A Single Nearest Neighbour Fuzzy Approach for Pattern Recognition

Sameer Singh

Department of Computer Science
University of Exeter
Exeter EX4 4PT
Plymouth PL4 8AA
United Kingdom

tel: +44-1392-264053
fax: +44-1392-264067
e-mail: s.singh@exeter.ac.uk

Singh, S. "A Single Nearest Neighbour Fuzzy Approach for Pattern Recognition", *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 13, no. 1, pp. 49-54, 1999.

ABSTRACT

The main aim of this paper is to introduce the single nearest neighbour approach for pattern recognition and the concept of incremental learning of a fuzzy classifier where decision making is based on data available up to time *t* rather than what may be available at the start of the trial, i.e. at t=0. The single nearest neighbour method is explained in the context of solving the classic two spiral benchmark. The proposed approach is further tested on the electronic nose coffee data to judge its performance on a real problem. This paper illustrates: 1) a novel fuzzy classifier system based on the single nearest neighbour method; 2) its application to the spiral benchmark taking the incremental pattern recognition approach; and 3) results obtained when solving the two spiral problem with both non-incremental and incremental methods and coffee classification with the non-incremental method. The results show that incremental learning leads to improved recognition performance for spiral data and it is possible to study the behavioural characteristics of the classifier with possibility related parameters.

keywords: pattern recognition, spiral problem, fuzzy classifier, nearest neighbour

1. Introduction

The analysis of non-linear data which is noisy, imprecise and possesses temporal characteristics has been an area of considerable interest in pattern recognition over the last few years. Fuzzy techniques have proved successful in several such applications since they take a more flexible approach to pattern recognition (Pal and Majumder, 1986). In this paper a single nearest neighbour fuzzy approach to pattern recognition is described. The approach is novel in its description however some of its elements represent the synthesis of previous research in the area. This paper describes this new approach in the context of other nearest neighbour and fuzzy pattern recognition techniques, and demonstrates the usefulness of the proposed method on solving the two spiral benchmark and recognising different blends of coffee using electronic nose data. The paper also discusses how non-iterative fuzzy classifiers may operate with incremental learning which should enhance the quality of their decisions.

Pattern recognition systems that work reliably with noisy and uncertain data are important for several applications. There exists a considerable amount of active research on the analysis of non-linear temporal data (Boulard, 1989). This data may be produced by two different types of sources: sources that are static in nature but produce data which is time-dependent, and sources which are dynamic and evolve with time, i.e. the higher order data function is variable: if $y = f(x)$, then for static sources f is fixed whereas in dynamic sources it changes with time. In practice most research addresses the analysis of first data type because recognition and validation methods for the second type are scarce (Bishop, 1995). Connectionist approaches are often appropriate when dealing with highly non-linear small amounts of data and previous research has addressed uncertainty management during information processing (Levine, 1995; Yeap et al., 1990); a summary is available in Boulard (1989). Fuzzy techniques have been used in the past for managing issues relating to non-linearity and noise in temporal data in conjunction with off-line systems including expert systems, for example SURTEL: a tool for programming temporal knowledge (Chen and Terrier, 1992), temporal uncertainty quantification using possibility distributions (Chen and Terrier, 1991), and fuzzy temporal reasoning for process supervision (Chen, 1995). Possibility is usually considered separate from probability and is defined as the ease with which an event can happen than its likelihood. For improving decision making in real environments, there is considerable interest in neuro-fuzzy systems, e.g. intelligent fuzzy control. Fuzzy neural networks have been shown to work better with noisy data compared to conventional techniques, for example fuzzy neural networks performance on electronic nose data is superior to non-fuzzy networks (Singh et al., 1996; Singh et al., 1997). In other practical applications, fuzzy pattern recognition methods often manage uncertainty in decision making better than conventional non-fuzzy methods (Klir and Folger, 1989; Pal and Majumder, 1986).

In the past, fuzzy pattern recognition has been applied in most cases with off-line systems such as expert systems and neural networks. For a few on-line operations, we are limited to applying such pattern recognition tools in applications where data is either linear, filtered or easily predictable. This limitation exists because on-line learning for non-linear data using a connectionist approach as with neural networks is very slow and computationally very expensive. In practice, on-line pattern recognition systems are attractive in several domains. Real-time industrial applications often need classifiers that train in real-time and that are: 1) reliable and consistent in their decision making in the presence of noise; 2) maximise the use of the available knowledge through intelligent data processing; and are 3) tolerant to some level of data inconsistency. Fortunately, fuzzy approaches offer the infrastructure to satisfy these requirements for building such classifiers.

1.1 *Spiral problem*

In this paper a difficult pattern analysis benchmark is analysed with the proposed single nearest neighbour method. The spiral benchmark is used to test the proposed technique before further experimentation with the real electronic nose data. The spiral problem is a classic example of non-linear data. The spiral data is considered here in two dimension since previous work exists in this area for comparison. The spiral program, available from the Carnegie Mellon AI repository, generates two sets of points, each set with 96 * density + 1 data points (3 revolutions of 32 times the density plus one end point). If a total of N data points are to be generated, then the spiral shape parameters change as follows, $1 \le i \le N$:

angle = (*i* * PI) / (16.0 * density) ...(1)

radius = maxRadius * ((104 * density) - *i*) / (104 * density) ...(2)

x = radius * cos(angle) ...(3)

y = radius * sin(angle) ...(4)

Here x and y are the spiral data points generated by the program. Since data points are generated in sequence, equations 1-4 are time dependent. The temporal nature of the resultant spiral is shown in Figure 1. Here the angle and radius of the spiral changes as new data is generated in sequence. The abscissa represents the *i*th pattern generated with each time step where each pattern comprises an (x, y) measurement. It should be noted that we do not analyse the temporal aspect of the spiral in any way in this paper.

Figure 1 here

The two spirals are governed by three parameters: density φ, radius σ, and offset δ[1]. The density variable defines the total number of points generated within an envelope defined by the radius. Data belonging to two different classes lie on these two different spirals (represented as a sequence of white and black circles in Figure 1). By manipulating spiral parameters, it is possible to generate different spirals with varying radius and length. A pattern classifier working on the problem should be able to recognise the training set before making test classifications. The benchmark has a training set, a validation set and a test set. The training set is represented by the vector $\{\mathbf{x}, \mathbf{y}\}$. The test and validation sets may be thought of as noisy

[1] The original spiral was proposed with $\varphi = 1$, $\sigma = 6.5$ and $\delta = .1$ (ref: Carnegie Mellon AI Repository)

spirals, i.e. training set plus a uniform level of noise δ. The validation set may be represented as {**x**, **y**+δ} and test set as {**x**+δ, **y**} where δ is a pre-defined scalar. This method of offsetting test and validation sets on different variables has been used for historical reasons than personal preference. It is possible to generate several different training and validation sets of noisy spirals with varying δ. Uncertainty in pattern recognition occurs when: the offset δ is large for a small radius σ spiral or when data is dense, i.e. φ is large.

Recognising the two spiral benchmark is a difficult task for several pattern recognition approaches since spiral data is highly non-linear. The problem has been difficult to solve using neural approaches (Touretzky and Pomerleau, 1989). It has been observed that backpropagation and its relatives encounter significant problems when training the neural network. In particular, deriving the optimal architecture is difficult, and furthermore, the training times are large. In addition, the spiral is under-constrained, i.e. data not lying on the spiral is often misclassified. For this reason, the two spiral problem has been particularly popular for testing novel neural and statistical pattern recognition classifiers. Considerable work has been done in the area since mid 1980s and in 1990s : neural networks (Fahlman, 1988; Fahlman and Lebiere, 1990; Lang and Witbrock, 1988; Tay and Evans, 1994), neurofuzzy methods (Sun and Jang, 1993) and data encoding methods (Chua et al., 1995; Jia and Chua, 1995) have been applied to solving the spiral problem. In addition, several other studies have tested their proposed pattern recognition methods on this benchmark problem since this process served as an indicator of their success with real-world problems, e.g. hypercube separation algorithm's initial success with this benchmark confirmed superior results with hand-written character recognition data (Ulgen et al., 1996). It is well recognised that the task to recognise two spirals that coil around each other and the origin is an important problem because: 1) spiral data is found in real applications and 2) the complexity of the problem can be varied by changing spiral parameters.

The fuzzy classifier proposed here will be first operated in the *static* mode. The term ‘static mode’ may be described as the conventional process of learning with most classifiers. Here, the application involves training and test data. The training data is used by the classifier system to learn training patterns and this information is then used for decision making in novel test situations. The knowledge may be embedded in the classifier during learning in the form of weights as with neural networks or statistical indices for other types of system. Once training is complete, test data is classified by the system with a certain degree of confidence. In most cases, the system learns by example which come from training data only. The training data remains fixed for all patterns tested during the test cycle. In the second approach, *incremental* mode, the system continues to learn new information after every pattern is tested if there is a reasonable confidence that the correct classification was made. This implies that incremental systems re-learn or adapt as they work on test data. This is an iterative process. The system follows a train→test→modify-training-set cycle as opposed to a train→test cycle with the *static* mode. With every test pattern classified, the system decides whether it is beneficial to now include it in its training set and re-train before making a decision on the next test pattern. This obviously leads to a training set which enlarges every time a correct classification is made during testing. Unfortunately, several iterative learning systems such as neural networks find it difficult to justify adopting this strategy no matter how attractive it may seem because of high computational costs involved in re-training. The non-iterative nature of training in most fuzzy classifiers is however well suited to the incremental approach.

In order to succeed, the incremental method must resolve the stability-plasticity dilemma (Grossberg, 1987). Grossberg states the dilemma as: “How can a learning system be designed to remain plastic in response to significant new events, yet also remain stable to irrelevant events?” (p. 30). This phenomenon addresses one of the key capabilities of the human cognitive system. In fact, it also happens to be the goal of fuzzy decision making to mimic the human thinking process and hence the dilemma must be resolved. In the *incremental* mode, the classifier system should be able to include (adapt) test data for retraining which enhances its decisiveness for future tests, and at the same time remain stable to those parts of training data which are of little relevance for a particular decision. This issue will be addressed later in light of experimental results on spiral data.

2. Single nearest neighbour approach

A single nearest neighbour approach will be proposed in this paper and applied to the spiral and coffee data. The proposed fuzzy classifier system works by detecting the nearest neighbour of a test pattern in the training set. The approach is to detect a single nearest neighbour rather than K-nearest neighbours (Bishop, 1995). The K-nearest neighbour approach yields poor results, less than 55% correct classification for spiral data. This may be shown in Figure 2 where according to Bishop (1995): "It involves finding a hypersphere around a point **x** which contains K points (independent of their class), and then assigning **x** to the class having the largest number of representatives inside the hypersphere" (p. 57). With this approach, the two spiral classes have an equal number of representatives in the hypersphere and classification is difficult (in Figure 2, which is a simpler representation of the two spiral benchmark, black and white circles represent training patterns of two different classes inside the hypersphere and the black square is the test pattern). The results obtained with the K-nearest neighbour method are also sensitive to the radius of the hypersphere chosen and the level of noise present. It is not very well suited to problems where data is non-linear to the extent that data of different types is strongly clustered together in an equidistant form as with the spiral benchmark.

Figure 2 here

A better approach to the above problem can be taken by detecting only a single nearest neighbour of the test pattern in the training set. The quality of spiral classification in this case depends on the sensitivity of the method for detecting the truly nearest neighbour. The single nearest neighbour approach is described below.

The data is initially divided as of training and test type. The purpose of the classifier is to allocate a class to the test data on the basis of training data on which it has been trained. The performance of the classifier is measured on the basis of correct test predictions that it makes. For a given test pattern X, the fuzzy classifier computes the membership of X in different classes C_1, ... C_j ... C_m where $1 \bullet j \bullet m$. The membership of X in class C_j can be expressed as $\mu_j(X)$. The test pattern is allocated to a class for which the membership function yields the maximum value. The overall process may be mathematically explained as: Consider an unknown pattern X represented by a point in a multi-dimensional space $\bullet_X$ consisting of *m* pattern classes C_1 ... C_m. Let R_1 ... R_j ...R_m be the reference vectors where R_j associated with C_j contains h_j number of prototypes such that, $R_j^{(l)} \in R_j$, $l = 1, 2, \ldots h_j$.

$$\mu_j^l(X) = [1 + \{d(X, R_j^{(l)})/F_d\}^{F}e\,]^{-1.0} \qquad \text{... (5)}$$

where $\mu_j^l(X)$ is the membership of X in class C_j as determined through the class sample *l*, and $d(X, R_j^l)$ is the distance between X and R_j^l. In eqn. (1), F_e and F_d are positive constants that determine the degree of fuzziness in the membership space. The main purpose of using the membership function is to map an *n*-dimensional feature space into a m-dimensional membership space which is a unit hypercube and satisfies the following conditions:

$$\mu_j^l(X) \rightarrow 1 \text{ as } d(X, R_j^l) \rightarrow 0 \qquad \text{... (6)}$$

$$\mu_j^l(X) \rightarrow 0 \text{ as } d(X, R_j^l) \rightarrow \text{infinity} \qquad \text{... (7)}$$

$$\mu_j^l(X) \rightarrow \text{increases, as } d(X, R_j^l) \rightarrow \text{decreases} \qquad \text{... (8)}$$

The test pattern X is allocated to class *i* if,

$$\mu_i(X) \;\geq\; \mu_j(X) \text{ for } i \neq j, \text{ where } i, j = 1\ldots m, \text{ and } \mu_j(X) = \max \sum_{l=1}^{h_j} (\mu_j^l(X)) \qquad \text{... (9)}$$

In this manner, the class of X is the class for which it has the highest membership value.

This approach is based on finding the nearest neighbour of the test pattern in the training data. The fuzzy approach to classification is generic to any test domain, but possibility calculation methods may need modification in different domains, e.g. manipulating the fuzzy parameters F_d and F_e in eqn. 1. One of the important advantages of the fuzzy pattern recognition method is that possibilistic algorithms can perform decision making with equal or weighted role of input features and with rejection thresholds where decisions based on low possibility counts are discarded, Bishop (1995, p. 28). In order to solve the spiral problem, the procedure for membership computation is explained below:

❶ Label the data as for training set Ω, validation set V and test set T. The benchmark consists of N =194 patterns in each of these sets.

❷ Separate class C_1 data and class C_2 training data in two training files F1 and F2.

❸ For every pattern $p_V = (x_v, y_v)$ in the validation set V, perform the following steps:

❹ Find the upper and lower bounds of x_v for class C_1 from F1. These may be represented as $x_v(lb_1)$ and $x_v(ub_1)$. Here lb_1 and ub_1 are positions at which lower and upper bound are found in the training data array Ω for class C_1. If $x_v > x_\Omega$ for all $x_\Omega \in$ F1, then x_v has only a lower bound. Similarly, if $x_v < x_\Omega$ for all $x_\Omega \in$ F1, then x_v has only an upper bound.

If C_1 has a total of h samples, the upper bound of x_v in C_1 is x_i such that

$x_i - x_v < x_k - x_v$ for all $x_i \geq x_v$ and $x_k \geq x_v$, $x_i \in C_1$, $x_k \in C_1$, $1 \leq i \leq h$, $1 \leq k \leq h$, $i \neq k$. Similarly, a lower bound can be found.

❺ For F1, calculate class membership for the following cases:

Case1

p_v already exists in F1 as a class C_1 pattern : $\mu_1(x_v) = 1.0$; : $\mu_1(y_v) = 1.0$;

Case2

x_v exists in F1 at position i, $1 \le i \le N$ but not y_v in the same position.

$\mu_1(x_v) = 1.0$; $\mu_1(y_v) = 1.0/(1.0 + \eta_1)$ where $\eta_1 = |y_v - y_j|$

$\text{Lt}_{\eta_1 \to 0}\ \mu_1(y_v) = 1.0$;

Case3

y_v exists in F1 at position j but not x_v in the same combination

$\mu_1(y_v) = 1.0$; $\mu_1(x_v) = 1.0/(1.0 + \eta_2)$ where $\eta_2 = |x_v - x_j|$

$\text{Lt}_{\eta_2 \to 0}\ \mu_1(x_v) = 1.0$;

Case4

x_v and y_v do not occur in F1

$\mu_1(x_v) = 1.0/(1.0 + \eta_3)$ if $\eta_3 \le \eta_4$ else

$\mu_1(x_v) = 1.0/(1.0 + \eta_4)$ if $\eta_4 < \eta_3$

where $\eta_3 = |x_v - x_v(lb_1)|$ and $\eta_4 = |x_v(ub_1) - x_v)|$

$\mu_1(y_v) = 1.0/(1.0 + \eta_5)$ if $\eta_5 \le \eta_6$ else

$\mu_1(y_v) = 1.0/(1.0 + \eta_6)$ if $\eta_6 < \eta_5$

where $\eta_5 = |y_v - y_v(lb_1)|$ and $\eta_6 = |y_v(ub_1) - y_v)|$

❻ Perform steps 4 & 5 on F2 to calculate $\mu_2(x_v)$ and $\mu_2(y_v)$.

❼ Derive an optimal function ξ for the following:

If $\xi(\mu_1(x_v), \mu_1(y_v)) > \xi(\mu_2(x_v), \mu_2(y_v))$ then $p_v \in C_1$, else $p_v \in C_2$.

The results shown later have used the multiplication function for possibility combination.

❽ Determine the recognition rate through correctly classified patterns.

❾ Test the approach on the test set T by following steps 2 to 8.

In the above discussion, the membership μ in case 2 is inversely proportional to the distance between the target and the actual *y* value in the same position. In case 4 when both *x* and *y* test values are not present in the training set, the membership is inversely proportional to the distance between the test value and either its upper or lower bounds depending on the nearest one in the training file. It should be noted that in the described method, $F_e = 1$ and $F_d = 1$ from our experiment. In the above algorithm, if we find x_v or y_v in F1 or F2 in more than one position, the membership function at these different positions is calculated and the highest value is chosen. In rare cases it is also possible that x_v or y_v may have two or more upper and lower bounds at different positions in F1 or F2. As previously, the membership function is computed for these different cases and the highest value is chosen to be $\mu_1(x_v)$ or $\mu_1(y_v)$ as may be the case.

The above possibility generation algorithm is non-parametric in nature. The conventional method for gaussian populations as proposed by Zadeh (1965; 1987) and Mamdani and Gaines (1981) has not been followed for this problem for two main reasons: the spiral data is non-parametric and poor experimental results were obtained using their method. The often used original method proposed by Zadeh for fuzzy possibility determination is based on the assumption that training data is normal in nature. Their method determines the mean and the range statistics of the training data and divides the possibility density spectrum in four distinct regions, i.e. $\alpha_{min} \rightarrow \beta_1$, $\beta_1 \rightarrow \gamma$, $\gamma \rightarrow \beta_2$, $\beta_2 \rightarrow \alpha_{max}$ (Zadeh, 1965; Zadeh, 1987). In the training data, α_{min} is the minimum measurement for a given input feature, β_1 is the measurement which is less than the mean γ and has a possibility of occurrence equal to 0.5, β_2 is a measurement greater than the mean γ and has a possibility of occurrence equal to 0.5, and α_{max} is the maximum measurement of the input feature (in a possibility distribution curve). All these measurements are made on the same variable in the training data. Any measurements below the minimum and greater than the maximum have a zero possibility of occurrence and the mean γ has the highest possibility of occurrence, i.e. 1.0. For a test pattern $p_\Gamma = (t_1, t_2, ..., t_n)$ with each of its *i*th feature measurement t_i, Zadeh's (1965) method finds the possibility that this pattern belongs to a particular class in the training data based on the considered feature using a quadratic function.

The minimum, maximum and mean values for a particular input variable (feature) for a given class in training data is first calculated. The possibility that the test value for a given feature belongs to a particular class is based on its proximity to the mean training data value of that feature. This possibility decreases quadratically as one moves away from mean, i.e. lower or higher values. This possibility determination formula can be used as a part of the nearest neighbour approach for normal data so that only a single nearest neighbour is selected with the possibility calculation method. In some applications highly successful results have been achieved using this approach (Singh and Steinl, 1996). Unfortunately this technique does not work very well with non-parametric data. In such application data, measurements encountered in practice are no longer clustered around a mean value but the spread is much wider, irregular and skewed. The proposed possibility calculation method is well suited in such circumstances.

Figure 3 shows the gray scale image of a possibility grid generated by the proposed single nearest neighbour algorithm. The grid plots the possibility for four possible combinations of X $\pm$ d and Y $\pm$ d where d is the offset δ.

Figure 3 here

In Figure 3, the data point where the four edges meet is the training data point with the possibility of occurrence equal to 1.0. The possibility decreases for test values as the value of the offset increases in all four quadrants (remembering that test data is training data point added to a positive or negative offset). The possibility decrease is well-defined and linear when only one of the variables X or Y is offset as shown by the four perpendicular edges where possibility declines gracefully. This decline will continue for a larger grid till the possibility becomes near zero. However, when both variables are offset then the possibility of occurrence of test data drops rapidly in all four quadrants. At larger offsets, the change in possibility is very little as shown by flat surfaces as we move towards the end of each quadrants.

3. Recognition results

The results are presented for two different data sets. The proposed single nearest neighbour approach is applied to spiral and electronic nose data classification.

3.1 *Spiral data*

It is important to test the robustness of the proposed technique exhaustively on more than one 2-D test spirals. The following discussion relates to the results obtained on recognising test spirals of varying radii and offsets. The term "offset δ" refers to a uniform level of noise present in noisy test spirals (remember that spiral test data is the training set mixed with noise). We generate two different training sets: one for a spiral with radius 6.5 which is used in conjunction with eleven noisy test sets that represent the original training data for **x** offset for each pattern by varying a uniform noise factor $.1 \leq \delta \leq 2.9$ (results in Table 1), and a more dense spiral with radius 3.5 where we have a varying uniform noise factor $.15 \leq \delta \leq 1.2$ (results in Table 2). The noise factor is uniform in the sense that the same noise value is added to all test patterns.

In the spiral recognition task, there are two things to study: the recognition rate of the classifier on the given task, and the behaviour of the classifier. The recognition rate on the validation and the test set may be denoted as $\Re_v$ and $\Re_\Gamma$. These represent, in percentage, the ratio of the correctly classified test patterns to the total number of patterns tested. A given pattern is correctly classified if the class to which it has been allocated by the fuzzy algorithm is its true class. This is possible since we know in advance the true class of tested data. A robust classifier should be capable of high recognition rates and graceful degradation in performance with increasing offsets.

The behaviour may be explained for the complete test run with an entropy function. For our purposes, entropy ε represents the stability of the system: $\varepsilon = \sum_{j=1}^{N} p_j$ where p_j represents the sum of the possibilities that pattern j belongs to class C_1 and C_2. This description should be taken in the context of our results and is by no means a standard definition. In some applications this may actually represent *reverse* entropy. Results in the *static* recognition mode of the single nearest neighbour method on spiral data are discussed first. These are shown in Table 1.

Table 1 here

The table shows the change in entropy and recognition rates for the validation and test sets with varying offsets on a spiral of radius $\sigma = 6.5$. The results show that the recognition rates vary between 98.9 and 80.9% on the test set, and between 100 and 83% for the validation set as the offset δ increases. This is a highly encouraging result for the single nearest neighbour classifier. The degradation in performance with offset increase is graceful as expected since uncertainty in decision making increases with increasing noise factor δ. This increase in uncertainty by increasing the noise offset is observed by decreasing entropy values representing low possibility (confidence) in decision making.

In order to test and improve the decision making of the classifier, *incremental* mode of pattern recognition is now implemented. This time we scale the problem to a different dimension; we reduce the radius and offsets by nearly half. This implies that the complexity of the problem remains unchanged for comparison with Table 1 in terms of spiral size. However, decision making this time has become more difficult since spiral density is the same as before, i.e. this time the same number of data points as before in a smaller envelope. The results are shown in

Table 2 which compares the *incremental* mode results with those obtained using the *static* mode.

Table 2 here

In the incremental mode, the train→test→modify-training-set cycle is followed. With every test pattern tested at time *t*, if the test output matches the target with reasonable confidence, then this pattern is added to the training set. The confidence level can be set using possibility thresholds, e.g. the winner class must have a possibility of μ_{min}. This may be set by the user as for example in our experiment $\mu_{min} = .5$. Thus, at time $t+1$, decisions on the next test pattern are based on the current state of the training set. In this procedure, the size of the training set increases in a step-wise manner as more test patterns are added. In Table 2 the results are based on a more dense, smaller spiral ($\sigma = 3.5$) and only the test set results are reported. Base_1 and base_2 are the total number of training patterns of class C_1 and C_2 at the beginning of each trial (different train-test trials for different test sets; each test set consists of test patterns with a different level of mixed noise). These training bases are progressively increased with c_1 and c_2, the total number of class C_1 and C_2 test patterns correctly predicted with winning possibilities greater than 0.5, *incrementally*-after every test pattern if correctly predicted. In Table 2, $\Re_1$ represents the *static* recognition rate achieved with the conventional train→ test cycle and $\Re_2$ represents the *incremental* recognition rate achieved with the train→test→modify-training-set cycle.

Table 2 shows that incremental pattern recognition rates are comparatively very high, stable and resistant to increasing offset. The logical explanation is that the fuzzy classifier makes good decisions on spirals with increasing noise it has learnt to discriminate between spirals with smaller amounts of noise. The system adapts to novel situations on the basis of all of its current information. The system on the other hand continues to degrade in performance in the *static* mode since it operates with an inflexible training set.

The table shows that with every trial, the number of correct classifications with the incremental approach remains stable at nearly 98%. This is because the system modifies its training base with correctly predicted test patterns. The presence of more relevant information strengthens the decision making and the classifier continues to perform at a desired level. The *static* approach also yields high quality results with the single nearest neighbour algorithm. Recognition rates are obtained between 98.9 and 81.4% confirming the high quality classification ability. At the lower end of the table, recognition rates degrade gracefully with increasing offsets. In practical applications, these rates may be improved by only considering test results that are made with a reasonable amount of confidence, i.e. where winner possibilities exceed rejection thresholds.

An important feature of the incremental pattern recognition approach is that it adapts the classifier to novel information but at the same time produces results which show that the system is stable to irrelevant details. The irrelevant details may be rejected using thresholds or through diminishing their role in decision making. The single nearest neighbour approach as described before is aimed at detection than modification. In other words, the aim is to detect a nearest neighbour than modify a learning function. In this manner, undesired training information plays no part in decision making which would have otherwise affected decision making in environments where all of the training data modifies the learning function. The classifier system so produced is plastic to novel situations and yet stable to irrelevant details, thus managing the plasticity-stability dilemma.

The incremental approach should be used with care. In several industrial applications where the output target for test data may not be known in advance, including weakly classified test data in training data for future decision making can seriously jeopardise the capability of the classifier system or even destabilise the system. In such circumstances, other indicators of belief in decisions may be considered to decide whether to include tested patterns for modifying the training set, if they are strong classifications, or for throwing them out if they are weak classifications. Incremental pattern recognition is especially useful where the initial training set is small in size and where we know that the more information we have for decision making, the better we are. However, the method is only applicable in practice with non-iterative statistical approaches to make it viable, i.e. it becomes computationally viable.

3.2 *Coffee classification*

The spiral data set is an artificial benchmark with the advantage that it allows extensive experimentation through the manipulation of different parameters. It is equally important to investigate that the single neighbour algorithm proposed here has practical utility. In order to confirm the usefulness of the proposed fuzzy algorithm for real applications, the technique was tested on electronic nose data (Singh et al., 1996). An electronic nose is an equipment which is used to collect data on various odours and substances. This data is then used for their recognition and classification. The nose comprises of a set of odour sensors that exhibit differential response to a range of odours and vapours. The data is collected by a set of twelve semi-conducting tin-oxide sensors. These sensors exhibit differential change in their steady-state conductance for different types of coffees. The measurements are made under constant ambient conditions (e.g. at 30° C and 50% RH). The task here is to classify three different blends of coffee. Each coffee class is a different blend. Each blend data consists of 30 patterns except the last one with 29 patterns (total 89 patterns) . The classification task is to learn to discriminate between the three blends. In previous work, both non-fuzzy and fuzzy neural networks have been tested on this nose data and results for comparison are available.

The coffee data is small in size and highly non-linear. Since there are no separate test sets, cross-validation is used. Cross-validation is the process of estimating the true performance of a classifier on a given data by segmenting it into disjoint sets of training and test data (see Bishop, 1995, p. 374 for a detailed description). A ten fold cross-validation is chosen. In this process, a total of ten train-test trials are conducted. Each time 90% of the data is used for training and the remaining 10% is used for testing. In each trial or 'fold', the training data is always distinct from the test data. The recognition rates are then averaged over these folds for a realistic estimate of the classifier performance on all test data.

Table 3 shows the comparative performance of the single nearest neighbour fuzzy algorithm against the standard backpropagation neural network implemented as a 12*x*3*x*3 architecture using a ten fold cross-validation. This architecture specifies the use of a multi-layer perceptron with twelve input nodes for twelve input features, three hidden nodes and three output nodes for the three blends of the coffee. The single nearest neighbour approach has been described for only two input variables in section 2. This was extended following the same principles for a total of twelve input variables as in the coffee problem. The approach has the advantage that it may be used with standard validation methods and therefore cross-validation comparison is possible in Table 3.

Table 3 here

In the above table, the performance of the backpropagation neural network has been taken as the best out of 50 trials with different starting weights in every fold. Since this data has been analysed with neural networks before, the above performance represents the best achieved in this and previous studies after experimentation with different network architectures (see Singh et al. (1996) for further description).The single nearest neighbour fuzzy algorithm performs superior (recognition rate of 84.4%) in comparison with the standard backpropagation algorithm on this task by more than 3% difference in their recognition rates. It is a reasonable conclusion that the algorithm can be useful for other practical applications too.

4. Conclusion

In the above description , a fuzzy method for pattern classification has been proposed and its operation in static and incremental learning modes has been discussed taking the two spiral and coffee data as its experimental focus. The behaviour of the classifier was studied using system entropy. It was noted that the process of incremental pattern classification yields superior recognition performance compared to the static approach. The single nearest neighbour approach makes the assumption that data is continuous and unencoded. In theory, it may be used for any classification problem which may be otherwise solved with other nearest neighbour methods. It is particularly well suited in environments where data sets are small, non-linear and fuzzy. This is proved by our latter experimentation with electronic nose data. The algorithm out-performs a three layered neural network trained with the backpropagation method. The incremental approach yields attractive results and should now be tested in other industrial applications. Hopefully, the single nearest neighbour approach will be adopted and tested in industrial domains where training data is small, non-linear, temporal and requires real-time recognition.

References

C. M. Bishop, *Neural networks for pattern recognition*, Clarendon Press, Oxford, 1995.

M. Brown and C. Harris, *Neurofuzzy adaptive modelling and control*, Prentice Hall, 1994.

H. A. Boulard, “Continuous speech recognition: Hidden Markov models vs. the connectionist hope,” Proceedings of the 23rd Asilomar Conference on Signals, Systems and Computers. Maple Press. San Jose. vol. 1. pp. 331-335, 1989.

Z. Chen and F.Terrier, “ About temporal uncertainty,” Proceedings of the IEEE/ACM International Conference on Developing and Managing Expert System Programs. IEEE computer society press, CA. pp. 223-230, 1991.

Z. Chen and F. Terrier, “SURTEL: A tool for programming temporal and uncertain knowledge,” 12th International Conference on AI, Expert systems and Natural language, Nanterre, France. vol. 1. pp. 485-496, 1992.

Z. Chen, “Fuzzy temporal reasoning for process supervision. *Expert Systems*. vol. 12. issue 2 (1995) 123-137.

H. Chua, J. Jia, L. Chen and Y. Gong, “Solving the two-spiral problem through input data encoding,” *Electronics letters*. vol. 31**,** issue 10 (1995) 813-14.

S. E. Fahlman, “Faster-learning variations on back-propagation: An empirical study,” Proceedings of the 1988 Connectionist Models Summer School. Morgan Kaufmann, 1988.

S. E. Fahlman and C. Lebiere, “The cascade-correlation learning architecture,” in Advances in neural information processing systems 2, ed. D. S. Touretzky, Morgan Kaufmann, 1990.

S. Grossberg, “Competitive learning: From interactive activation to adaptive resonance,” *Cognitive Science* 11 (1987) 23-63.

J. Jia and H. Chua, "Solving two-spiral problem through input data representation," Proceedings of the IEEE International conference on neural networks, vol. 1, pp. 132-135, 1995.

G. J. Klir and T. Folger, *Fuzzy sets, uncertainty and Information*. Addison-Wesley, Massachusetts, 1989.

B. Kosko, *Neural networks and fuzzy systems - A dynamical systems approach to machine intelligence*, Prentice Hall International edition, 1992.

K. J. Lang and M. J. Witbrock, "Learning to tell two spirals apart," Proceedings of the 1988 Connectionist Models Summer School. Morgan Kaufmann, 1988.

E. Levine, "A time warping neural network," International conference on Acoustics, Speech, and Signal Processing, IEEE press. vol. 5. pp. 3339-3341, 1995.

E. H. Mamdani and B. R. Gaines (ed.), *Fuzzy reasoning and its applications*, Academic Press, 1981.

S. K. Pal and D. D. Majumder, *Fuzzy mathematical approach to pattern recognition*, John Wiley, New York, 1986.

S. Singh and M. Steinl, "Fuzzy search techniques in knowledge-based systems," Proceedings of the 6th International conference on data and knowledge systems for manufacturing and engineering (DKSME'96). Tempe. Arizona, pp. 1-10, 1996.

S. Singh, E. L. Hines and J. Gardner, "Fuzzy neural computing of coffee and tainted water data on electronic nose," *Sensors and Actuators B,* vol. 30, issue 3 (1996) 190-195.

S. Singh, E. L. Hines and J. W. Gardner, "Classifier Systems Based on Possibility Distributions: A Comparative Study," Proc. 3rd International Conference on Artificial Neural Networks and Genetic Algorithms ICANNGA97, Norwich, UK, Wein: Springer, 1997.

C. T. Sun and J. S. Jang, "A neuro-fuzzy classifier and its applications," Proceedings of the IEEE International conference on fuzzy systems. vol. 1, pp. 94-98, 1993.

L. P. Tay and D. J. Evans, "Fast learning artificial neural network (FLANN II) using the nearest neighbour recall," *Neural, parallel and scientific computations*, vol. 2. issue 1 (1994) 17-27.

D. S. Touretzky and D. A. Pomerleau, "What's hidden in the hidden layers?" *Byte* (1989 August issue) 227-233.

F. Ulgen, N. Akamatsu and T. Iwasa, "The hypercube separation algorithm: a fast and efficient algorithm for on-line handwritten character recognition," *Applied Intelligence*. vol. 6, issue 2 (1996) 101-116.

T. H. Yeap, S. G. Zaky, J. K. Tsotsos and H. C. Kwan, "A hierarchical neural network for temporal pattern recognition," IEEE International symposium on circuits and systems. IEEE press. vol. 4. pp. 2967-2970, 1990.

L. A. Zadeh *Fuzzy logic and its applications,* Academic Press, New York, 1965.

L. A. Zadeh, "Fuzzy sets as a basis for a theory of possibility," in *Fuzzy sets and application,* R. R. Yager, S. Orchinnikov, R. M. Tong and H. T. Nguyen (eds.). Wiley Interscience Publication, pp. 193-218, 1987.

Figure 1. 2D Spiral Data. Change in X and Y measurements of two spirals with time. Each successive pattern (x, y) is generated at the next step.

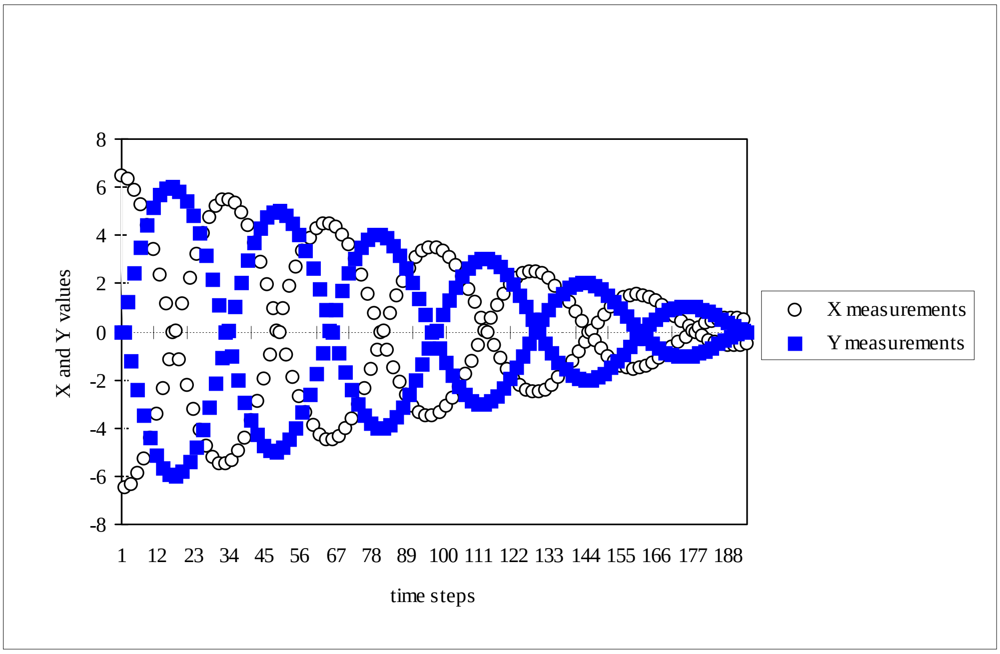

Figure 2. 2D Spiral data scatterplot. Two spirals with a maximum radius of 6.5 coil around each other. The two different classes are highlighted in a hypersphere with their training data (white and black points) and a test pattern is illustrated with a black square.

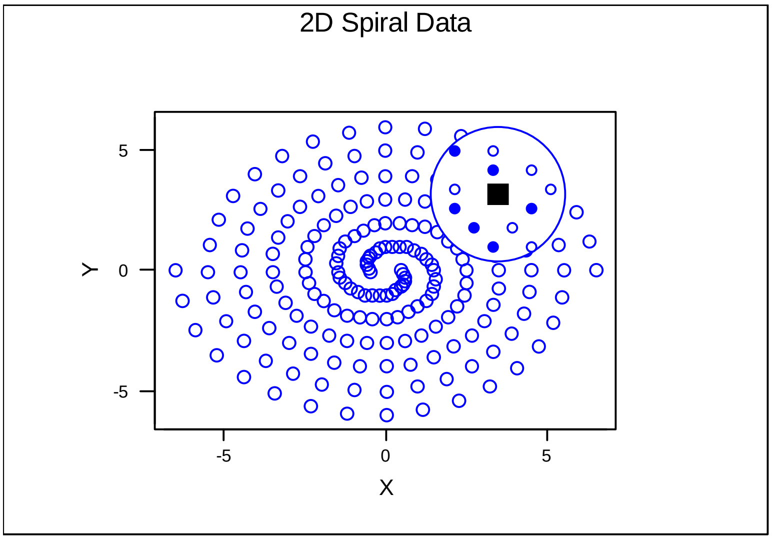

Figure 3. A possibility grid for spiral data. The point with the highest possibility 1 is a training data point. The grid shows possibility generation for all four quadrants, i.e. X ± d, Y ±d where d is the offset δ. The grid has been plotted taking X = 5.24, Y = 3.50 and d = .01 for a total of 2500 test data points.

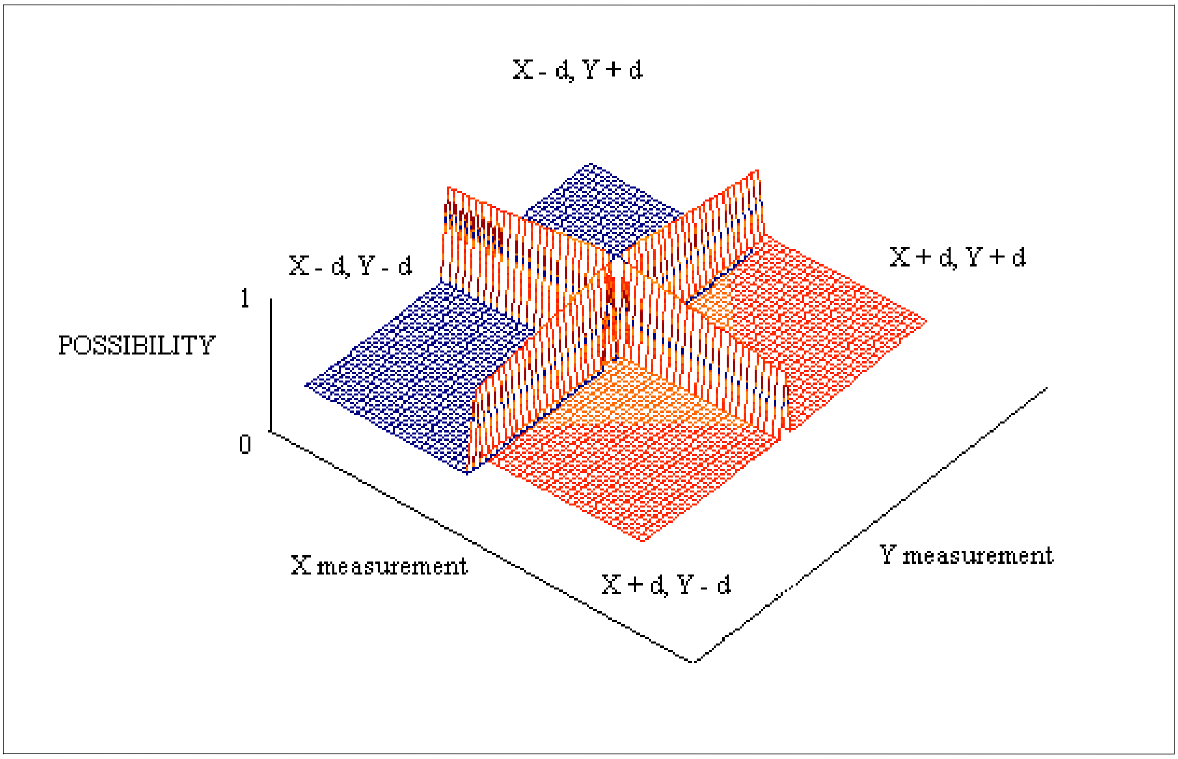

Table 1. The variation in the entropy ε, and recognition rate ℜ % as a function of the spiral offset δ with radius σ = 6.5. Entropy and recognition rate for the validation set are ε_v and $\Re_v$ and for the test set are ε_Γ and $\Re_\Gamma$.

Offset δ	ε_v	ε_Γ	$\Re_v$	$\Re_\Gamma$
.1	284.5	284.2	100	98.9
.5	262.3	261.3	100	99.5
.9	253.4	247.7	98.5	99.5
1.3	238.2	238.9	98.9	98.5
1.7	230.0	232.4	99.5	98.5
2.1	222.0	227.6	98.5	92.8
2.5	220.0	223.9	95.4	90.7
2.9	219.1	220.5	83.0	89.1

Table 2. *Static* r*ecognition rate* $\Re_1$ % *and incremental recognition rate* $\Re_2$ % for spirals with varying levels of noise offset δ (*radius* σ = 3.5; *offset* .15 ≤ δ≤ 1.2).

Offset δ	*base1*	*base2*	c_1	c_2	$\Re_1$	$\Re_2$
.15	97	97	95	95	98.9	97.9
.30	192	192	95	95	97.9	97.9
.45	287	287	95	95	97.9	97.9
.60.	382	382	95	95	97.9	97.9
.75	477	477	95	95	97.4	97.9
.90	572	572	95	95	97.9	97.4
1.05	667	667	94	95	93.2	97.4
1.20	761	762	95	95	90.2	97.9

Table 3. Misclassified patterns and the recognition rate for the two techniques on electronic nose data using 10 fold cross-validation

Cross-validation fold	*Neural Net with Backpropagation*	*Fuzzy Single Nearest Neighbour*
1	1	1
2	1	1
3	2	3
4	3	1
5	1	1
6	1	4
7	1	0
8	1	1
9	3	1
10	2	1
Misclassified	16	14
Recognition rate	***81.1%***	***84.4%***