20	20
9	21	18	21	21	21	21
Average Misclassification	21.4	19.3	20.6	21.0	21.1	21.1
	IRIS Data					
1	6	6	6	6	6	6
2	7	6	6	6	6	6
3	6	5	5	5	5	6
4	5	5	5	5	5	5
5	5	4	4	5	5	5
6	6	4	4	4	4	4
7	5	4	4	4	4	4
8	7	4	4	4	4	4
9	6	4	4	4	4	4
Average Misclassification	5.9	4.7	4.7	4.8	4.8	4.9

TABLE III
CONFUSION MATRICES FOR THE K-MEANS ALGORITHM

	IRIS[1]			TWOCLASS[2]		
	1	2	3		1	2
1	50	0	0	1	114	7
2	0	48	2	2	15	106
3	0	14	36			

[1] Terminated in three iterations.
[2] Terminated in ten iterations.

Examining the results given in Table II for the K-nearest neighbor classifier with nearest neighbor sample membership initialization, the following observations can be made. First of all, the results show a somewhat lower overall error rate. But, more importantly, the number of misclassified vectors with high assigned membership (greater than 0.8) in the wrong class is considerably less than half of the misclassified vectors for most choices of KINIT. In addition, the correctly classified samples were given relatively higher membership in their known class than in other classes. Therefore, more sophisticated classification schemes utilizing these memberships (other than just maximum membership) could be devised to increase the overall correct classification rate, and the final membership values produce a natural confidence measure.

While the main concern of this paper was to demonstrate that the fuzzy K-NN technique dominates the crisp version in both

TABLE IV
CONFUSION MATRICES OF THE FIRST-NEAREST PROTOTYPE CLASSIFIER

IRIS Data

	Four Features						Features Three and Four					
	Crisp			Fuzzy			Crisp			Fuzzy		
	1	2	3	1	2	3	1	2	3	1	2	3
1	50	0	0	50	0	0	50	0	0	50	0	0
2	0	45	5	0	45	5	0	48	2	0	48	2
3	0	7	43	0	7	43	0	4	46	0	4	46

TWOCLASS Data

	Four Features				Features Three and Four			
	Crisp		Fuzzy		Crisp		Fuzzy	
	1	2	1	2	1	2	1	2
1	113	8	113	8	113	8	113	8
2	12	109	12	109	12	109	12	109

decreased error rates and information content of the results, our algorithm compares favorably to several other more complicated techniques on these data sets. In particular, we have run the *K*-means clustering procedure, two types of linear discriminant function algorithms, and a Bayes classifier on the data. For the *K*-means [3], we initialized the cluster centers to the sample mean and those vectors furthest from the mean. The results are shown in Table III. The fuzzy *K*-NN does at least as well as this procedure (better for the IRIS data) with the added information on class membership.

The perceptron is representative of a class of iterative schemes for finding linear decision boundaries between classes using the gradient descent approach. This procedure is guaranteed to converge to a solution if the data sets are linearly separable [3]. However, since neither of our data sets possess this property, the perceptron does not converge. Stopping it after a fixed number of iterations may or may not produce a reasonable decision boundary. For example, terminating it after two iterations on the TWOCLASS data, the linear discriminant result misclassified 62 of the 242 vectors; after 70 iterations, 107 vectors were misclassified; and after 150 iterations, the number of misclassified was down to 29 but by 200 iterations, it was back to 67. This of course demonstrates the erratic behavior of this algorithm on overlapping data. There have been approaches to modify this technique to produce reasonable boundaries even in the nonseparable case. One such method, using fuzzy sets, is reported in [31]. This technique converged in two iterations misclassifying 21 vectors, again comparable to the fuzzy *K*-NN. Similar results are obtained on the IRIS set.

While the primary use of the *K*-NN algorithms is in those situations where the *a priori* probabilities and class conditional densities are unknown, the TWOCLASS data was in fact generated with equal *a priori* probabilities and multivariate normal distributions. So as a final comparison, consider the results of the Bayes classifier for the TWOCLASS data. Running a ten-percent jacknife procedure[1] and assuming equal *a priori* probabilities for both classes, the Bayes classifier misclassified twenty of the samples. Clearly, depending on the value chosen for *K*, the fuzzy nearest neighbor classifier can perform as well as a Bayes classifier, but with much less restrictive assumptions. Certainly, the posterior probabilities in the Bayes classifier provide a measure of strength of classification; i.e., they represent the probability that the object is a member of each class. The fuzzy *K*-nearest neighbor classifier provides different form of this information. In this case, we obtain a measure of how "typical" this object is of each class.

[1]This procedure involves taking ten percent of the samples as test data and the remaining as training data, classifying these, and then repeating the procedure until all samples have been used as test samples.

TABLE V
FUZZY CLASSIFIER MEMBERSHIP ASSIGNMENTS[1]

	IRIS Data		TWOCLASS Data	
	A	*B*	*A*	*B*
Misclassified samples (membership assigned > 0.7)	1	1	3	3
Classified samples (membership assigned > 0.5 and < 0.7)	15	15	36	36

[1]The intent here is to illustrate that there are very few correctly classified samples in the "fuzzy" region between 0.5 and 0.7. *A* denotes four feature and *B* denotes features three and four used.

The nearest prototype classifier in both the crisp and fuzzy versions is the quickest and simplest of the classifiers considered. The reason is that in both versions of the first-nearest prototype algorithm, an unknown sample is compared to one prototype per class as opposed to the *K*-nearest neighbor algorithms, where an entire set of labeled samples representing each class must be compared before the *K* nearest are obtained. The results reported in Table IV show that the fuzzy prototype classifier and the crisp nearest prototype classifier produced equivalent results. But, by looking at the memberships of the misclassified samples in terms of the number with membership greater than 0.7 in the wrong class, given in Table V, it is clear that these memberships do provide a useful level of confidence of classification. Further, the number of correctly classified samples with memberships in the range between 0.4 and 0.7 is small compared to the number of correctly classified samples that have membership in the correct class greater than 0.7. Thus, we can be assured, based on the memberships assigned, that the samples are correctly classified.

VI. CONCLUSION

A fuzzy *K*-NN decision rule and a fuzzy prototype decision rule have been developed along with three methods for assigning membership values to the sample sets.

The fuzzy *K*-nearest neighbor and fuzzy nearest prototype algorithms developed and investigated in this report show useful results. In particular, concerning the fuzzy *K*-nearest neighbor algorithm with fuzzy *K*-nearest neighbor initialization, the membership assignments produced for classified samples tend to possess desirable qualities. That is, an incorrectly classified sample will not have a membership in any class close to one while a correctly classified sample does possess a membership in the correct class close to one. The fuzzy nearest prototype classifier, while not producing error rates as low as the fuzzy nearest neighbor classifier, is computationally attractive and also produces membership assignments that are desirable.

REFERENCES

[1] E. Hunt, *Artificial Intelligence*. New York: Academic, 1975.
[2] P. Winston, *Artificial Intelligence*. Reading, MA: Addison-Wesley, 1977.
[3] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.
[4] B. G. Batchelor, *Pattern Recognition*. New York: Plenum, 1978.
[5] R. Gnanadesikan, *Methods for Statistical Data Analysis of Multivariate Observations*. New York: Wiley, 1977.
[6] A. Reynolds and P. Flagg, *Cognitive Pyschology*. Cambridge, MA: Winthrop, 1977.
[7] P. Dodwell, *Visual Pattern Recognition*. New York: Holt, Rinehart, and Winston, 1970.

[8] K. T. Spoehr, and S. W. Lehmkuhle, *Visual Information Processing*. San Francisco, CA: Freeman, 1982.
[9] *Computer Diagnosis and Diagnostic Methods*, J. A. Jacquez, Ed. Springfield, Ill.: Charles Thomas, 1972.
[10] A. Wardle and L. Wardle, "Computer aided diagnosis—a review of research," *Meth. Inform. Med.*, vol. 17, no. 1, pp. 15–28, 1978.
[11] T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inform. Theory*, vol. IT-13, pp. 21–27, Jan. 1967.
[12] K. Fukunaga and L. D. Hostetler, "*K*-nearest neighbor Bayes risk estimation," *IEEE Trans. Inform. Theory*, vol. IT-21, no. 3, pp. 285–293, 1975.
[13] P. Hart, "The condensed nearest neighbor rule," *IEEE Trans. Inform. Theory*, vol. IT-14, pp. 515–516, 1968.
[14] T. M. Cover, "Estimates by the nearest neighbor rule," *IEEE Trans. Inform. Theory*, vol. IT-14, no. 1, pp. 50–55, 1968.
[15] M. E. Hellman, "The nearest neighbor classification rule with a reject option," *IEEE Trans. Syst. Sci. Cybern.*, vol. SSC-6, pp. 179–185, 1970.
[16] I. Tomek, "A generalization of the *k*-NN rule," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-6, no. 2, pp. 121–126, 1976.
[17] B. V. Dasarathy, "Visiting nearest neighbor—A survey of nearest neighbor classification techniques," in *Proc. Int. Conf. Cybern. Soc.*, 1977, pp. 630–636.
[18] P. A. Devijver, "New error bounds with the nearest neighbor rule," *IEEE Trans. Inform. Theory*, vol. IT-25, pp. 749–753, 1979.
[19] P. Scheinok and J. Rinaldo, "Symptom diagnosis: optimal subsets for upper abdominal pain," *Comp. Bio Res.*, vol. 1, pp. 221–236, 1967.
[20] G. T. Toussaint and P. Sharpe, "An efficient method for estimating the probability of misclassification applied to a problem in medical diagnosis," *Comp. Biol. Med.*, vol. 4, pp. 269–278, 1975.
[21] A. Whitney and S. J. Dwyer, III, "Performance and implementations of *k*-nearest neighbor decision rule with incorrectly identified training samples," in *Proc. 4th Ann. Allerton Conf. on Circuits Band System Theory*, 1966.
[22] B. V. Dasarathy, "Nosing around the neighborhood: A new system structure and classification rule for recognition in partially exposed environments," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-2, no. 1, pp. 67–71, 1980.
[23] L. A. Zadeh, "Fuzzy Sets," *Inf. Control*, vol. 8, pp. 338–353, 1965.
[24] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, New York: Plenum Press, 1981.
[25] M. Gupta, R. Ragade, and P. Yager, Eds., *Advances in Fuzzy Set Theory and Applications*. Amsterdam: North-Holland, 1979.
[26] P. P. Wang and S. K. Chang, Eds., *Theory and Applications to Policy Analysis and Information Systems*. New York: Plenum, 1980.
[27] A. Kandel, *Fuzzy Techniques in Pattern Recognition*. New York, John Wiley, 1982.
[28] A. Kaufmann, *Introduction to the Theory of Fuzzy Subsets*, vol. I. New York: Academic, 1975.
[29] A. Jozwik, "A learning scheme for a fuzzy *K*-NN Rule," *Pattern Recognition Letters*, vol. 1, pp. 287–289, July 1983.
[30] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Ann. Eugenics*, vol. 7, pp. 179–188, 1936.
[31] J. Keller and D. Hunt, "Incorporating fuzzy membership functions into the perceptron algorithm," submitted to *IEEE Trans. Pattern Anal. Machine Intell.*

0018-9472/85/0700-0585$01.00 ©1985 IEEE