Monograph:

Lim, Chee Peng and Harrison, R.F. (1996) On-Line Pattern Classification with Multiple Neural Network Systems: An Experimental Study. Research Report. ACSE Research Report 651 . Department of Automatic Control and Systems Engineering

X

On-line Pattern Classification with Multiple Neural Network Systems : An Experimental Study

Chee Peng Lim and Robert F. Harrison

Department of Automatic Control and Systems Engineering
The University of Sheffield
Mappin Street, Sheffield S1 3JD
United Kingdom

Research Report No. 651
November 1996

1 Introduction

In the field of pattern recognition, researchers have proposed the application of multiple classifiers to the same data set, and the combination of the results using some decision algorithm to improve the performance of individual classifiers [1] [2] [3] [4]. The use of a single system for pattern classification hinges on the assumption that the system is able to capture and process all the input features regardless of what the features might be. In cases where the above assumption fails to hold true, *e.g.* the input features might consist of a mixture of syntactic primitives, linguistic variables, continuous, discrete or nominal attributes, presenting all these features to one classifier for it to make a decision is difficult owing to the diverse types of features. Concatenating all the features into a high-dimensional input vector not only will increase computational burden, but will also cause accuracy and tractability problems for some classifiers owing to the so-called "curse-of-dimensionality" [5].

Another reason for using multiple classifiers is that there is a large number of different classification algorithms available in the pattern recognition literature, *e.g.* statistically-based classifiers, distance-based classifiers, syntactically-based classifiers, decision trees, and more recently, neural network classifiers, and fuzzy-neural classifiers. For a specific problem, different algorithmic classifiers may attain different degrees of accuracy. Hence, one can design a multiple classifier system using a statistically-derived classifier to handle statistically-invariant features, a syntactically-derived classifier to handle linguistic variables and primitives, and another classifier which is good at handling incomplete input features or missing data, and so on. The predictions from various classifiers can then be combined using some decision combination procedure. The algorithms for combining multiple classifiers should take advantage of the strengths of the individual classifiers, avoid their weaknesses, and improve overall classification accuracy. As a result, multiple classifiers provide an alternative for developing a more reliable recognition system.

Based on the output information, there are three levels for integrating decisions from multiple classifiers [3]: level 1 (abstract level), where each classifier only yields a unique level indicating the predicted class, *e.g.* in a three-class situation, C_2 is predicted as final output; level 2 (rank level), where each classifier ranks the target outputs in an descending order, *e.g.* C_2 is the most probable class, followed by C_1, and then C_3; level 3 (measurement level), where each classifier assigns every class label a measurement value to indicate the degree to which the input belongs to that class, *e.g.* 70% in C_2, 20% in C_1, and 10% in C_3. It is obvious that classifiers operating at the abstract level produce the least information, and any type of classifier will at least supply output at the abstract level. Classifiers in levels 2 and 3 can easily be transformed to those in level 1 by omitting information such as degrees of rank or measurement, and by selecting only the most probable output as the predicted class. Thus, decision combination at level 1 constitutes the most general framework for integrating outputs from any type of classifiers.

In this paper, we report experimental work using multiple neural network classifiers in two benchmark problems. In the next section, an introduction to the classifier, which has been employed throughout the experiments, and a justification of its use for on-line learning tasks are given. In section 3, two experiments are presented where the experimental results are analyzed and their implications are discussed. Some conclusions are included in section 4. Note that it is beyond the scope of this paper to cover either the neural network classifier or the decision combination algorithms in great detail. Summarized descriptions of these algorithms are given in the appendix and the detailed mathematical derivations can be found in the appropriate references.

2 The Neural Network Classifier

Feedforward neural networks, such as the Multi-Layer Perceptron (MLP) network and the Radial Basis Function (RBF) network, possess some attractive properties when the objective is to develop a classifier to operate in a probabilistic environment. They are known to be able to *represent* any (smooth enough) function to an arbitrary degree of accuracy [6] [7]. For a wide variety of objective functions, they can be shown to *estimate* the appropriate Bayesian posterior probabilities directly, under some mild conditions [8] [9] [10]. It seems likely then, that feedforward networks can offer a direct solution to the problem of developing a one-from-many classifier. Unfortunately, one of the problems associated with such an approach is that the operating environment has to be stationary, and that the data sample used in training has to be sufficiently representative. In cases where learning takes place in a non-stationary environment or on-line, it is either necessary to allow a feedforward network to carry on learning or to re-train it, off-line. The latter suggests that whenever the operating environment changes, re-training is necessary in order to adapt to new situations which is a time-consuming and laborious process. Without any special learning methodologies, the former is not to be recommended especially for the standard MLP configuration owing to the likelihood of catastrophic forgetting [11] [12] [13].

In our previous work [14] [15], we have developed a hybrid network which is capable of incremental learning, and thus avoids the problems of catastrophic forgetting and re-training when operating on-line, and/or in non-stationary environments. The network is based upon an integration of two neural network architectures: Fuzzy ARTMAP [16] and the Probabilistic Neural Network [17]. This hybrid system has been demonstrated to be capable of providing outputs which estimate Bayesian posterior probabilities, and of achieving the theoretical (Bayes optimal) classification rates, on-line, without prior knowledge of impending changes in the environments. The hybrid network achieves comparable performance with other approaches in a number of benchmark problems [15] [18], but with the ability of on-going (causal) learning.

2.1 Fuzzy ARTMAP

It is well documented that networks of the Adaptive Resonance Theory (ART) family offer an alternative for solving the so-called stability-plasticity dilemma—how a learning system can safely adapt to new information without corrupting or forgetting previously learned information [19]. The ART networks are able to continue to learn incrementally in a changing data environment, and thus appear to offer a viable approach to the development of autonomously learning classifiers for on-line applications. More recent developments have provided a supervised mapping neural network model known as Fuzzy ARTMAP [16] which realizes a synthesis of ART and fuzzy logic by exploiting a close formal similarity between the computations of fuzzy subsethood and ART category formation and learning.

Figure 1 depicts a schematic diagram of the Fuzzy ARTMAP (FAM) network. It consists of two identical fuzzy ART [20] modules, ART$_a$ and ART$_b$, linked by a map field, F^{ab}. The ART$_a$ (as well as ART$_b$) module has two layers of nodes: F_1^a (F_1^b) is the input layer; and F_2^a (F_2^b) is a dynamic layer where each node encodes a prototype pattern of a cluster of input samples. F_0^a (F_0^b) is a pre-processing layer in which an M-dimensional input vector, $\boldsymbol{a} \in [0,1]^M$, is complement-coded [16] [20] so that the size of the input vector is kept constant in order to avoid the category proliferation problem.

In ART$_a$, the number of nodes in F_2^a can be increased when necessary, and the speed of increment is controlled by a vigilance parameter, $\overline{\rho}_a$, which is a user-defined threshold value between 0 and 1. This is the key feature of FAM where a novelty detector is used to measure, against a threshold, the similarity between the prototype patterns stored in the network and the input patterns. When the criterion is not satisfied, a new node is created, and the input is coded as its prototype pattern. As a result, the number of nodes grows with time, subject to the novelty criterion, in an attempt to learn a good network configuration autonomously and on-line. As different tasks demand different network structures, this learning methodology avoids the need to specify a pre-defined static network size, or to re-train the network off-line.

The above scenario is equally applicable to ART$_b$. During supervised learning, ART$_a$ receives a stream of input pattern vectors, $\{A\}$, whereas ART$_b$ receives the corresponding target-class vectors, $\{B\}$. However, in one-from-N classification (*i.e.* each input pattern belongs to only one of the N possible output classes), ART$_b$ can be replaced by a single layer containing N nodes. Then, the N-bit teaching stimulus can be coded to have unit value corresponding to the target category and zero for all others.

In summary, FAM operates as an incremental clustering algorithm (similar to the sequential leader clustering algorithm [21]) to classify arbitrary sequences of input patterns into different recognition categories. As mentioned earlier, FAM does not directly associate input patterns at ART$_a$ to target patterns at ART$_b$. Rather, input patterns are first self-organized into prototypical category clusters before being associated with their target stimuli at ART$_b$ via a map field. At each input pattern presentation, this map field establishes a link from the winning category prototype in ART$_a$ to the target output in ART$_b$. This association is used, during testing, to recall a prediction when an input pattern is presented to ART$_a$.

2.2 The Probabilistic Neural Network

The Probabilistic Neural Network (PNN) is a neural network model that implements Bayes' theorem in its learning methodology. It learns instantaneously in one-pass through the data samples and is able to formulate complex decision boundaries which approximate asymptotically the Bayes optimal limits. In addition, the decision boundaries can be modified on-line when new data is available without having to re-train the network. Another advantage of the PNN is its speed of learning, which is often orders of magnitude faster than that of the MLP [17]. However, as the PNN encodes every input sample as a new node into the network, this increases the network complexity and computational cost if large or unbounded training sets are used.

The key feature of the PNN is its ability to estimate probability density functions (pdfs) by using the Parzen-windows technique [22] based on the data samples, *i.e.* a non-parametric density estimation procedure. Figure 2 depicts a schematic diagram of the PNN for binary classification tasks (class A or class B). The PNN consists of four layers of nodes: the input layer, pattern layer, summation layer, and output layer [17]. Nodes in the pattern layer are organized in groups corresponding to different target classes. The pattern nodes belonging to the same output are then linked to a summation node dedicated to that particular target class.

During operation, the input pattern, $\boldsymbol{x}$, is first fanned-out to the pattern layer where each pattern unit computes a distance measure between the input and the weight pattern represented by that node. The distance measure (*e.g.* dot-product) is then transformed by an activation function, which is a Parzen kernel function. Outputs from the Parzen kernels are summed by the summation nodes. These outputs correspond to estimates of the pdfs of the input pattern with respect to each target class, *e.g.* $P(\boldsymbol{x}|A)$, $P(\boldsymbol{x}|B)$. For classification problems, these estimates can be weighted by their prior probabilities, $P(A)$, $P(B)$, thus, allowing calculation of the posterior probabilities according to Bayes rule, $P(A|\boldsymbol{x}) = P(\boldsymbol{x}|A)P(A) / P(\boldsymbol{x})$. On the other hand, different risk (loss) factors can also be assigned to each correct or incorrect decision to implement the minimum, risk-weighted Bayes classification rule.

2.3 Probabilistic Fuzzy ARTMAP

Our studies have found that there is a close similarity in the network topology between FAM and the PNN. Notice that in Figures 1 and 2, the F_1^a and F_2^a layers correspond to the input and pattern layers, whereas the map field layer (F^{ab}) corresponds to the summation layer. In essence, in one-from-*N* classification, each node in F_2^a is permanently associated with only one node in F^{ab} which is then linked to the target output in F_2^b. Thus, the F^{ab} nodes can be used to sum outputs from all the F_2^a nodes corresponding to a particular target category, taking the role of the summation units in the PNN.

In view of the suitability of the incremental learning property and the similarity of the network topology between FAM and the PNN, a novel hybrid network, based on the integration upon a modified version of FAM [23] and the PNN, has been proposed for on-line classification and probability estimation tasks, and is called Probabilistic Fuzzy ARTMAP (PFAM) [14] [15]. The advantage of this integration is two-fold: (i) a probabilistic interpretation of output classes is established which enables the application of Bayes, risk-weighted, classification in FAM; (ii) the number of pattern nodes in the PNN is reduced by the clustering procedure of FAM. The on-line PFAM algorithm is divided into two phases. First, the FAM clustering procedure is used for classifying the input patterns into different categories (learning phase). Subsequently, the PNN probability estimation procedure is used to predict a target output (prediction phase).

The above description provides a conceptual framework for incorporating FAM and the PNN into a unified, hybrid system, and the rationale behind their integration. In practice, several modifications are necessary to allow effective combination of both the networks, and to increase generalization ability of the resulting system. These include modifications to the map field dynamics during the learning phase as well as the procedures to estimate kernel centers and widths for probability estimation during the prediction phase. A summary of the PFAM algorithm is given in Appendix A.

2.4 Multiple Classifier Systems

In this paper, three methods for combining decisions from multiple classifiers has been implemented to form a PFAM-based Multiple Classifier System (MCS), as shown in Figure 3. The first one is a simple majority voting scheme where the target class which receives the highest number of votes is selected as the winner. However, in this case, each classifier is treated equally as one vote without considering its predictive errors. This leads to the second approach where the predictive accuracy of each classifier is taken into account in the combination process using the Bayesian approach [3]. In the Bayesian formalism, highly accurate classifiers' results are given more weight than less accurate ones by exploiting the confusion matrices formulated from their previous predictions. Nevertheless, one of the criticisms of the Bayesian approach is the assumption that all classifiers operate independently in order to tackle the computation of the joint probabilities. This may not be true in applications. A third combination method proposed in [4] is employed. A Behavior-Knowledge-Space (BKS) is introduced to record the decisions of all classifiers on each learned sample concurrently. This approach has a close similarity with the Bayesian formalism, but without the assumption of independence of the classifiers involved. Appendix B presents a description of the Bayesian and BKS decision combination algorithms.

The PFAM-based MCSs can be used in two ways:

a) the concatenation approach—where the data samples are used in their original form. In this approach, multiple PFAM classifiers are first trained on different orderings of the training samples, and then tested on the test samples. The predictions from these classifiers are combined using the voting, Bayesian or BKS methods to give an overall decision.

b) the modular approach—where each data sample is divided into groups of relevant attributes. In the training phase, each group of attributes of a training item is assigned to a PFAM classifier. In the test phase, the predictions from these classifiers, each on a group of attributes of a test item, are combined using the three combination methods to give an overall decision. However, this approach does not alter the problem of problem of data ordering although the extension to this case is straight forward.

The above two approaches correspond to two variants of the ART systems, namely FAM and Fusion ARTMAP [24] [25]. Fusion ARTMAP is an extension of FAM. It introduces a modularized technique for sensor data fusion and classification. Figure 4 shows a schematic diagram of the Fusion ARTMAP network. In Fusion ARTMAP, an independent, unsupervised ART classifier module (ART$_c$) is employed for each input sensor. The outputs from these ART$_c$ modules (representing the compressed recognition codes of different sensors) are concatenated to form an input to a global ART classifier (ART$_a$) to make an overall prediction. A target output classifier, ART$_b$, is employed to moderate supervised learning for the system via the map field. In response to any predictive error, a *parallel match tracking* mechanism is initiated. Parallel match tracking raises the vigilance parameter of ART$_c$ classifiers simultaneously until the module with the lowest level of match between the input and prototype patterns is found, and reset. The advantage here is that a global classification error only resets the classifier with the least confidence in its prediction without affecting those with higher confidence. As a result, each classifier can be dynamically, and independently, re-configured to correct its own errors. In addition, a parsimonious network connection can be achieved to reduce the complexity of the resulting system.

There are a few differences between the Fusion ARTMAP network and a PFAM-based MCS. In Fusion ARTMAP, outputs from all the ART$_c$ modules have to be concatenated to form an input pattern to a global classifier before a prediction is given. A MCS does not include this secondary concatenation step, and avoids any dependence on a global classifier to reach a final outcome. Instead, various methods can be adopted as the decision combination algorithm to integrate predictions from multiple classifiers. On the other hand, in response to an incorrect prediction, Fusion ARTMAP incorporates a parallel match tracking mechanism to deactivate the winning category in the least confident classifier, and to initiate a new search phase to look for a better category prototype. This feedback corrective action is absent in a PFAM-based MCS.

3 The Experiments

In the following sections, two benchmark classification problems are employed to compare the performance of the voting, Bayesian, and BKS methods. Another aspect of interest is to examine the effects of the concatenation and modular approaches in forming input samples to the MCSs. In addition, the three combination schemes include a variable confidence threshold, λ, to regulate the reliability of the final outcome. For example, in voting, one could state that the vote count of the predicted class must more than two-thirds of the total votes in order to accept the decision. The same principle applies to the Bayesian and BKS approaches where the combined beliefs of the predicted class have to exceed λ before accepting the prediction. The effects of this confidence threshold on the classification results are also studied in detail. The two benchmark problems have been studied by Asfour [25] with FAM and Fusion ARTMAP, hence the results can be used as a baseline comparison with those obtained from the concatenation and modular PFAM-based MCSs.

3.1 Quadruped Mammals Data Set

This data set is an artificial domain representing quadruped animals to evaluate the CLASSIT unsupervised machine learning algorithm [26]. There are four types of mammals, namely *cats*, *dogs*, *horses*, and *giraffes*. Each instance is described by a set of eight cylinders representing 8 different components: head, tail, neck, torso, and four legs, as shown in Figure 5. Each cylinder includes 9 attributes: texture, height, radius, three locations, and three axes. Hence, there are 72 attributes per instance. According to [26], it is believed that real-world objects have at least such an order of complexity, and that a robust concept formation system should be able to handle instances of this form. This representation of objects can be viewed as a simplification of Binford's generalized cylinders [27] which have received wide attention within the machine vision community. In addition, such representations are considered as reasonable approximations of the output of human's visual system [28]. The CLASSIT unsupervised learning algorithm achieves

95% accuracy after learning 35 instances of different quadruped animals incrementally [26]. The program for generating the data samples used in this experiment is obtainable from the UCI Repository of Machine Learning Databases and Domain Theories [29].

3.1.1 Off-line Learning

(a) Results of the Concatenation Approach

Asfour [25] conducted some experiments on this quadruped mammal data set using single-channel FAM and multi-channel Fusion ARTMAP. In Fusion ARTMAP, each of the 8 components (head, tail, neck, torso, and four legs—each with 9 attributes) was presented to a different ART classifier module. Two different training set sizes of 100 and 1000 samples were applied, and the trained system was tested on 1000 samples. The performance, averaged over 3 runs, was compared to the concatenation approach applying of all 72 attributes to a single FAM classifier.

Here, two simulations were conducted using the same number of training and test set sizes. Throughout all the simulations, the important PFAM parameters used were: baseline vigilance parameter, $\overline{\rho}_a = 0.0$; learning rate, $\beta_a = 1.0$ (fast learning); $\alpha_a \approx 0.0$ (conservative mode) [16]; overlapping parameter, $r = 1.0$ [15]; confidence threshold of MCS, $\lambda = 0.0$ (Appendix B). The first simulation used five PFAM classifiers trained on five different sets of training samples. In addition to averaging the five individual results, MCSs (concatenation approach) were formed to combine the outcomes. Table 1 lists all the experimental results. The FAM performance reported in [25] is also included to serve as a comparison.

Training Set Size	FAM	PFAM	Multiple Classifier System		
			Voting	Bayesian	BKS
100	100%	97.5%	99.5%	100%	100%
1000	100%	99.4%	100%	100%	100%

Table 1 Performance of individual and multiple classifiers using the concatenation approach for the Quadruped Mammal database.

A few observations can be made from Table 1. The performance of PFAM is inferior to that of FAM which shows perfect results in both simulations. The failure of PFAM to achieve the same performance as FAM might be due to the small number of ART$_a$ categories (prototype patterns) being created in the system. For PFAM, the average number of prototypes was 6.2 for

100 training samples, and 8.1 for 1000 training samples. Recall that the prediction phase of PFAM utilizes the Parzen-window probability estimation procedure. Accuracy of the estimated pdf depends on the number of prototypes available. The larger the number of prototypes, the more accurate the estimated pdf becomes. Theoretically, the pdf will converge asymptotically to the actual underlying function as the number of prototypes extends to infinity. As can be seen in Table 1, performance of PFAM improves as the training samples increase from 100 to 1000, in which case more prototypes were established.

In comparing performance between single and multiple classifiers, MCSs are able to improve the results of individual classifiers. This is true for all the three decision combination methods. On the other hand, the more complicated approaches of Bayesian and BKS show a better performance than the simple majority voting strategy.

(b) Results of the Modular Approach

In this experiment, three MCSs were formed, each with 8 modules of PFAM classifiers. Each classifier was dedicated to one group of the data components—head, tail, neck, torso, and four legs. All the simulations were repeated 5 times with 5 randomly-ordered data sets. The averaged results of individual classifiers are listed in Table 2. The performance of MCSs and the Fusion ARTMAP results reported in [25] are shown in Table 3.

Again, the multiple classifier approach proves to be useful in improving the performance of single classifiers. Perfect results (100% accuracy) were obtained with the Bayesian and BKS approaches over 5 runs using both 100 and 1000 training samples. One improvement is in the voting results where the concatenation approach achieved 99.5% accuracy (Table 1), but the modular approach achieved 99.9% accuracy (Table 3). The results listed in Table 3 are also better than that of Fusion ARTMAP for the 100-sample case. Nevertheless, by increasing the training samples to 1000, all the classifiers were able to perform with perfect accuracy.

When individual classifiers are concerned, the concatenation approach (Table 1—FAM and PFAM) achieve better performance than the modular approach (Table 2). By reducing the input dimension from 72 to 8, fewer prototypes were formed using the modular approach (average numbers of prototypes were between 4.4 and 8.0). As a result, the estimated pdf tends to be less accurate compared with the concatenation approach, which in turn causes a lower classification accuracy. However, this drawback can be overcome by combining the decisions from multiple disparate classifier modules, as shown in Table 3.

Classifier	Accuracy (%)	
	100 Training Samples	1000 Training Samples
Head	97.6	98.5
Neck	98.9	99.0
Torso	96.6	98.2
Tail	97.3	97.4
Leg 1	97.9	98.5
Leg 2	96.9	98.8
Leg 3	96.6	99.7
Leg 4	97.5	97.8

Table 2 Performance of individual classifiers using the modular approach for the Quadruped Mammal database.

Training Set Size	Fusion ARTMAP	Multiple Classifier System		
		Voting	Bayesian	BKS
100	96%	99.9%	100%	100%
1000	100%	100%	100%	100%

Table 3 Performance of multiple classifiers using the modular approach for the Quadruped Mammal database.

3.1.2 On-line Learning

In on-line learning mode, the system imitates the condition of a human operating in a natural environment. Each incoming datum is used as a training sample as well as a test sample. The on-line operational cycle proceeds as follows: an input pattern is first presented to ART$_a$, with its target class to ART$_b$. A prediction is sent from ART$_a$ to ART$_b$, and compared with the actual class. The outcome constitutes a classification result. Then learning ensues to associate the input vector with its target class.

The modular approach with 8 classifiers (each has 9 attributes) was compared to the concatenation approach of a single PFAM classifier. Note that in on-line learning, the voting, Bayesian and BKS methods are not applicable to results of the concatenation approach. This is because there is no differentiation between training and test sets—all the data samples are first tested and then trained in a fixed order. Hence, knowledge established in multiple classifiers will be the same if they are trained on the same sequence of samples. In the modular approach, however, each classifier is trained on only a group of attributes of the sample. Therefore, MCSs can be formed for combining the predictions from classifiers based on disparate attribute groups.

In this experiment, 1000 data samples were generated. To calculate the on-line accuracy, a 100-sample window was applied, *e.g.* accuracy at sample 200 was the percentage of correct predictions from trials 101-200. All the on-line simulations were averages across 5 runs. Figure 6 depicts a comparison of the on-line results between the concatenation and modular approaches. The standard deviations of 5 runs are plotted as error bars to indicate the spread of individual results across the averages. As can be seen, the modular approach attained a performance superior to that of the concatenation approach. Perfect results (100% accuracy) were achieved by the Bayesian and BKS formalisms after encountering fewer than 200 incoming samples. This perfect performance was maintained until the end of the experiments in all 5 attempts. Although voting exhibits inferior results to the Bayesian and BKS formalisms, it still outperforms the concatenation approach.

3.2 Landsat Satellite Images

This database was generated from data purchased from NASA by the Australian Center for Remote Sensing, and used for research at the University of New South Wales [30]. It consists of a small sub-section (82x100 pixels) of a scene from the original data set, where each pixel covers an area of approximately 80x80 meters on the ground. One frame of the Landsat Multi-Spectral Scanner (MSS) imagery comprises intensities of four spectral bands of the same scene. Two of the spectra are in the visible region (corresponding approximately to green and red regions), and two are in the (near) infra-red region.

Again, the Landsat MSS data is obtainable from the UCI Repository of Machine Learning Databases and Domain Theories [29]. The data set is divided into a training set of 4435 samples, and a test set of 2000 samples. Each sample comprises 36 integer-valued attributes—4 spectral values for each pixel of a 3x3 neighborhood [29]. The target output is the class label of the central pixel in one of the 6 classes—red soil, cotton crop, gray soil, damp gray soil, soil with vegetation stubble, and very damp gray soil[*]. The aim of this data set is to predict the classification associated with the central pixel, given the multi-spectral values.

The Landsat MSS data set serves as a more challenging benchmark problem as it comprises real and noise-corrupted satellite images. Asfour [25] has also used this data set to assess the performance of FAM and Fusion ARTMAP. Furthermore, many classification algorithms have been evaluated using this data as part of the Statlog project [30], and the results are listed in Table 4. Thus, it is useful to compare performance of our MCS with other approaches.

Algorithm	Accuracy (%)
k-NN	90.6
LVQ	89.5
DIPOL92	88.9
RBF	87.9
ALLOC80	86.8
CART	86.2
IndCART	86.2
Back-prop	86.1
Baytree	85.3
NewID, CN2, C4.5	85.0
Cal5	84.9
Quadisc	84.5
AC^2	84.3
SMART	84.1
Logdisc, Cascade	83.7
Discrim	82.9
Kohonen	82.1
CASTLE	80.6
NaiveBay	71.3
Default	23.1

Table 4 Classification accuracy from various algorithms on the Landsat Satellite data set [30]. The Default accuracy is calculated by categorizing all the test samples as belonging to the class that has the highest number of samples, i.e. maximum *a priori* classification.

3.2.1 Off-line Learning

With the concatenation approach, the input sample to PFAM consisted of a 36-dimensional vector which had been normalized between 0 and 1. With the modular approach, the same input sample was divided into 4 spectral bands, each comprising a 9-dimensional vector (normalized between 0

[*] Classification for each pixel was performed on the basis of an actual site visit by Ms. Karen Hall, when working for Professor John A. Richards, at the Center for Remote Sensing, University of New South Wales, Australia; conversion to 3x3 neighborhoods, and splitting into training and test sets was done by Alistair Sutherland [29].

and 1) corresponding to a 3x3 pixel grid from a different spectral scanner. The training data was randomized to produce 5 differently ordered training sets. The results tabulated below are the averages of 5 runs for individual PFAMs, and the combined 5 run results for the MCSs.

(a) Results of the Concatenation Approach

Table 5 shows the results of FAM in [25] and the results of PFAM as well as the MCSs from our experiments. Two baseline vigilance values were tested, *i.e.* a low value of $\overline{\rho}_a = 0.0$ to create coarse recognition categories of input samples, and a high value of $\overline{\rho}_a = 0.9$ to create finer recognition categories. The rest of the important parameters were the same as for the quadruped mammal experiment. As might be expected, a considerable improvement in classification accuracy was achieved with $\overline{\rho}_a = 0.9$. The trade-off being the increase numbers of ART$_a$ categories created by FAM and PFAM which were about 8 fold and 6 fold, respectively. Nevertheless, the *k*-NN approach, which achieved the best accuracy in Table 4, had to store all 4435 training samples as prototype nodes [25] [30]. Thus, when operating at $\overline{\rho}_a = 0.9$, FAM and PFAM could achieve a slightly lower accuracy (89% vs. 90.6%) but with a higher degree of code compression.

Algorithm		$\overline{\rho}_a = 0.0$	$\overline{\rho}_a = 0.9$
FAM	Accuracy (%)	83.0	89.0
	No. of ART$_a$ Categories	89	704
PFAM	Accuracy (%)	81.4	89.0
	No. of ART$_a$ Categories	87	518
Accuracy of MCS (%)	Voting	86.1	90.8
	Bayesian	87.0	91.6
	BKS	91.7	94.5

Table 5 Performance of single and multiple classifiers using the concatenation approach for the Landsat MSS database.

In terms of performance comparison, the three MCSs operating at $\overline{\rho}_a = 0.9$ outperformed not only FAM and PFAM, but also all the classification results from a variety of algorithms listed in Table 4. The BKS approach, again, achieved the best results of 91.7% for $\overline{\rho}_a = 0.0$ and

94.5% for $\overline{\rho}_a = 0.9$. All three MCSs were also able to improve on the results of individual PFAM classifiers.

With $\overline{\rho}_a = 0.0$, the performance of PFAM was relatively poor when compared with most other approaches in Table 4. This may be accounted for by the same phenomenon observed in the quadruped mammal experiments, *i.e.* accuracy of the estimated pdf is directly affected by the number of prototype nodes. That is why a substantial improvement could be achieved by PFAM with $\overline{\rho}_a = 0.9$, in which more than five hundred prototype nodes were created.

(b) Results of the Modular Approach

In addition to the concatenated input samples, a modular approach of dividing the input sample attributes into the four corresponding scanner modality bands has been tested. Table 6 summarizes the average results in terms of accuracy and the number of ART$_a$ categories of 5 runs

The modular approach proved to be a failure with the Landsat MSS data set. As can be seen from Table 6, not only were the classification results exceptionally poor, but there was a proliferation of prototype nodes. Even by raising $\overline{\rho}_a = 0.9$, the results were still significantly inferior to those using the concatenation approach. Among the 4 spectral bands, the accuracy of band 3 classifier was the worst with the highest number of prototype nodes.

Classifier		$\overline{\rho}_a = 0.0$	$\overline{\rho}_a = 0.9$
Band 1	Accuracy (%)	52.7	56.2
	No. of ART$_a$ Categories	891	991
Band 2	Accuracy (%)	51.2	54.0
	No. of ART$_a$ Categories	628	1021
Band 3	Accuracy (%)	37.3	42.2
	No. of ART$_a$ Categories	1063	1375
Band 4	Accuracy (%)	50.9	54.9
	No. of ART$_a$ Categories	766	1099

Table 6 Performance of individual classifiers using the modular approach for the Landsat MSS database.

Deficiency of the modular approach is also experienced with the Fusion ARTMAP network. According to [25], the best performance of Fusion ARTMAP was about 70%, and the same problem of category proliferation was also observed. Failure of the modular approach on this data set is due to two factors: inter-spectral dependency and the presence of noise in the satellite images. Among the Statlog databases, the Landsat satellite database is the only data set that has very large correlation between attributes of the input samples [30]. By segmenting the attributes into different scanner bands, inter-spectral information is lost. Without the benefits of information from other spectra, noise in the data is aggregated. For example, the "red" sensor is insensitive to certain "green" frequencies, and it will produce very irregular images associated with different output classes when the region under scrutiny is mostly vegetation [25]. Without reinforcement information from the "green" sensor on vegetation imagery, inputs to the "red" classifier would more likely mismatch the target classes, which in turn would result in more new categories being created to encode the samples. Similar reasons of inter-spectral dependency and sensor insensitivity also accounted for the category proliferation problem in other classifiers, as observed in the experimental results. This pitfall is avoided in FAM when all the sensor information are concatenated into a single input sample. The resulting input samples would be more consistent with the target classes, and therefore enable the FAM system to achieve better generalization with a reduced number of prototype nodes.

MCS	$\overline{\rho}_a = 0.0$	$\overline{\rho}_a = 0.9$
Voting	59.4%	65.6%
Bayesian	69.2%	74.8%
BKS	83.3%	86.1%

Table 7 Performance of multiple classifiers using the modular approach for the Landsat MSS database.

Table 7 shows the results, averaged over 5 runs, of the MCSs by combining the outcomes from the 4 individual classifiers. Generally, classification accuracy was improved with the voting, Bayesian or BKS methods. This experiment, once again, demonstrated the benefits of using multiple classifiers in improving the performance of single classifiers when dealing with classification problems. In particular, the BKS approach produced an increase of more than 30% in accuracy with both $\overline{\rho}_a = 0.0$ and $\overline{\rho}_a = 0.9$ (these results are comparable with those of concatenation approach shown in Tables 4 and 5).

(c) Results of Variable Confidence Threshold

As pointed out earlier, each decision combination scheme includes a threshold to regulate the confidence associated with the predicted outcome from multiple classifiers. This variable confidence threshold, $0 \leq \lambda \leq 1$, can be manipulated either to accept the classifier's prediction if the combined belief is higher than λ, or to reject the prediction otherwise. As a result, reliability of the classifier's performance can be adjusted accordingly. To examine the effects of λ, the following terms are recommended to quantify the performance [3] [4]:

Recognition rate: ratio of the number of correct classifications to the total number of samples (*i.e.* the accuracy index in previous experiments);

Substitution rate: ratio of the number of incorrect classifications to the total number of samples;

Rejection rate: ratio of the number of rejected classifications to the total number of samples;

Reliability rate : $Reliability = \dfrac{Recognition}{1 - Rejection}$

In the following experiments, these parameters are expressed as percentages.

The experimental results of multiple classifiers in subsection (b) (the modular approach) were re-evaluated using different threshold values. Figure 7 depicts a plot of the substitution rate against the recognition rate, parameterized by the confidence threshold. The plot illustrates that by increasing λ from 0 to 1, more and more input samples are being rejected as it becomes more difficult for the predicted output to satisfy the confidence level, *i.e.* trade-off between recognition and rejection took place as λ was increased from zero to unity. From another point of view, manipulation of the confidence threshold provides a means for designing a classifier system to perform with high or low reliability to suit the problem under investigation. Different values of confidence threshold also result in different performances for the three decision combination schemes. From Figure 7, one can see that both the recognition and substitution rates gradually decrease to 0% while the reliability rate increases to 100% since the system has to be very "confident" in any predicted answer. Notice that even in the situation of low substitution rates, the BKS approach is able to maintain a relatively high recognition rate (and high reliability) compared with the Bayesian and voting methods. These observations are true for both experiments using $\overline{\rho}_a = 0.0$ and $\overline{\rho}_a = 0.9$.

3.2.2 Dual-mode Learning

In cases when there is a high correlation and dependency between attributes of the input samples, segmenting these attributes into different classifiers offers no benefits at all. This situation can be observed in the experiments with the Landsat MSS data set where the concatenation approach performed significantly better than the modular approach using individual classifiers. As a result, on-line learning experiments with modularized inputs, such as those conducted for the quadruped mammal data, do not seem to be appropriate in this regard.

With the concatenation approach, on-line combination of decisions across an ensemble of classifiers is unrealizable as in-coming data is received in a fixed order. It therefore seems that the MCSs are not applicable in on-line learning environments using the concatenation approach. In practice, however, there is no reason why such MCSs could not be employed on-line, after an initial period of training. As a result, a dual-mode learning strategy is devised where an off-line learning process is first carried out to equip each individual classifier with a "knowledge" base before on-line learning is initiated. The fact that autonomy is lost in the early stages may not be a disadvantage. This is because, in any application, a series of trials on historical data will normally be required before the system is allowed to become operational. In off-line learning, different classifiers will establish different category prototypes, thus predictions will be different when they are switched to on-line learning even though the classifiers are now facing incoming samples in a specific order. The decision combination schemes, once again, can be implemented to combine outcomes from a variety of differently trained classifiers using concatenated samples. We call this strategy *dual-mode learning*.

Two sets of dual-mode learning experiments were conducted with $\overline{\rho}_a = 0.0$ and $\overline{\rho}_a = 0.9$. Five classifiers were initially trained with 1000 samples, each with a different ordering of data. The five trained classifiers were then tested on the remaining 5435 samples. Because learning continued in the test phase, the classification accuracy was calculated with a 1000-sample moving window as in the on-line learning experiments. Figure 8 shows the overall classification accuracy against increasing number of input samples for the average results of five classifiers, as well as the voting, Bayesian and BKS MCSs. A few observations can be drawn from the dual-mode learning results: (1) performance improves with a high vigilance value (Figures 8(a) and 8(b) are plotted using the same scale for the *y*-axis to illustrate the improvement on the classification results when

$\overline{\rho}_a$ is increased from 0.0 to 0.9); (2) all three MCSs perform better than the average of the individual classifiers; (3) the BKS approach performs the best, followed by the Bayesian approach and then the voting method. Nevertheless, with $\overline{\rho}_a = 0.9$, both the Bayesian and voting methods achieved virtually the same performance.

In comparison with the off-line learning approach, one would notice that the on-line or dual-mode learning strategies employ more data samples for learning as all input samples are treated as test data as well as training data. This is, in fact, the advantage of on-line learning systems, such as PFAM, which are capable of learning incrementally and continually. Instead of the conventional "train/test" method where a system is only allowed to learn from the training samples, the PFAM system makes use of as much information as possible in an attempt to find a good network configuration to solve the problem under investigation. Using the dual-mode learning strategy, the classifiers are equipped with some "knowledge" about the environment before they are put into use, and hence they do not have to start from a naive condition as in purely on-line operations. In addition, the system is able to learn continually in a possibly non-stationary environment. Thus, the problems associated with off-line learning such as pre-determined network size and re-training can be avoided. This on-going (causal) learning ability is the reason why ART-based networks might be preferred over other types of systems for the development of autonomous agents.

4 Summary

A composition of multiple neural classifier systems has been studied to solve pattern classification problems. The classifier used is based on a hybrid system of FAM and PNN neural networks. It is an incremental adaptive system capable of on-line, supervised learning and probability estimation. In the manifestation of multiple classifiers presented, an independent classifier module is dedicated to handle a set of attributes, and outputs from these classifiers are then combined using some decision combination schemes to give an overall prediction. Three algorithms have been implemented to integrate the results of different classifiers, namely the majority voting, Bayesian, and BKS methods. The efficacy of these algorithms have been studied experimentally using two benchmark data sets taken from a public-domain repository.

Various parameter settings have been investigated using two benchmark problems taken from a public domain repository—the Quadruped mammal and Landsat MSS data sets. From the experiments, it is found that multiple classifiers are able to enhance the performance of individual classifiers. Among the three decision combination algorithms, the BKS approach achieves the best performance followed by the Bayesian and voting methods. In addition, the effect of the confidence threshold has also been studied. In comparison with the voting and Bayesian methods, the BKS approach is able to maintain a relatively high level of classification accuracy (recognition) with a high degree of reliability under the circumstance of high confidence threshold. This finding corresponds to the results reported in [5] where the BKS method achieved a superior performance compared to the voting and Bayesian strategies in a hand-written character recognition task. The superiority of the BKS method might because of the inclusion of the predictive accuracy of each classifier (unlike voting) and the exclusion of the independence assumption of each classifier (unlike the Bayesian approach) in combining the results from multiple classifiers.

Apart from the usual concatenation approach to represent input patterns, a modular approach has been examined where the input patterns are segmented into groups of related attributes. The use of this modular approach is strongly dependent on the correlation between the input attributes. If strong correlation exists, the modular approach will not only diminish the overall performance, but will also induce unnecessary complexity of the hybrid classifier system used in this paper.

Since the classifier is able to operate on-line, experiments have also been conducted to study its incremental learning properties. One practical strategy is to employ a dual-mode learning approach where the classifiers are trained with a set of samples with different ordering before they are put into use. This approach helps establish a knowledge base in each classifier before on-line learning and prediction is engaged. With the use of the dual-mode learning strategy and the combination of decisions from multiple classifiers, a robust and accurate autonomously learning classification system can be realized.

References

[1] B. Duerr, H. Haettich, H. Tropf, and G. Winkler, "A Combination of Statistical and Syntactical Pattern Recognition Applied to Classification of Unconstrained Handwritten Numerals", *Pattern Recognition*, vol. 12, pp. 189-199, 1980.

[2] T.K. Ho, J.J. Hull, and S.N. Srihari, "Decision Combination in Multiple Classifier Systems", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 16, pp. 66-75, 1994.

[3] L, Xu, A. Krzyzak, and C.Y. Suen, "Methods of Combining Multiple Classifiers and Their Applications to Handwriting Recognition", *IEEE Transactions on Systems, Man, and Cybernetics*, vol 22, pp. 418-435, 1992.

[4] Y.S. Huang, and C.Y. Suen, "A Method of Combining Multiple Experts for the Recognition of Unconstrained Handwritten Numerals", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 17, pp. 90-94, 1995.

[5] R.O. Duda, and P.E. Hart, *Pattern Classification and Scene Analysis*, New York: John Wiley and Sons, 1973.

[6] G. Cybenko, "Approximation by Superposition of a Sigmoidal Function", *Mathematics of Control, Signals and Systems*, vol 2, pp. 303-314, 1989.

[7] F. Girosi, and T. Poggio, "Networks and the Best Approximation Property", *Biological Cybernetics*, vol 63, pp. 169-176, 1990.

[8] E.A. Wan, "Neural Network Classification: A Bayesian Interpretation", *IEEE Transactions on Neural Networks*, vol 1, pp. 303-305, 1990.

[9] H. White, "Learning in Artificial Neural Networks: A Statistical Perspective", *Neural Computation*, vol 1, pp. 425-464, 1989.

[10] M.D. Richard, and R.P. Lippmann, "Neural Network Classifiers Estimate Bayesian *a posteriori* Probabilities", *Neural Computation*, vol 3, pp. 461-483, 1991.

[11] M. McCloskey, and N.J. Cohen, "Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem", In Bower, G.H. (Ed.). *The Psychology of Learning and Motivation*, New York: Academic Press, 1989.

[12] R. Ratcliff, "Connectionist Models of Recognition Memory: Constraints Imposed by Learning and Forgetting Functions", *Psychological Review*, vol 96, pp. 523-568, 1990.

[13] N.E. Sharkey, and A.J.C. Sharkey, "An Analysis of Catastrophic Interference", *Connection Science*, vol 7, pp. 301-329, 1995.

[14] C.P. Lim, and R.F. Harrison, "Probabilistic Fuzzy ARTMAP: An Autonomous Neural Network Architecture for Bayesian Probability Estimation", *Proceedings of the fourth IEE International Conference on Artificial Neural Networks*, pp. 148-153, 1995.

[15] C.P. Lim, and R.F. Harrison, "An Incremental Adaptive Network for On-line Supervised Learning and Probability Estimation", Accepted for publication in *Neural Networks*.

[16] G.A. Carpenter, S. Grossberg, N. Markuzon, J.H. Reynolds, and D.B. Rosen, "Fuzzy ARTMAP: A Neural Network Architecture for Incremental Supervised Learning of Analog Multidimensional Maps", *IEEE Transactions on Neural Networks*, vol 3, pp. 698-712, 1992.

[17] D.F. Specht, "Probabilistic Neural Networks", *Neural Networks*, vol 3, pp. 109-118, 1990.

[18] C.P. Lim, and R.F. Harrison, "Estimation of Bayesian *a posteriori* Probability with an Autonomously Learning Neural Network", *Proceedings of UKACC International Conference on Control '96*, vol 1, pp. 199-204, 1996.

[19] G.A. Carpenter, and S. Grossberg, "A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine", *Computer Vision, Graphics and Image Processing*, vol 37, 54-115, 1987.

[20] G.A. Carpenter, S. Grossberg, and D.B. Rosen, "Fuzzy ART : Fast Stable Learning and Categorization of Analog Patterns by an Adaptive Resonance System", *Neural Networks*, vol 4, 759-771, 1991.

[21] J.A. Hartigan, *Clustering Algorithms*, New York: John Wiley and Sons, 1975.

[22] E. Parzen, "On Estimation of a Probability Density Function and Mode", *Annals of Mathematical Statistics*, vol 33, pp. 1065-1076, 1962.

[23] C.P. Lim, and R.F. Harrison, "Modified Fuzzy ARTMAP Approaches Bayes Optimal Classification Rates: An Empirical Demonstration", Accepted for publication in *Neural Networks*.

[24] R.Y. Asfour, G.A. Carpenter, S. Grossberg, and G.W. Lesher, "Fusion ARTMAP: A Neural Network Architecture for Multi-channel Data Fusion and Classification", *Proceedings of World Congress on Neural Networks*, Vol. II, pp. 210-215, 1993.

[25] R.Y. Asfour, "Fusion ARTMAP: Neural Networks for Multi-sensor Fusion and Classification", *Ph.D. Thesis*, Boston University, 1995.

[26] J.H. Genneri, P. Langley, and D. Fisher, "Models of Incremental Concept Formation", *Artificial Intelligence*, vol 40, pp. 11-61, 1989.

[27] T.O. Binford, "Visual Perception by Computer", *IEEE Conference on Systems and Control*, Miami, 1971.

[28] D. Marr, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. San Francisco: Freeman, 1982.

[29] P.M. Murphy, and D.W. Aha, UCI Repository of Machine Learning Databases, [Machine-readable Data Repository]. Irvine, CA: University of California, Department of Information and Computer Science, 1995.

[30] D. Michie, D.J. Spiegelhalter, and C.C. Taylor, *Machine Learning, Neural and Statistical Classification.* Oxford: Oxford Press, 1994.

[31] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, San Mateo, CA: Morgan Kaufmann, 1988.

Appendix A The Probabilistic Fuzzy ARTMAP Algorithm

Following the notation used in the FAM paper [16], let $2M_a$ be the number of nodes in F_1^a and N_a be the number of nodes in F_2^a. The Short Term Memory (STM) traces or activity vectors of F_1^a and F_2^a are denoted by $\boldsymbol{x}^a \equiv (x_1^a, \cdots, x_{2M_a}^a)$ and $\boldsymbol{y}^a \equiv (y_1^a, \cdots, y_{N_a}^a)$, and $\boldsymbol{w}_j^a \equiv (w_{j1}^a, \cdots, w_{j,2M_a}^a)$, $j = 1, \cdots, N_a$ is the jth ART$_a$ weight vector or the Long Term Memory (LTM) trace. All the notation applies to ART$_b$ when the superscripts or subscripts a and b are interchanged. In the map field, $\boldsymbol{w}_j^{ab} \equiv (w_{j1}^{ab}, \cdots, w_{j,N_b}^{ab})$, $j = 1, \cdots, N_a$ is the weight vector from the jth F_2^a node to F^{ab}, and $\boldsymbol{x}^{ab} \equiv (x_1^{ab}, \cdots, x_{N_b}^{ab})$ is the map field activity vector.

In ART$_a$, each F_2^a category node weight vector fans-out to all the nodes in the F_1^a layer. These weight vectors are initialized to unity, i.e.

$$w_{j1}^a(0) = \cdots = w_{j,2M_a}^a(0) = 1 \qquad j = 1, \cdots, N_a$$

There are three parameters associated with ART$_a$ (as well as ART$_b$), namely the choice parameter, α_a, learning rate, β_a, and baseline vigilance parameter, $\overline{\rho_a}$. To operate in the conservative mode where recoding during learning will be minimized, α_a should be initialized close to 0, i.e. $\alpha_a \to 0$. The values of β_a and $\overline{\rho_a}$ are set between 0 and 1. The same initialization procedure is also applicable to ART$_b$. In the map field, the vigilance parameter, ρ_{ab}, is also initialized between 0 and 1, whereas the weight vectors from F_2^a to F^{ab} are set to unity, i.e.

$$w_{j1}^{ab}(0) = \cdots = w_{j,N_b}^{ab}(0) = 1 \qquad j = 1, \cdots, N_a$$

Note that the number of nodes in F^{ab} is the same as the number of nodes in F_2^b, and there is a one-to-one permanent link between each corresponding pair of nodes.

The entire algorithm of the proposed PFAM for on-line learning and probability estimation is as follows:

Learning Phase:

(i) Complement-code an M-dimensional input vector, $\mathbf{a} \in [0,1]^M$, in F_0^a to a $2M$-dimensional vector in F_1^a as:

$$A = (\mathbf{a}, 1-\mathbf{a}) \equiv (a_1, \cdots, a_m, 1-a_1, \cdots, 1-a_m)$$

(ii) Feed forward A from F_1^a to F_2^a via $\mathbf{w}^a$. Compute the match function as

$$T_j(A) = \frac{\left|A \wedge \mathbf{w}_j^a\right|}{\alpha_a + \left|\mathbf{w}_j^a\right|} \qquad j = 1, \cdots, N_a$$

Select the winning node, and denote it as node J.

(iii) Feed back the prototype of the winning node from F_2^a to F_1^a and perform the vigilance test as

$$\frac{\left|\mathbf{x}^a\right|}{|A|} = \frac{\left|A \wedge \mathbf{w}_J^a\right|}{|A|} \geq \rho_a$$

(iv) If the vigilance test fails, trigger the search cycle and go to step (ii).

(The above cycle goes on in ART$_b$ simultaneously)

(v) The comparison between F_2^a and F_2^b activities takes place in the map field. If K is the winning node in ART$_b$, then

$$y_k^b = \begin{cases} 1 & \text{if} \quad k = K \\ 0 & \text{otherwise} \end{cases} \qquad k = 1, \cdots, N_b$$

Assuming that that both ART$_a$ and ART$_b$ are active, the F^{ab} activity vector, $\mathbf{x}^{ab}$, obeys

$$\mathbf{x}^{ab} = \mathbf{y}^b \wedge \frac{\mathbf{w}_J^{ab}}{\left|\mathbf{w}_J^{ab}\right|}$$

which forms a prediction from the Jth ART$_a$ category to the Kth ART$_b$ target class via $\mathbf{w}_J^{ab}$.

(vi) Perform the map field vigilance test as

$$\frac{\left|\mathbf{x}^{ab}\right|}{\left|\mathbf{y}^b\right|} \geq \rho_{ab}$$

(vii) If the map field vigilance test fails, trigger match-tracking where ρ_a is raised to

$$0 \le \rho_a \le \min\left(1, \frac{|A \wedge w_{a-J}|}{|A|} + \delta\right)$$

where δ a small positive value slightly greater zero and go to step (ii)

(viii) Update the weight vectors as follows:

ART$_a$ weights: $\left(w_J^a\right)^{\text{new}} = \beta_a\left(A \wedge \left(w_J^a\right)^{\text{old}}\right) + (1-\beta_a)\left(w_J^a\right)^{\text{old}}$

Map field weights: $w_J^{ab} = w_J^{ab} + x^{ab}$

A new set of weight vectors is introduced in PFAM, called the center weight vectors (the original weight vectors, w_j^a, hereafter, are called category weight vectors),

$$w_j^{a-c} \equiv (w_{j1}^{a-c}, \cdots, w_{jM_a}^{a-c}) \qquad j = 1, \ldots N_a$$

which covers only the original dimension of the input space. These weight vectors are updated according to:

$$\left(w_J^{a-c}\right)^{\text{new}} = \left(w_J^{a-c}\right)^{\text{old}} + \frac{1}{\left|w_J^{ab}\right|}\left(a - \left(w_J^{a-c}\right)^{\text{old}}\right)$$

Prediction Phase:

(i) Feed forward the original input vector, $\boldsymbol{a}$, from F_0^a to F_1^a, and then to F_2^a together with w_J^{a-c}

(ii) Based on a heuristic of the nearest-neighbor of a distinct class, the width estimation for the i category is computed as

$$d_i = \min_{1 \le j \le N_a,\ \text{class}\ i \ne j} \left\| w_i^{a-c} - w_j^{a-c} \right\| \quad 1 \le i \le N_a$$

$$\sigma_i^2 = \frac{d_i^2}{r}$$

where r is an "overlapping parameter".

(iii) The kernel estimate for the i category prototype is computed using a Gaussian function as

$$\phi(\left\|a - w_i^{a-c}\right\|) = \frac{1}{(2\pi)^{M/2}\sigma_i^M} \exp\left(-\frac{(x - w_i^{a-c})^t (x - w_i^{a-c})}{2\sigma_i^2}\right)$$

(iv) Sum the kernel estimates from F_2^a, weighted by $|w^{ab}|$, to the corresponding categories in the map field to obtain the estimated class pdfs

$$p(x|C_k)=\frac{\sum_{i=1}^{N_a}\phi_i w_{ik}^{ab}}{\sum_{i=1}^{N_a} w_{ik}^{ab}} \qquad k=1,\cdots N_b$$

(v) Compute estimates of the prior probabilities, $p(C_k)$,

$$S_k=\sum_{j=1}^{N_a} w_{jk}^{ab} \qquad S_{Total}=\sum_{k=1}^{N_b} S_k$$

$$p(C_k)=\frac{S_k}{S_{Total}}$$

(vi) Select the highest posterior estimate or the minimum-risk estimate according to the Bayes' rule

$$p(C_k|x)=p(x|C_k)p(C_k)l(C_k|C_j)$$

where the class pdf, $p(x|C_k)$, is weighted by the prior probability, $p(C_k)$, and the risk or loss factor, $l(C_k|C_j)$ (the risk of choosing C_k when C_j is the actual class, $j\neq k$), and assuming that the probability that x occurs is unity.

Appendix B Decision Combination Algorithms

(a) The Bayesian Approach

In the following section, we present computation of the Bayesian combination procedure which is adapted from [3]. Given a data set containing N samples, the performance index of a classifier, e_k, where $k=1,\cdots,K$, is recorded in its confusion matrix as follows:

$$CM^k=\begin{pmatrix} n_{11}^k & n_{12}^k & \cdots & n_{1(M+1)}^k \\ n_{21}^k & n_{22}^k & \cdots & n_{2(M+1)}^k \\ \vdots & \vdots & \ddots & \vdots \\ n_{M1}^k & n_{M2}^k & \cdots & n_{M(M+1)}^k \end{pmatrix}$$

where $n_{ij}^{k}, i = 1,\cdots M, j = 1,\cdots, M+1$ indicates the number of samples belonging to C_i, but assigned to class *j* by e_k. The total number of samples is $N = \sum_{i=1}^{M}\sum_{j=1}^{M+1} n_{ij}^{k}$, in which the number of samples of C_i is $n_{i\bullet}^{k} = \sum_{j=1}^{M+1} n_{ij}^{k}$ (*i.e.*, summation through row *i*), and the number of samples that are assigned to label *j* is $n_{\bullet j}^{k} = \sum_{i=1}^{M} n_{ij}^{k}$ (*i.e.*, summation through column *j*). This confusion matrix provides information regarding a classifier's ability to identify accurately samples from a particular target class. In other words, the prediction by e_k that $\boldsymbol{x}$ belongs to C_i is associated with a factor of uncertainty that could be expressed as conditional probabilities of $\boldsymbol{x} \in C_i$, given that $e_k(\boldsymbol{x}) = j$, *i.e.*,

$$P\left(\boldsymbol{x} \in C_i | e_k(\boldsymbol{x}) = j\right) = \frac{n_{ij}^{k}}{n_{\bullet j}^{k}} = \frac{n_{ij}^{k}}{\sum_{i=1}^{M} n_{ij}^{k}}, \quad i = 1,\cdots M \tag{B1}$$

This technique is, in fact, utilized for evidence gathering and uncertainty reasoning using the Bayesian framework in artificial intelligence [30].

With *K* classifiers, the objective is to find a combined result, $E(\boldsymbol{x})$. In order to simplify the procedure for integrating the conditional probabilities together using Bayes' theorem, assume that the classification environment, *EN*, consists of *K* independent events, $e_k(\boldsymbol{x}) = j_k$, $k = 1,\cdots,K$, with *M* mutually exclusive and exhaustive sets of target outputs. Let $H(\boldsymbol{x})$ denote the overall hypothesis $e_1(\boldsymbol{x}) = j_1,\cdots,e_K(\boldsymbol{x}) = j_K$. The combined posterior probabilities under the common environment *EN* can then be expressed as

$$\begin{aligned} P(\boldsymbol{x} \in C_i | e_1(\boldsymbol{x}) = j_1,\cdots,e_K(\boldsymbol{x}) = j_K, EN) &= P(\boldsymbol{x} \in C_i | H(\boldsymbol{x}), EN) \\ &= \frac{P(H(\boldsymbol{x}) | \boldsymbol{x} \in C_i, EN) P(\boldsymbol{x} \in C_i | EN)}{P(H(\boldsymbol{x}) | EN)} \end{aligned} \tag{B2}$$

With the assumption that all the classifiers perform independently, the joint probabilities can be reduced to

$$\frac{P(H(\boldsymbol{x}) | \boldsymbol{x} \in C_i, EN)}{P(H(\boldsymbol{x}) | EN)} = \frac{\prod_{k=1}^{K} P(e_k(\boldsymbol{x}) = j_k | \boldsymbol{x} \in C_i, EN)}{\prod_{k=1}^{K} P(e_k(\boldsymbol{x}) = j_k | EN)} = \frac{\prod_{k=1}^{K} P(\boldsymbol{x} \in C_i | e_k(\boldsymbol{x}) = j_k)}{\prod_{k=1}^{K} P(\boldsymbol{x} \in C_i | EN)} \tag{B3}$$

Substituting equation (B3) into (B2), we have

$$P(x \in C_i | H(x), EN) = \frac{\prod_{k=1}^{K} P(x \in C_i | e_k(x) = j_k)}{\prod_{k=1}^{K} P(x \in C_i | EN)} P(x \in C_i | EN) \qquad \text{(B4)}$$

The computation of $P(x \in C_i | EN)$ requires the estimation of posterior probabilities of class C_i from each classifier. However, information on posterior probabilities is not available for decision combination at level 1. Thus, an estimate of equation (B4) has to be formulated. For practical implementation, [3] proposes to use the following equation to approximate equation (B4)

$$P(x \in C_i | H(x), EN) \approx \frac{\prod_{k=1}^{K} P(x \in C_i | e_k(x) = j_k)}{\sum_{i=1}^{M} \prod_{k=1}^{K} P(x \in C_i | e_k(x) = j_k)} \qquad \text{(B5)}$$

where each $P(x \in C_i | e_k(x) = j_k)$ is computed from the confusion matrix using equation (B1) by replacing j with j_k. Based on the combined probabilities, the one with the highest value is selected as the final outcome, *i.e.*,

$$E(x) = \begin{cases} j, & \text{if } P(x \in C_j | H(x),\ EN) = \max_{i \in \Lambda} P(x \in C_i | H(x), EN) \\ & \quad \text{and } P(x \in C_j | H(x),\ EN) \geq \lambda \\ M+1, & \text{otherwise} \end{cases} \qquad \text{(B6)}$$

where $0 \leq \lambda \leq 1$ is a threshold to regulate confidence associated with the final decision.

(b) The Behavior-Knowledge Space (BKS) Approach

One of the criticisms of the Bayesian approach is the assumption that all classifiers operate independently, which indeed may not always be true in real-world applications, in order to tackle the computation of the joint probabilities. To avoid using this assumption, [4] proposed a combination procedure which makes use of a so-called Behavior-Knowledge Space (BKS) that concurrently records the decisions of all classifiers on each learned sample.

A BKS is a K-dimensional space where each dimension corresponds to the decision of one of the K classifiers. In a BKS, there are $(M+1)^K$ units, where each unit accumulates the number of samples belonging to each C_i. An example is presented here in order to explain the procedure clearly. Suppose that two classifiers are used to categorize the input samples into M output classes. Then, a two-dimensional BKS can be formed as below,

e_1 e_2	1	2	$\cdots$	$M+1$
1	U_{11}	U_{12}	$\cdots$	$U_{1(M+1)}$
2	U_{21}	U_{22}	$\cdots$	$U_{2(M+1)}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$M+1$	$U_{(M+1)1}$	$U_{(M+1)2}$	$\cdots$	$U_{(M+1)(M+1)}$

Each BKS unit, U_{ij}, can further be divided into M elements, $n_1^H, \cdots, n_M^H$, where H denotes the overall hypothesis of all classifiers, $e_1, \cdots, e_K$. Each element indicates the number of samples for C_i. When an input sample, x, is presented, one of the BKS units will become active when it receives the decisions from all K classifiers, *e.g.*, U_{34} will be selected as the focal unit if $e_1(x)=3$ and $e_2(x)=4$. Then, the total number of samples in the focal unit is computed, and the best representative class (*i.e.*, the one which contains the highest number of samples) is identified.

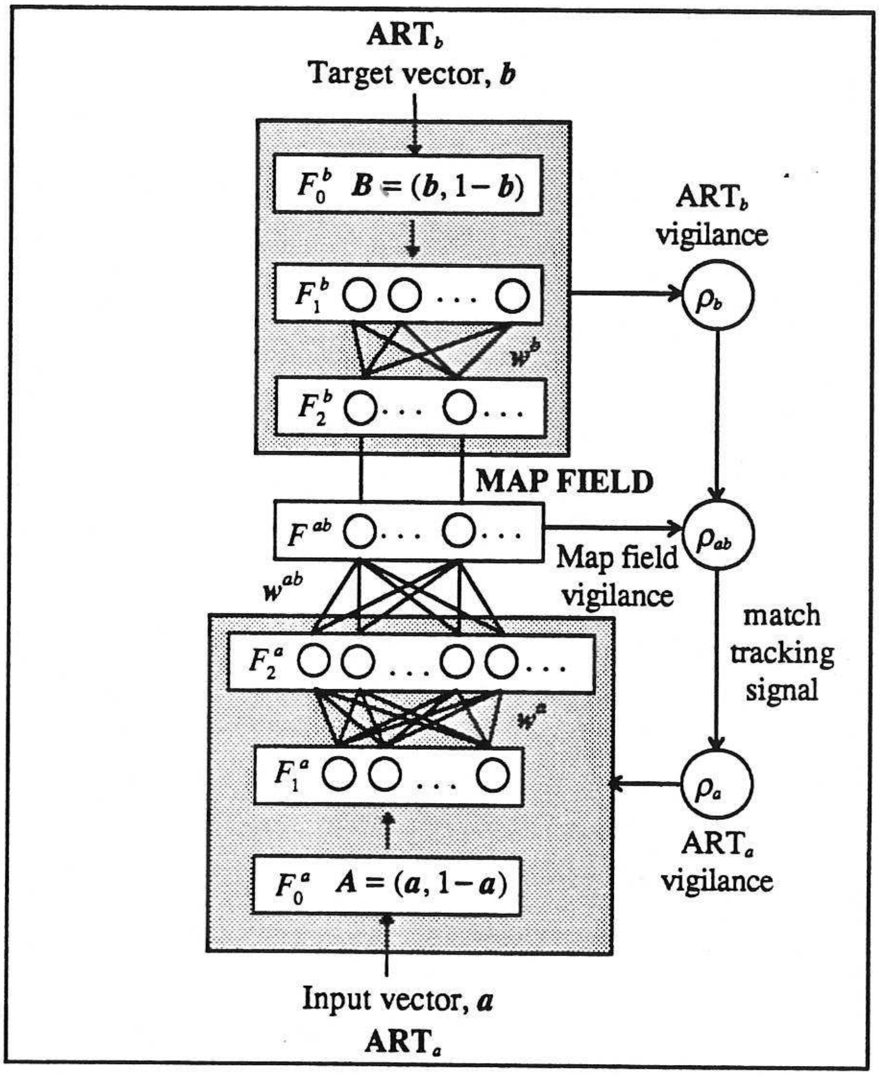
$$\text{Total number of samples} = T(H) = \sum_{i=1}^{M} n_i^H \tag{B7}$$

$$\text{Best representative class} = R(H) = j \text{ where } n_j^H = \max_{i \in \Lambda}(n_i^H) \tag{B8}$$

The decision rule that determines the final outcome is formulated as

$$E(x) = \begin{cases} R(H), & \text{if } T(H) > 0 \text{ and } \dfrac{n_{R(H)}^H}{T(H)} \geq \lambda \\ M+1, & \text{otherwise} \end{cases} \tag{B9}$$

where $0 \leq \lambda \leq 1$ is a user-defined confidence threshold.



ART_b
Target vector, $\boldsymbol{b}$
F_0^b $\boldsymbol{B} = (\boldsymbol{b}, 1-\boldsymbol{b})$
F_1^b
w^b
F_2^b
ART_b vigilance
ρ_b
MAP FIELD
F^{ab}
ρ_{ab}
Map field vigilance
w^{ab}
match tracking signal
F_2^a
w^a
F_1^a
ρ_a
ART_a vigilance
F_0^a $\boldsymbol{A} = (\boldsymbol{a}, 1-\boldsymbol{a})$
Input vector, $\boldsymbol{a}$
ART_a

Figure 1 A schematic diagram of the Fuzzy ARTMAP network

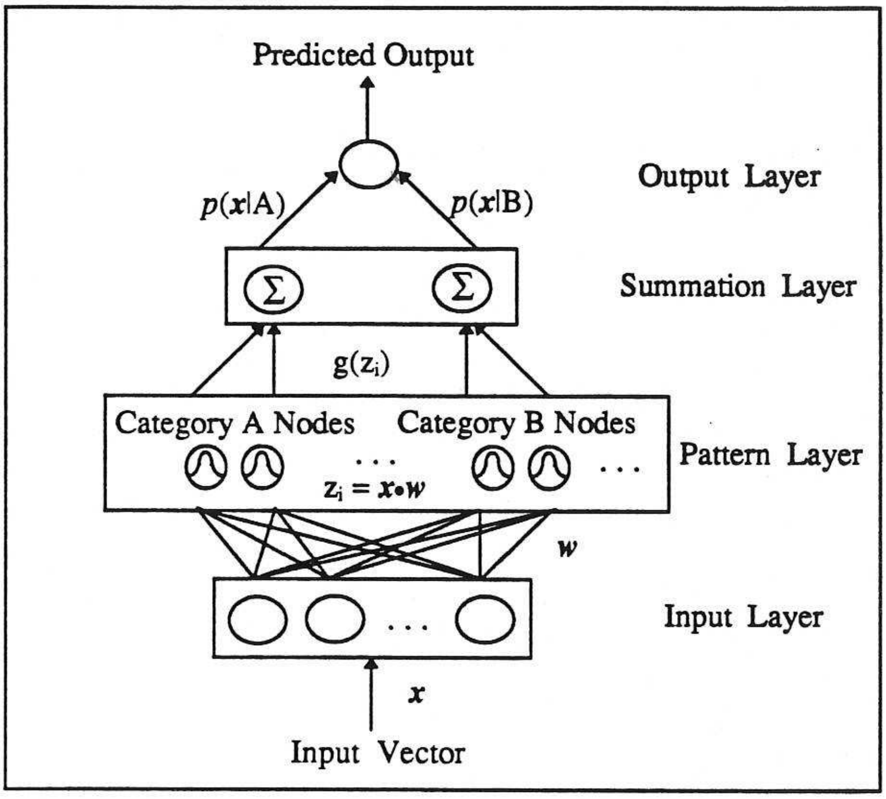


Figure 2 A schematic diagram of the Probabilistic Neural Network

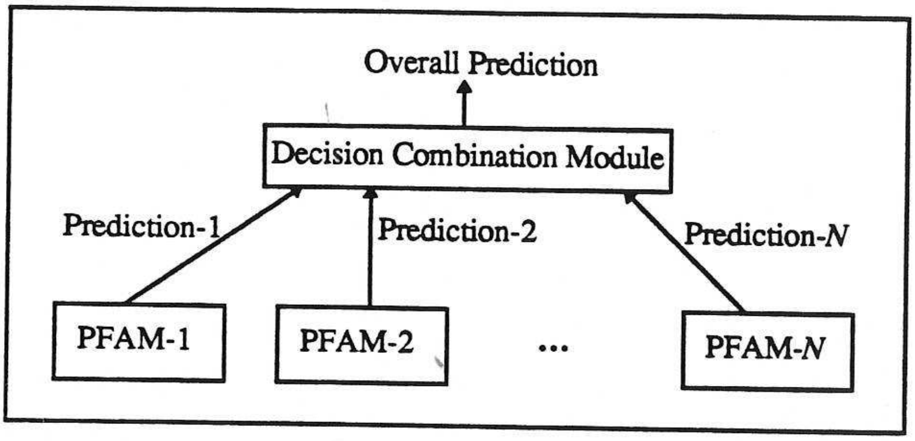


Figure 3 A schematic diagram of a multiple classifier system. There are *N* channels of independent, supervised PFAM classifiers. Predictions from these classifiers are combined with some decision combination schemes to give an overall prediction.

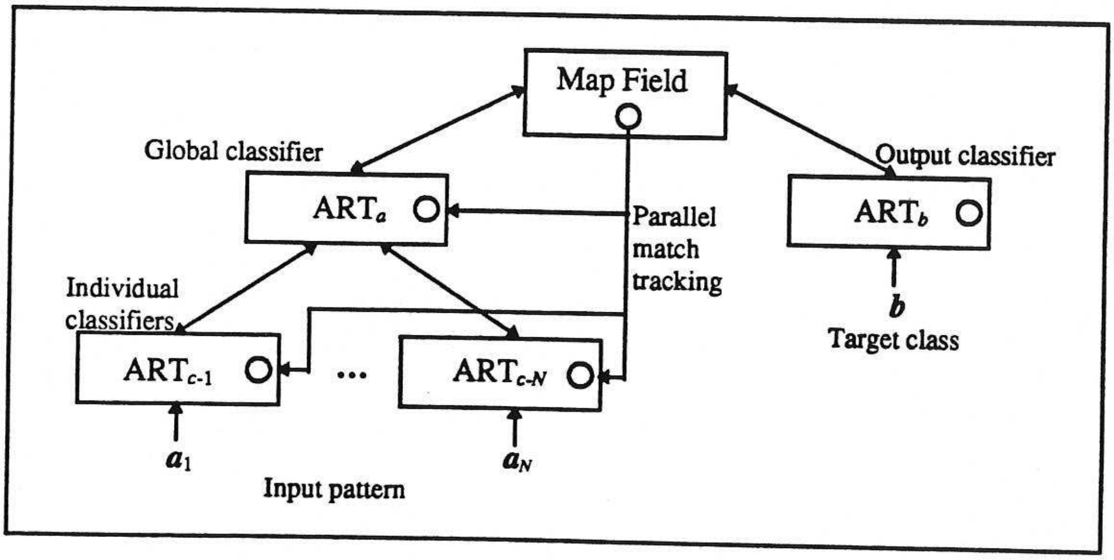


Figure 4 A schematic diagram of the Fusion ARTMAP architecture. It consists of *N* unsupervised Fuzzy ART classifiers (ART$_c$), one for a group of related input features. A global classifier, ART$_a$, is used to combine outputs from the individual classifiers, and a target output classifier, ART$_b$, is used to impose supervision for the system via the map field.

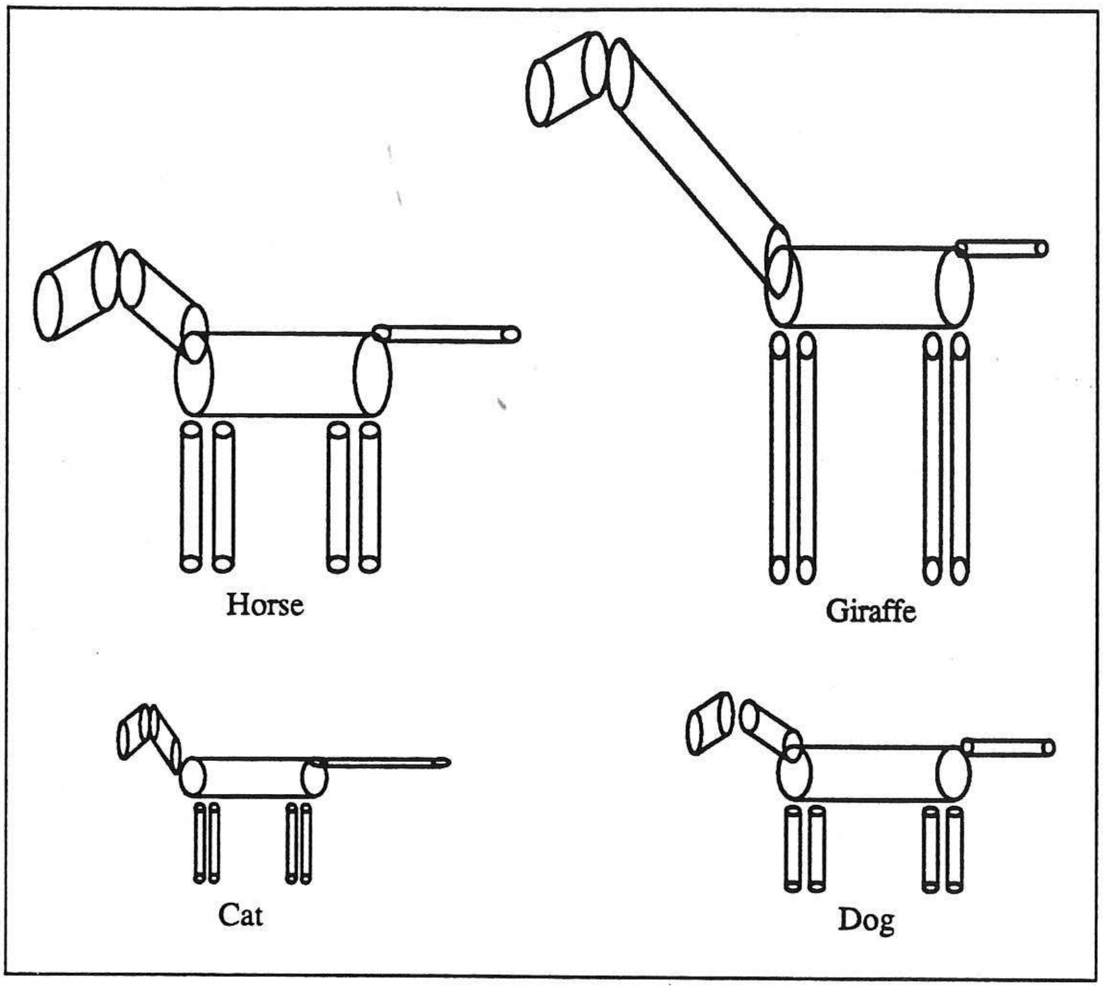


Figure 5 Four classes of quadruped mammals. Each mammal is modelled by eight cylindrical components: head, neck torso, tail, and four legs, and each cylinder is further described by nine attributes: length, height, texture, three axis locations, and three rotation angles.

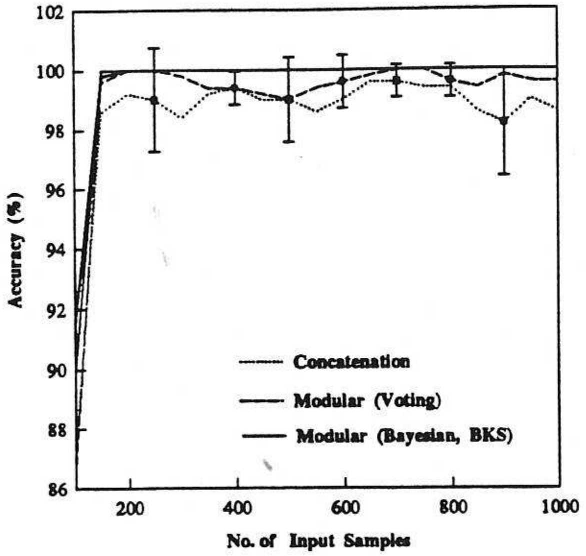


Figure 6 A comparison of the on-line results (average of 5 runs) between the individual classifiers with the concatenation approach and the multiple classifier systems with the modular approach. The error bars indicate the standard deviations of the 5 runs.

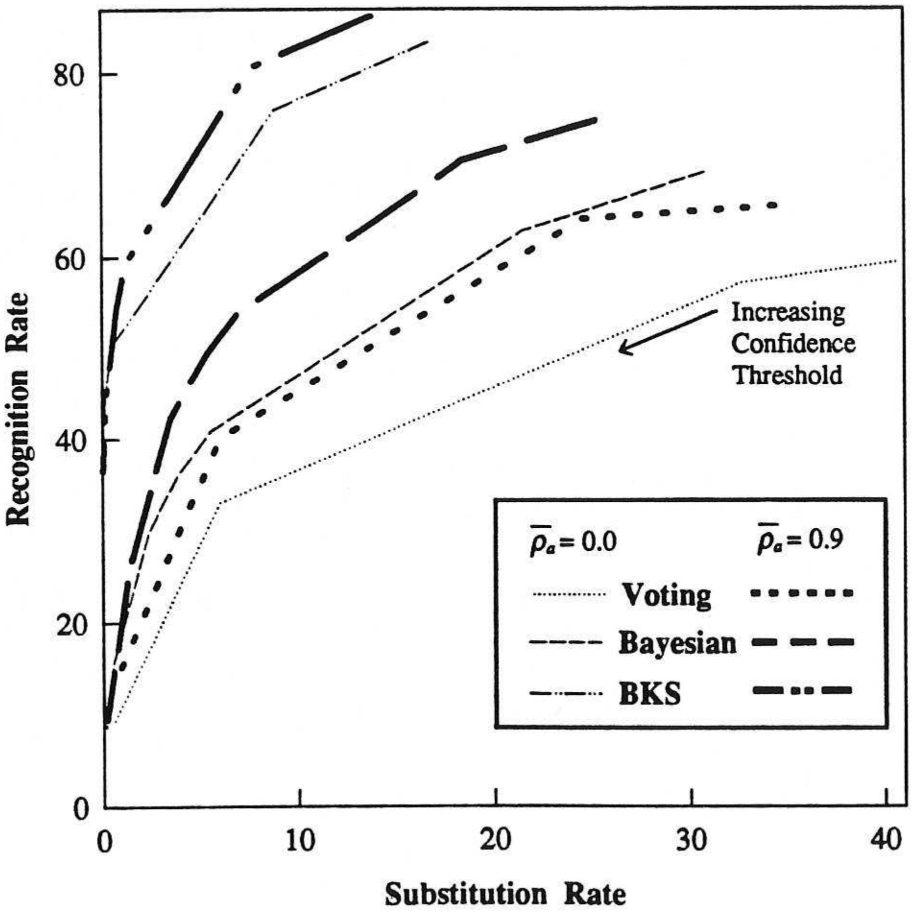
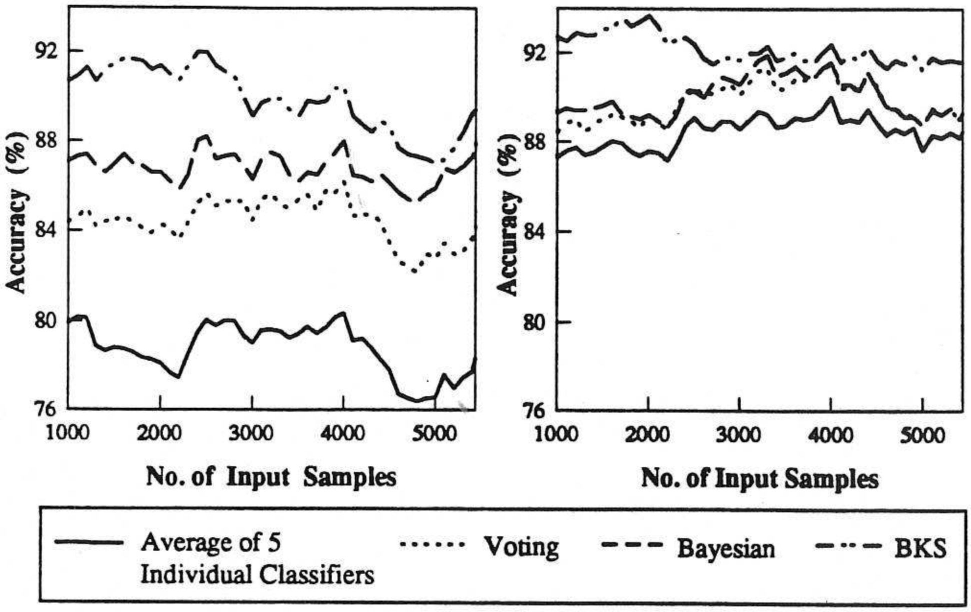


Figure 7 A plot of the substitution rate against the recognition rate for the multiple classifier systems using varying confidence thresholds.



(a) $\overline{\rho_a} = 0.0$

(b) $\overline{\rho_a} = 0.9$

Figure 8 The overall classification accuracy against increasing number of input samples for the average results of 5 independent classifiers as well as the voting, Bayesian and BKS multiple classifier systems