Monograph:
Chee, Peng Lim, Harrison, R.F. and Kennedy, R. Lee. (1996) Application of Autonomous Neural Networks Systems to Medical Pattern Classification Tasks. Research Report. ACSE Research Report 652 . Department of Automatic Control and Systems Engineering

Application of Autonomous Neural Network Systems to Medical Pattern Classification Tasks

Chee Peng Lim,[a] Robert F. Harrison,[a] and R. Lee Kennedy[b]

[a]Department of Automatic Control and Systems Engineering
The University of Sheffield

[b]Department of Medicine
The University of Sunderland

Research Report No. 652
November 1996

Abstract

This paper presents a study of the application of autonomously learning multiple neural network systems to medical pattern classification tasks. In our earlier work, a hybrid neural network architecture has been developed for on-line learning and probability estimation tasks. The network has been shown to be capable of asymptotically achieving the Bayes optimal classification rates, on-line, in a number of benchmark classification experiments. In the context of pattern classification, however, the concept of multiple classifier systems has been proposed to improve the performance of a single classifier. Thus, three decision combination algorithms have been implemented to produce a multiple neural network classifier system. Here the applicability of the system is assessed using patient records in two medical domains. The first task is the prognosis of patients admitted to coronary care units; whereas the second is the prediction of survival in trauma patients. The results are compared with those from logistic regression models, and implications of the system as a useful clinical diagnostic tool are discussed.

Keywords:

Multiple neural network classifiers, pattern classification, Bayesian decision, on-line learning, decision support systems, medical diagnosis and prognosis

1 Introduction and Motivation

Diagnostic problem solving in a medical domain is a prime example of decision making under uncertainty. Almost all medical prognostic and diagnostic problems are based on more than one item of data because it is highly exceptional for a single symptom or measurement to be pathologically significant to one disease without any associations with others [1, 30, 35]. Indeed, many different outcomes may correspond to identical sets of data or, conversely, distinct sets of data may point to the same disease. In addition, some data items may be noisy or imprecise making the whole diagnostic process probabilistic in nature. These factors can often result in an overall degradation of diagnostic accuracy by human beings. As a result, it would be potentially useful if automated "intelligent" systems could be developed to improve the decision making process in the medical consultation and diagnosis arena.

In general, a decision support system often integrates data-based analytical techniques, such as decision and classification theory, and knowledge-based approaches to improve decision making. Neural networks possess both the above properties, and have been investigated for their suitability as inference engines in many medical decision support systems, for example, the diagnosis of epilepsy [2], low back disorders [3], early diagnosis of heart attack [4], and the diagnosis of breast cancer [5]. In a neural network, learning proceeds by feeding the network with data items and verified outcomes (supervised learning) for it to form complex associations directly from domain exemplars. This learning methodology therefore avoids the usual painstaking and time-consuming knowledge acquisition process required for the formation of rules in expert systems. Another attractive property of neural networks is that they are self-adaptive, and can generalise well to cover unseen patterns.

The main thrust of our work is to devise appropriate methods and strategies for the development of useful, usable and valid clinical decision aids using neural networks and pattern recognition techniques. A useful decision support system must also be robust, easy to use, and portable. Hence, on-line or causal learning is a desirable characteristic of such systems, so that they can be adapted to function under varying operational conditions, *e.g.* differing geographical or demographical conditions; changing clinical practices from site to site; or improvements in diagnostic procedures owing to advances in medical knowledge and technology. The goal is to enable the system to learn autonomously and continually with minimum intervention, and

ultimately to allow domain experts, rather than computer systems engineers, to develop and fine-tune their own "intelligent" decision support tools. We concentrate on this aspect in this paper.

Feedforward networks, such as the Multi-Layered Perceptron (MLP) network [6] and the Radial Basis Function (RBF) network [7, 8] have been extensively investigated and used in many applications. Research indicates that these networks possess rich enough architectures to approximate, theoretically, any (smooth enough) function to an arbitrary degree of accuracy [9, 10]. However, there are several practical issues associated with the functionality of "standard" feedforward networks. In general, it is usually not known what finite *size* of network, *i.e.* number of hidden nodes, should be used to solve a given problem [11]. If the network is small, it may not be capable of forming an accurate approximation to the underlying function. On the other hand, if the network is large, the network may be too specialised to the training samples such that the solution obtained is likely to be a poor estimate of the underlying function.

Another problem associated with feedforward networks is that they are *static* after training. A feedforward network is generally trained off-line with a set of data pairs. Once the training cycle is completed, learning is suppressed and the network is put into operation. Such an approach is viable when there is good reason to believe that the data environment is stationary, and the data samples used in training are sufficiently representative. However, should operational conditions change over time, or should one wish to use a network developed in one location in another where demography is different, the network cannot tune-in to the new environment. Re-training is then necessary—a process which is expensive and time-consuming. The alternative to this is, of course, to allow the network to continue to train, *in situ*, whilst being used to support decision making. For the most commonly used neural networks, the feedforward networks exemplified earlier by the MLP and RBF networks, there are two objections to this. The first is purely technical and arises from the notion of "learning" in such systems. This takes the form of an optimisation procedure which is in general non-convex and thus may possess more than one solution. How then, can an autonomous system know if it has reached an adequate solution? Furthermore, in the face of temporal or geographical changes, a network structure which was once adequate may no longer be so. One way out of this problem is to determine the size of the network on-line in which the network is allowed to grow sequentially by itself in order to find a suitable structure for the task at hand in an incremental manner. In fact, this is the main principle of several "growing" and/or "shrinking" network architectures proposed in the literature [12, 13,

14]. The second objection is more fundamental and stems from the fact that feedforward networks are indiscriminate in what they learn—all data are treated as being equally valid and no mechanism is present to determine whether the current input is novel but valid, or is spurious (noise).

The second objection above underlies the so-called "stability-plasticity" dilemma [15, 16]. This poses the questions: how can a learning system remain plastic (or adaptive) in response to significant events and yet remain stable in response to irrelevant events? How can a system adapt to new information without corrupting or forgetting previously learned information [16]? In response to this dilemma, Carpenter, Grossberg and colleagues have developed a family of neural networks called Adaptive Resonance Theory (ART) networks.

ART has an incremental learning architecture which self-organises and self-stabilises in response to an arbitrary sequence of sample patterns in stationary and non-stationary environments [15]. The key feature of the ART network is the design of a feedback mechanism, in addition to the feedforward structure, where a novelty detector is used to measure, against a threshold, the similarity between the prototype patterns stored in the network and the current input pattern. If the similarity criterion is not satisfied, a new node can be recruited to encode the input as a prototype pattern in the network. As a result, and subject to the novelty criterion being satisfied, the number of nodes (or hidden units) grows with time in an attempt to learn incrementally an adequate network structure. As different tasks demand different network capabilities, this learning methodology is able to avoid the need to specify a pre-defined static network size, or to re-train the network in non-stationary environments. Such capabilities have clear potential in decision support.

Recent developments have produced a family of mapping networks called ARTMAP [17, 18] which is capable of supervised learning whilst retaining the desirable properties of the earlier ART networks. It seems that the ARTMAP networks offer greater potential for developing an autonomously learning system which is capable of classifying data according to different outcomes in a stable way, and at the same time, of protecting the system from memory erosion as a result of spurious information. In terms of decision support, this capability means that the system can continue safely to learn *in situ* whilst providing useful predictions, *i.e.* analogously to the behaviour of an apprentice.

This paper presents an investigation of autonomously learning multiple neural network classifier systems for clinical decision support. In section 2, the multiple neural network systems are described. First, a hybrid neural network resulting from the integration of two neural network models is presented. Then, three decision combination algorithms for aggregating the predictions from individual classifiers are introduced. In section 3, two case-studies involving data gathered from a number of teaching hospitals in the United Kingdom are employed to evaluate the applicability of the multiple neural network systems as medical decision support tools. The results are compared to those obtained from logistic regression models. Some concluding remarks and suggestions for further work are provided in section 4.

2 The Multiple Neural Network Systems

2.1 Fuzzy ARTMAP

Fuzzy ARTMAP (FAM) [18] is a neural network architecture capable of performing incremental supervised learning of recognition categories and multi-dimensional maps in response to arbitrary sequences of binary, analogue, or fuzzy input patterns. Figure 1 depicts a schematic diagram of the FAM network. It consists of two identical fuzzy ART [19] modules, ART_a and ART_b, inter-linked by a map field, F^{ab}. The ART_a (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 the number of nodes (hidden units) can be increased when necessary, and each node encodes a prototype pattern for 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 so that the size of the input vector maintains a constant norm in order to avoid the category proliferation problem [17, 19].

During supervised learning, ART_a receives a stream of input pattern vectors, $\{\boldsymbol{A}\}$, whereas ART_b receives the corresponding target-class vectors, $\{\boldsymbol{B}\}$. In general, ART_b consists of an independent Fuzzy ART module to self-organise the target vectors. 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.

The key feature of FAM is that a novelty detector is used to measure, against a threshold, the similarity between the prototype patterns stored in the network and the current input

(vigilance test). If the vigilance test is satisfied, then the input is accepted as a member of the target class. If the test fails, a new search cycle will be triggered to seek a better prototype in F_2^a. However, if all the F_2^a prototypes fail the vigilance test, a new node is recruited with the input encoded as its prototype pattern. As a result, the number of nodes grows with time, governed by the novelty criterion, in an attempt to learn a good network configuration autonomously and on-line to suit the problem at hand.

The role of the map field is to impose supervision on the network. It determines the predictions from prototype nodes in ART_a to target classes in ART_b. After a winner has been selected in F_2^a, a target class is primed in ART_b. In response to an incorrect prediction, the F_2^a winner is inhibited, and the search cycle continues in ART_a until a correct prediction, possibly from a newly recruited node, occurs. An association between the F_2^a winning prototype and the ART_b target class is then established. This association is permanent so that a target output can be recalled when an input is presented to ART_a.

2.2 The Probabilistic Neural Network

The Probabilistic Neural Network (PNN) [20] is a neural network model that implements a non-parametric density estimation procedure for classification, mapping or associative memory. It learns instantaneously in one-pass through the data samples, and is able to formulate complex decision boundaries which approximate the theoretically achievable (Bayes) limits of performance. The decision boundaries can be modified on-line when new data is available without having to re-train the network.

The key feature of the PNN is its ability to estimate the probability density functions directly from training samples by using the Parzen-windows technique [21]. Figure 2 depicts a schematic diagram of a four-layer PNN architecture for binary classification tasks. During operation, the input pattern, $\boldsymbol{x}$, is first fanned-out to the pattern layer where each pattern unit forms a dot-product of the input and weight patterns. The dot-product is then transformed by an activation function in accordance with the Parzen kernel estimators. The summation nodes sum all the pattern node outputs corresponding to each class to give an estimate of the resulting probability density functions, $p(\boldsymbol{x}|A)$ and $p(\boldsymbol{x}|B)$. For classification problems, these estimates can be weighted by their respective prior probabilities, $P(A)$ and $P(B)$ respectively. This enables the

output unit to calculate the posterior probability of x belonging to a particular category according to the Bayesian decision criterion, *i.e.* $P(A|x) = p(x|A)P(A) / p(x)$. There are several forms of kernel functions for implementation with the PNN structure, *e.g.* Gaussian kernels, city-block kernels, and Euclidean kernels [20, 22].

Learning in the PNN is accomplished by generating a pattern node, connecting it to the summation node of the target category, and assigning the input vector as the weight vector. This process is non-iterative and can be implemented on-line. However, this learning approach means that the number of pattern nodes required is the same as the number of data samples. While this is not an issue for small data sets, it will result in an explosive number of pattern nodes if large or unbounded sample sets are employed. Besides, the storage requirement and computational burden increase correspondingly with the size of the data sets.

2.3 Probabilistic Fuzzy ARTMAP

Given an input pattern, FAM is able to make a deterministic prediction of the target class. However, many pattern recognition problems require an estimate of the probability that an input belongs to a given class. In addition, overlapping regions can frequently occur in the input space, especially in statistical classification tasks, in which a particular cluster may belong to more than one class, subject to different probabilities of class membership. Thus, when an input is presented, it would be beneficial if FAM could produce a probability estimate for different target classes.

As stated earlier, a drawback of the PNN is the need to create a new pattern node for each input vector. This problem can be alleviated by using a clustering technique to reduce the number of pattern nodes required [22, 23]. The idea is to find a set of reference vectors, or prototypes, to represent sets of samples clustered in the input space. Instead of using all the input samples, these prototypes are used to locate the kernel functions. Consequently, the storage and computational burdens can be lightened while the ability to estimate probability density functions is maintained.

It seems that both the PNN and FAM networks complement each other in mitigating one another's shortcomings. Indeed, our studies have found that there is a close similarity in the network topology between FAM and the PNN. Figure 3 shows the corresponding network

structure between FAM and the PNN. The F_1^a and F_2^a layers correspond to the input and pattern layers of the PNN whereas the map field layer (F^{ab}) corresponds to the PNN's 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 ART_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.

Therefore, a novel hybrid network, based on the integration of a modified version of FAM [24] and the PNN, has been proposed for on-line classification and probability estimation tasks, and is referred to as Probabilistic Fuzzy ARTMAP (PFAM) [25]. The on-line PFAM algorithm is divided into two phases. First, the FAM clustering procedure is adopted for classifying the input patterns into different categories (learning phase). Subsequently, the PNN probability estimation procedure is used to select the most probable target output (prediction phase). 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 above descriptions provide a conceptual structure for incorporating FAM and the PNN into a unified framework, and the rationale behind their integration. In practice, several modifications are necessary to allow effective combination of these networks, and to increase the generalisation ability of the resulting system. A summary of the PFAM algorithm is given in the Appendix A.

The PFAM network has been shown to be capable of asymptotically attaining the theoretically achievable (Bayes optimal) classification rates, on-line, in a number of benchmark classification problems [25, 26]. In the present work, however, a multiple classifier system consisting of several modules of PFAM is implemented for clinical decision support.

2.4 Multiple Classifier Systems

In the context of pattern recognition, there exists a variety of classification algorithms and methodologies, *e.g.* distance-based classifiers, syntactical classifiers, and neural network classifiers. It is often very difficult to construct a classifier which is able to utilise all the

discriminatory power of input features in a data set. For a specific problem, different algorithmic classifiers may attain different degrees of accuracy. Multiple classifier systems offer a way to tackle difficult classification tasks involving various types of input features (*e.g.* binary, continuous, syntactic), noisy inputs, missing data, or a large number of target classes [27, 28, 29]. Clearly, procedures for combining multiple classifiers should take advantage of the strengths of the individual classifiers, avoid their weaknesses, and improve overall classification accuracy.

As pointed out by Carpenter *et al* [18], the formation of ART_a cluster prototypes in the input space is affected by the sequence of input sample presentations. This would subsequently lead to different predictions of target classes, and thus different accuracy scores for each realisation of FAM (as well as PFAM since the learning methodology of PFAM is based upon FAM). The data ordering effect is further exacerbated if the prototypes are to be established autonomously, on-line, because in this case, the input samples are presented only once, and in a fixed order. One way to mitigate this problem is to train a pool of networks off-line, each with a different order of input samples. In operation, the results from these networks can then be combined to give an overall prediction. Thus, not only is the performance enhanced, but the confidence associated with the final prediction can be assessed.

Figure 4 depicts a schematic diagram of a PFAM-based MCS. Three methods for combining decisions from multiple classifiers have been implemented. The first one is a simple majority voting strategy where the predictions across an ensemble of individual classifiers are evaluated, and the target class with the highest number of votes is declared as the final prediction. However, each classifier is treated equally as having one vote without considering its past predictive errors. A more reasonable approach is to take the predictive accuracy of each classifier into consideration. Highly accurate classifiers' results should be given more weight than less accurate ones. This is the rationale behind the use of the Bayesian approach to combining decisions from multiple classifiers [27]. Nevertheless, one of the criticisms of the Bayesian approach is the assumption that all classifiers must operate independently which may not always be true in real-world applications. This is required to make the computation of the joint probabilities tractable. To avoid using this assumption, Huang and Suen [28] have proposed a combination procedure which makes use of a so-called Behaviour-Knowledge Space (BKS) that concurrently records the decisions of all classifiers on each learned sample. A review of the Bayesian and BKS approaches is presented in Appendix B.

3 Medical Applications

Recent years have seen increasing use of neural networks for solving problems in the medical domain. Some neural network models are able to combine statistical inference techniques with the machine learning objective of imitating human intelligence. The PFAM algorithm, in essence, implements a classical non-parametric estimation procedure, and is able to produce a decision within the framework of Bayes' theorem. In this work, we attempt to devise practical strategies which allow the system to function as a usable and useful decision support tool. Two databases involving real patient records were employed to demonstrate the applicability of the individual and multiple PFAM networks to medical diagnosis and prognosis problems. For each database, three different learning strategies were investigated and the results were compared to those from logistic regression analysis. To quantify the results, three performance indices, which are commonly used in medical diagnostic systems, are calculated, *i.e.*:

- Accuracy (ACC)—the ratio of the number of correct diagnoses to the total number of cases;
- Sensitivity (SENS)—the ratio of the number of correct positive diagnoses to the total number of patients having a positive outcome;
- Specificity (SPEC)—the ratio of the number of correct negative diagnoses to the total number of patients having a negative outcome.

In all experiments described below, the PFAM system was operated using its basic settings, where the important parameters were: the vigilance parameter of ART_a, $\rho_a = 0.0$ (forced choice); the learning rate parameter, $\beta_a = 1.0$ (fast learning); the choice parameter, $\alpha_a \approx 0.0$ (conservative mode); the overlapping parameter, $r = 1.0$ (see Appendix A); and the decision threshold in the MCSs, $\lambda = 0.0$ (Appendix B).

3.1 Experiments using the Database of Coronary Care Unit (CCU) Patients

Since the 1950s, there has been a progressive reduction in the recommended length of hospital stay for coronary care patients, which in turn has provided economic benefits without significantly increasing mortality rates [31]. However, early hospital discharge requires exact identification of patients who are subject to minimal risk of death. Thus, an accurate system for the prognosis of coronary care patients would not only help to improve the health care of these patients, but may also assist with the planning and management of hospital facilities.

The application highlighted here is a two-category classification problem: either C_1 (patient is at risk of serious complications, or of high Creatine Kinase (CK) assay $(>1000\,\mathrm{U}/1)$ indicative of myocardial infarction, or of heart attack, or death) or C_2 (none of the above risks). The study included a total number of 3721 consecutive admissions to the Leicester Royal Infirmary, United Kingdom, between August 1988 and August 1992. Information on symptoms, risk factors, therapy and complications were recorded on a proforma during the patient's admission. The final decision was assigned independently by senior clinicians involved in the patient's care.

3.1.1 Off-line Learning

In the off-line learning experiments, the data were divided into a training set of 1000 samples and a test set of 2721 samples. In an earlier study with this data set [32], logistic regression models were developed, and performance was assessed using the ROC curve—a useful tool for evaluating the performance of diagnostic systems that has been widely used in medical experiments [33, 34, 35, 36]. The study in [32] revealed 9 out of a total of the 43 clinical and electrocardiographic data items to be most significant in the patient's prognosis. These 9 items were abstracted from the data and used here.

A confirmatory logistic regression experiment was first conducted using the constant and coefficients reported in [32]. Figure 5(a) shows the ROC curve of the logistic regression model for the test data. The graph shows sensitivity (probability of detection) against 1-specificity (probability of false alarm), parameterised by diagnostic threshold. It is usual to set the threshold at 50%, *i.e.* C_1 is predicted if the posterior probability of C_1 is greater than that of C_2. However, examination of the ROC curve allows the operator to select a threshold which gives appropriate values of sensitivity and specificity. In particular, the choice of threshold at which the ROC curve intersects the leading semi-diagnonal is said to be optimal—at that point sensitivity = specificity (and hence accuracy), leading to a single performance indicator for the system. Another important performance indicator is the area under the ROC curve. This index corresponds to the probability of correct identification between "noise" and "signal-plus-noise" in a two-alternative, forced-choice test in signal detection [37]. The area under the ROC curve thus provides a measure of performance, independent of any threshold. The area ranges from a lower limit of 0.5 for an ROC curve lying along the major diagonal, defining performance at a 50-50 chance level,

to an upper limit of 1.0 for an ROC curve following the left-hand and top axes of the plot, defining perfect performance. Here, we use the trapezoidal rule to calculate the area under the ROC curve, and the method described by Hanley and McNeil [33, 34] for its corresponding standard error. These logistic regression results, presented in Table 1, serve as a baseline comparison for the performance of PFAM.

In the off-line experiments, an ensemble of 25 PFAM networks was tested. Results averaged across the ensemble should serve as a more stable performance indices, and confidence intervals can be estimated based on the ensemble results. Training was terminated after 10 epochs as there were no further new prototypes being created, *i.e.* category formation had stabilised. In addition to the average results, the results were combined using the multiple classifier approach described in section 2.4. A total of 12 MCSs was formed, each with 5 networks selected at random from the 25 individual networks, and with at least 4 different individual networks in each case. The individual networks were operated at their optimum threshold. The predicted outcomes from the individual networks on each sample in the test set were recorded for the construction of the confusion matrices and focal units in the Bayesian and BKS algorithms (see Appendix B).

		ACC	SENS	SPEC	ROC:Area ± S.E.
Logistic regression		73.0	73.1	72.9	79.0 ± 1.0
Averages of 25 individual networks		69.3 (64.2-72.5)	69.2 (64.6-73.0)	69.3 (63.4-73.3)	74.8 ± 1.0 (71.8-79.3, 0.9-1.1)
Averages of 12 MCSs	Voting	71.5 (70.2-73.6)	71.1 (68.4-78.3)	72.2 (65.1-74.6)	
	Bayesian	72.0 (69.2-74.1)	86.4 (82.2-91.7)	45.6 (31.8-52.0)	
	BKS	74.1 (73.3-74.6)	79.3 (74.6-84.4)	64.4 (55.2-71.0)	

Table 1 Off-line learning results for the database of CCU patients using a logistic regression model, as well as the individual and multiple PFAM systems operating at a near optimal threshold. Figures in parentheses indicate the maximum and minimum results.

Figure 5 depicts the ROC plots of logistic regression, as well as the minimum and maximum results achieved by PFAM. All the off-line experiment results are summarised in Table 1. Although a single run from PFAM could achieve the results equivalent to that of logistic regression, on average, the PFAM results (accuracy, sensitivity, specificity as well as the area under the ROC curve) were about 4% inferior compared to those of the logistic regression. The

difference of the areas under the two ROC curves (logistic regression and average PFAM results) was determined using the method described in [34], and it was found that the difference is significant (p=0.95). Indeed, there is evidence (from unpublished work) suggesting that the logistic regression model is operating near to optimality for this CCU patient data set because lengthy experimentation using Multi-layer Perceptron and Radial Basis Function networks has failed to attain a significantly better result.

From Table 1, notice that although all the individual networks were set at their optimum thresholds, the outcomes of MCSs (Bayesian and BKS) could not maintain a close level of performance for accuracy, sensitivity, and specificity. This observation is exceptional for the voting strategy where the three indices were still close to each other. The same observation had been encountered in another medical data set (the early diagnosis of heart attack) using FAM with voting [38]. The loss of levelling effect in MCSs is understandable as the combination algorithms only attempt to optimise the overall accuracy, *e.g.* by the use of confusion matrices and focal units, respectively, in the Bayesian and BKS approaches. Also, the ROC curve can no longer be constructed in MCSs because the combination is conducted at the very top level of predicted classes, where no probabilistic information is associated with each outcome. As a result, a better variable of comparison between MCSs and the individual networks is the accuracy index. In this case, the BKS approach achieved the best performance, followed by the Bayesian formalism and then voting. Nevertheless, the encouraging sign is that all three MCSs are able to improve the average results of individual PFAM classifiers. This observation demonstrates the benefit of using multiple classifiers in classification problems.

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 vector is first presented to ART_a with its target vector to ART_b. A prediction is then sent from ART_a to ART_b according to the prediction phase algorithm. The prediction is compared with the actual class and the outcome produces a classification result. Next, learning ensues to associate the input vector with its target class according to the learning phase algorithm.

Since learning takes place incrementally during the on-line learning mode, to reflect accumulation of knowledge through past experiences and to assess whether the performance improves through time, a 1000-sample moving window is applied to calculate the on-line performance statistics. For example, the accuracy at sample 2000 is the percentage of correct predictions from trials 1001-2000. A total of 12 PFAM networks was tested with all 3721 samples, each with random data ordering. Since each network is operated on random data ordering, it is impossible to use the MCS algorithms to give a combined, overall outcome at each sample point. In addition, the ROC analysis is inapplicable in on-line learning as it requires an off-line process to manipulate the threshold values. Nevertheless, in order to reduce the differences of the three performance indices, the operating threshold was updated on-line at each sample presentation as follows:

$$Threshold(n+1) = Threshold(n) + \eta(SENS - SPEC) \qquad (1)$$

with $\eta = 0.001$. The threshold was left unchanged if $|SENS - SPEC| \leq 5\%$. This variable threshold rule attempts to make the difference between SENS and SPEC small, thus, keeping accuracy, sensitivity and specificity close to each other.

The average results of 12 runs is shown in Figure 6, together with the growth pattern of the ART_a categories. The error bars indicate the 95% confidence intervals estimated using the *t*-distribution for small sample instances [30]. The use of the *t*-distribution method depends on the assumption that the data samples follow a Normal (Gaussian) distribution, which can be validated using a Normal plot [30]. A total of 100 randomly selected samples were examined using the Normal plot option in the *MINITAB* statistical package [39]. An average of 98.1% for the normal probability test was achieved, thereby confirming the validity of the confidence intervals estimated using the *t*-distribution.

A remark about on-line learning is that although the problem under scrutiny is stationary in nature, the on-line learning and prediction procedures form a non-stationary process owing to the build-up of templates, especially in the early stages of operation, which is known as the finite-operating-time problem. From Figure 6, it is clear that the performance indices were driven level to each other throughout the experiments by using equation (1). The moving average results at the end of 3721 samples are 68.1% accuracy, 67.9% sensitivity, and 68.6% specificity. Although these results are inferior to those from the off-line experiments, the advantage is that the learning

process is on-going and the system is able to learn autonomously without having to retrain the network upon arrival of new data.

3.1.3 Dual-mode Learning

In the on-line learning mode, the combination of decisions across an ensemble of networks is inappropriate as data ordering is different in each individual network. In practice, however, there is no reason why such a scheme could not be employed on-line, after an initial period of training. As a result, we propose a dual-mode learning strategy where an off-line learning process is used to equip the network with a knowledge base before on-line learning is initiated for the problem at hand. The fact that autonomy is lost in the early stages is not a disadvantage because, in any real application, a series of trials on historical data would be required before the system would be allowed to become operational. In off-line learning, the networks would establish different category prototypes, thus on-line predictions would be different. The combination strategies, once again, can be implemented to integrate outcomes from various networks for subsequent incoming samples during on-line learning.

In this experiment, 25 individual networks were trained initially with 1000 samples in random order, and then learning continued on-line for the remaining 2721 samples. Again, 12 MCSs, each with 5 networks, was constructed as described in section 3.1. Performance statistics were calculated with a 1000-sample moving window in the on-line learning mode, as in section 3.1.2. Figure 7 shows the overall average performance of the individual networks and MCSs. The error bars indicate the 95% confidence intervals of the average results (again using the *t*-distribution). Normal plots were used to check normality of the samples, and, again, an average of over 97.3% for the normality test results was obtained for 100 randomly selected samples. In general, performance improves when compared to the off-line and on-line learning results. The moving average results at the end of the experiments are summarised in Table 2.

It can be noticed from Figure 7 that the Bayesian strategy exhibited the worst levelling effect among accuracy, sensitivity, and specificity, followed by the BKS strategy, and voting still could maintain the three performance indices at a close level. As pointed out in the off-line learning experiments, a better comparison for the MCSs would be the accuracy index as the decision combination algorithms only attempt to optimise the overall classification accuracy. Figure 8 plots the average accuracy of 25 individual networks and the three MCSs along with

their 95% confidence intervals. The multiple classifier results were significantly better than those from individual classifiers. It can be seen that the BKS approach exhibits, statistically, the best performance, followed by the Bayesian formalism, and then voting. These results, once again, empirically demonstrate that multiple classifiers are able to outperform single classifiers.

		ACC	SENS	SPEC
Averages of 25 individual networks		70.5 (62.0-73.6)	70.8 (61.4-75.4)	70.0 (63.2-73.9)
Averages of 12 MCSs	Voting	73.4 (72.4-74.8)	75.1 (71.7-77.4)	72.3 (65.7-73.9)
	Bayesian	74.9 (73.1-76.1)	88.4 (85.5-92.1)	50.3 (38.2-57.5)
	BKS	76.4 (74.8-77.2)	82.5 (80.5-85.3)	65.1 (60.3-69.1)

Table 2 Dual-mode learning results at the end of the experiments for the database of CCU patients. Figures in parentheses indicate the maximum and minimum results.

3.2 Experiments using the Database of Trauma Patients

There is a widely held view that the standard of trauma patient care in the United Kingdom is sub-optimal, which is supported by evidence from preliminary reports of the UK Major Trauma Outcome Study (MTOS) [40] and the Scottish Trauma Audit Group (STAG) [41]. To improve the accuracy in diagnosis, a number of decision aids are available for trauma patients, and currently the most widely used statistical approach for auditing trauma care is the TRISS method [42]. In this study, we have a database of 6321 trauma patient records. The data were collected between February 1993 and February 1995 from six Scottish teaching hospitals—Aberdeen Royal Infirmary, the Royal Infirmary of Edinburgh, Glasgow Royal Infirmary, the Western Infirmary of Glasgow, the Victoria Infirmary of Glasgow, and the Southern General Hospital of Glasgow. Information was derived from all patients admitted for more than three days, those who died in hospital, and those admitted to an intensive therapy area.

Boyd *et al.* [42] have described the TRISS methodology for combining anatomical and physiological information in the administration of trauma diagnosis. The information includes Injury Severity Scoring (ISS), Revised Trauma Score (RTS), patient's age, and two types of injury (blunt and penetrating). Use of this method for calculating probability of survival for patients in the STAG database has been reported in [41].

The aim of this study is to examine the performance of single and multiple PFAM networks in categorising trauma patients into two classes— survival (C_1) or death (C_2). One of the difficulties encountered with this data set is that the data distribution is highly skewed with only about 5% instances of C_2. This deficiency of exemplars of one class compared with the other means that a classifier is likely to bias prediction towards the commonly seen class (C_1). Since approximately 95% of patients are in class C_1, the baseline performance of a classifier would be a 95% accuracy, *i.e.* by categorising all patients as C_1 (maximum *a priori* classification).

Three experiments comprising the three different learning strategies presented in section 3.1 were carried out. The accuracy, sensitivity, and specificity indices for identifying survivals were calculated in all experiments. The specificity index tended to fluctuate very greatly, which can be seen in the on-line plots presented later, owing to the small number of non-survival instances. It was very difficult to manipulate the operating threshold in the on-line learning mode to maintain a close performance among accuracy, sensitivity, and specificity. As a result, we decided to operate the PFAM system at the basic operating threshold of 50% in the experiments described below. By using this threshold, high sensitivity could be achieved at the expense of low specificity.

3.2.1 Off-line Learning

The data set was divided into a training set of 2000 samples, and a test set of 4321 samples. Each data sample consisted of the four attributes used in the TRISS experiment—ISS, RTS, age, and type of injury. First, an off-line experiment using the TRISS methodology was conducted using the coefficients reported in [40, 42]. Figure 9(a) shows the ROC curve obtained from the experimental results. As for PFAM, the network configurations for individual and multiple classifiers described in section 3.1.1 were employed. All the off-line results are summarised in Table 3.

The results from TRISS and PFAM exhibited almost similar outputs in the experiment. Unlike the database of CCU patients, PFAM was able to achieve comparable results with those of TRISS in terms of accuracy, sensitivity, specificity, as well as area under the ROC curve. Figures 9(b) and 9(c) depicts the minimum and maximum areas under the ROC curves achieved by PFAM. Using the method described in [34] to compare the difference between the areas under the two ROC curves (TRISS and average PFAM results), it was found that there is no significant

difference (p=0.95) in the performance between TRISS and average PFAM networks for this database.

		ACC	SENS	SPEC	ROC:Area ± S.E.
TRISS		96.6	99.0	55.3	94.1 ± 0.4
Averages of 25 individual networks		96.3 (93.5-97.4)	98.0 (94.7-99.7)	61.0 (46.5-73.0)	94.2 ± 0.4 (93.2-95.7, 0.3-0.5)
Averages of 12 MCSs	Voting	97.4 (97.1-97.5)	99.1 (98.7-99.5)	62.3 (58.0-65.5)	
	Bayesian	97.1 (96.7-97.3)	99.7 (99.6-99.9)	42.5 (33.0-49.0)	
	BKS	97.6 (97.5-97.7)	99.3 (99.1-99.6)	62.5 (59.0-66.5)	

Table 3 Off-line learning results for the database of trauma patients using the TRISS methodology, as well as the individual and multiple PFAM systems operating at 50% threshold. Figures in parentheses indicate the maximum and minimum results.

The accuracy index could be further enhanced in the three MCSs. Compared to individual averaged results, voting and BKS were able to improve all the three performance indices. However, the Bayesian approach achieved relatively high results for sensitivity and low results for specificity. As pointed out by Xu *et al* [27], stability of the Bayesian formalism greatly relies on the confusion matrices. If the confusion matrices of individual classifiers are well learned, then the Bayesian approach could produce very good results. By inspecting the confusion matrices, we noticed that the elements biased towards the *survival* class as might be expected which, in turn, causes the trade-off between sensitivity and specificity.

3.2.2 On-line Learning

In on-line learning, all the 6321 samples were presented once to the PFAM network. As pointed out earlier, the basic threshold of 50% was adopted throughout the experiment owing to the difficulty of maintaining a close level of accuracy, sensitivity, and specificity using equation (1). Figure 10 plots all the on-line performance indices, averaged across 12 runs, using a moving window of 1000 samples, and the growth pattern of the ART_a categories. The error bars are the 95% confidence intervals of the averages. Again, normality tests were conducted to check normality of the on-line samples, and an average of 98.7% normality test results were obtained for 100 samples. The end results, averaged across 12 ensembles, for the on-line experiments were 97.1% accuracy, 98.0% sensitivity, and 63.8% specificity. From this experiment, the on-line results reveal that PFAM is able to maintain the off-line performances, *i.e.* the percentages of

accuracy, sensitivity, and specificity are similar to those of the individual networks given in Table 3. This is the advantage of PFAM as it is able to learn incrementally using as much information as possible in attempting to formulate a good solution for the problem at hand. 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.

3.2.3 Dual-mode Learning

As argued in section 3.1.3, it is more practical to impose an off-line learning phase to establish a knowledge-base in the system before allowing it to learn on-line. This learning strategy has been shown to be able to achieve good performance using the CCU data set. The same network configurations and experimental procedure as in section 3.1.3 was adopted here. The data set was divided into a training set of 2000, and the subsequent 4321 samples constituted the test set, with on-going learning. Table 4 summarises the end results of the dual-mode experiments. Once again, this dual-mode learning strategy is able to achieve a better performance than either the off-line or on-line learning approaches.

		ACC	SENS	SPEC
Averages of 25 individual networks		97.0 (94.1-98.5)	98.0 (95.1-99.5)	71.4 (44.7-86.8)
Averages of 12 MCSs	Voting	98.2 (97.9-98.5)	99.1 (98.9-99.5)	74.6 (68.4-81.6)
	Bayesian	97.8 (97.4-98.1)	99.8 (99.6-100.0)	46.9 (34.2-55.3)
	BKS	98.6 (98.1-99.0)	99.5 (99.1-99.9)	76.5 (65.8-84.2)

Table 4 Dual-mode learning (moving average) results at the end of the experiments for the database of trauma patients. Figures in parentheses indicate the maximum and minimum results.

Figure 11 shows the dual-mode learning results of the individual and multiple classifiers with the 95% confidence intervals for the respective performance indices. A randomly selected sample of 100 gave an average of 98.4% for the normality probability test, which confirmed the validity of the *t*-distribution analysis for confidence intervals. From Figure 11, it can be seen that the results for specificity fluctuated significantly. All four plots for specificity show that the performance declined at the beginning of on-line learning, and improved progressively as more and more input samples were presented. This tendency to fluctuate might be due to a disruption to the knowledge-base of the network, which had been constructed using the off-line training set, when the network was switched to on-line learning. Nevertheless, another important contributing

factor to the variation might be the small number of exemplars of non-survival patients. This is because the variation only occurred in the specificity index, whereas the outcomes of accuracy and sensitivity were steady throughout the experiments as shown in Figure 11.

Figure 12 shows a comparison of classification accuracy among the individual and multiple classifier systems. The MCSs achieved a better result than individual classifiers, thereby justifying the use of multiple classifiers in this problem. In contrast to the database of CCU patients, voting is able to achieve a close performance to the Bayesian formalism. Again, the reason for this observation might be the poorly learned confusion matrices of the Bayesian decision combination method. The BKS approach demonstrated, for a second time, a superior performance among the three decision combination algorithms.

4 SUMMARY

The synthesis of FAM and the PNN has resulted in a new hybrid system—PFAM. We believe that the PFAM system has definite potential for pattern classification tasks. In this paper, the applicability of PFAM as an autonomously learning system for clinical decision support is demonstrated. Two medical domains, *i.e.* the databases of CCU and trauma patients, have been used in this study. In addition, algorithms for combining multiple modules of PFAM have been developed to form a multiple classifier framework in order to improve overall accuracy. Useful performance figures from multiple PFAM classifiers were obtained across the two medical domains here which potentially could benefit from computer-aided decision support. More importantly, the system is capable of learning incrementally and on-line with minimum intervention, other than the provision of verified supervisory signals. This autonomy may enable domain experts, such as doctors and clinicians, to design useful diagnostic systems with greater flexibility, according to their own criteria and operational requirements, without having to rely on computer systems engineers.

Three different learning strategies, namely off-line, on-line, and dual-mode learning, were studied. The results obtained were comparable to logistic regression analysis. Among the three strategies, the dual-mode learning scheme appeared to be a more practical approach to devising a comprehensive, autonomous learning system. 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.

For the three multiple classifier algorithms, the BKS approach exhibited the best performance. This finding corresponds to the results reported in [28] where the BKS formalism 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.

Although good performance is a pre-requisite for any system, further work is necessary to address some of the practical issues of applying PFAM as a usable, useful and valid decision support tool. One of the main criticisms of neural network applications is the opaqueness of the network outcomes. Usually, a set of input features is fed to the network, and a result is produced. Users have no understanding of what happens in-between, which in turn could lead to a strong resistance to acceptance of the network predictions. This concern is particularly true in medical domains. ARTMAP systems have recently been endowed with the capability of symbolic rule extraction to provide explanatory facilities for the network's reasoning. Carpenter and Tan [43] show that it is possible to derive fuzzy rules from the learned weights in FAM which are compact and comprehensible to the domain expert. In addition, Downs *et al.* [5] demonstrate that the extracted rules from FAM are in accord with an expert pathologist's opinions in a breast cancer diagnosis task. Thus, further research is needed to incorporate this rule extraction capability into PFAM to enable the system to justify its predictions.

At present, it is assumed that all the input attributes are accessible for PFAM to generate a prediction. This may not be the case in many real applications. For instance in medical diagnosis, some of the data items, such as ECG measurements, X-ray and other radiographic images, need to be interpreted and encoded by domain experts and thus might not be available immediately. To address the issues associated with missing or incomplete data, the multiple classifier framework can be further exploited to combine predictions from disparate sources sequentially [44]. In this way, a modularised PFAM-based system can be formed where the most instantly available data are grouped together to a classifier for the system to give an initial prediction. Then, information collated later may be fused to another classifier to reinforce or counteract the prediction. As more and more information becomes available, this sequential decision combination approach enables the modularised PFAM system to make predictions concomitant with increasing confidence as time goes on.

References

[1] S.S. Cross, R.F. Harrison, and R.L. Kennedy, Introduction to Neural Networks, *The Lancet* 346 (1995) 1075-1079.

[2] B. Apolloni, G. Avanzini, N. Cesa-Bianci, and G. Ronchini, Diagnosis of Epilepsy via Backpropagation, *Proc of the Int Joint Conf on Neural Networks*, II, (1990) 571-574.

[3] D. Bounds, P. Lloyd, and B. Mathew, A Comparison of Neural Network and Other Pattern Recognition Approaches to the Diagnosis of Low Back Disorders, *Neural Networks* 3 (1990) 583-591.

[4] R.F. Harrison, S.J. Marshall, and R.L. Kennedy, A Connectionist Approach to the Early Diagnosis of Myocardial Infarction, *Proc of the Third European Conf on Artificial Intelligence in Medicine (AIME-91)* (1991) 119-128.

[5] J. Downs, R.F. Harrison, R.L. Kennedy, and S.S. Cross, Application of the Fuzzy ARTMAP Neural Network Model to Medical Pattern Classification Tasks, *Artificial Intelligence in Medicine* 8 (1996) 403-428.

[6] D.E. Rumelhart, G.E, Hinton, and R.J. Williams, Learning Internal Representation by Error Propagation, in: D.E. Rumelhart, and J.L. McLelland (Eds.), *Parallel Distributed Processing I*, (Cambridge, MA: MIT Press, 1986) 318-362.

[7] D.S. Broomhead, and D. Lowe, Multivariate Functional Interpolation and Adaptive Networks, *Complex Systems* 2 (1988) 321-355.

[8] J. Moody, C.J. Darken, Fast Learning in Networks of Locally-tuned Processing Units, *Neural Computation* 1 (1989) 281-294.

[9] G. Cybenko, Approximation by Superposition of a Sigmoidal Function, *Mathematics of Control, Signals and Systems* 2 (1989) 303-314.

[10] F. Girosi, and T. Poggio, Networks and the Best Approximation Property, *Biological Cybernetics* 63 (1990) 169-176.

[11] O. Fujita, Optimisation of the Hidden Unit Function in Feedforward Neural Networks, *Neural Networks* 5 (1992) 755-764.

[12] Y.Q. Chen, D.W. Thomas, and M.S. Nixon, Generating-Shrinking Algorithm for Learning Arbitrary Classification, *Neural Networks* 7 (1993) 1477-1489.

[13] V. Kadirkamanathan, and M. Niranjan, A Function Approach to Sequential Learning with Neural Networks, *Neural Computation* 5 (1993) 954-975.

[14] B. Fritzke, Growing Cell Structures—A Self-organizing Network for Unsupervised and Supervised Learning. *Neural Networks* 7 (1994) 1441-1460.

[15] G.A. Carpenter, and S. Grossberg, A Massively Parallel Architecture for a Self-Organising Neural Pattern Recognition Machine, *Computer Vision, Graphics and Image Processing* 37 (1987) 54-115.

[16] G.A. Carpenter, and S. Grossberg, The ART of Adaptive Pattern Recognition by a Self-Organising Neural Network, *IEEE Computer* 21 (1988) 77-88.

[17] G.A. Carpenter, S. Grossberg, and J.H. Reynolds, ARTMAP: Supervised Real-time Learning and Classification of Nonstationary Data by a Self-Organising Neural Network, *Neural Networks* 4 (1991) 565-588.

[18] 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 Trans on Neural Networks* 3 (1992) 698-712.

[19] G.A. Carpenter, S. Grossberg, and D.B. Rosen, Fuzzy ART : Fast Stable Learning and Categorisation of Analogue Patterns by an Adaptive Resonance System, *Neural Networks* 4 (1991) 759-771.

[20] D.F. Specht, Probabilistic Neural Networks, *Neural Networks* 3 (1990) 109-118.

[21] E. Parzen, On Estimation of a Probability Density Function and Mode, *Annals of Mathematical Statistics* 33 (1962) 1065-1076.

[22] D.F. Specht, Enhancements to Probabilistic Neural Networks, *Proc. IEEE Int. Conf. Neural Networks* I (1992) 761-768.

[23] P. Burrascano, Learning Vector Quantisation for the Probabilistic Neural Network, *IEEE Trans on Neural Networks* 2 (1991) 458-461.

[24] C. P. Lim, and R.F. Harrison, Modified Fuzzy ARTMAP Approaches Bayes Optimal Classification Rates : An Empirical Demonstration, *Neural Networks* (in press).

[25] C. P. Lim, and R.F. Harrison, An Incremental Adaptive Network for On-line Supervised Learning and Probability Estimation, *Neural Networks* (in press).

[26] C.P. Lim, and R.F. Harrison, Estimation of Bayesian *a posteriori* Probability with an Autonomously Learning Neural Network, *Proc of Int Conf on Control '96* 1 (1996) 199-204.

[27] L. Xu, A. Krzyzak, and C. Y. Suen, Methods of Combining Multiple Classifiers and Their Applications to Handwriting Recognition, *IEEE Trans on Systems, Man, and Cybernetics* 22 (1992) 418-435.

[28] Y.S. Huang, and C.Y. Suen, A Method of Combining Multiple Experts for the Recognition of Unconstrained Handwritten Numerals, *IEEE Trans on Pattern Analysis and Machine Intelligence* 19 (1995) 90-94.

[29] T.K. Ho, J.J. Hull, and S.N. Srihari, Decision Combination in Multiple Classifier Systems, *IEEE Trans. on Pattern Analysis and Machine Intelligence* 16 (1994) 66-75.

[30] M. Bland, *An Introduction to Medical Statistics*, (Oxford University Press, Oxford, 1987).

[31] R.W. Parsons, K.D. Jamrozik, M.S.T. Hobbs, and D.L. Thompson, Early Identification of Patients at Low Risk of Death after Myocardial Infarction and Potentially Suitable for Early Hospital Discharge, *British Medical Journal* 308 (1994) 1006-1010.

[32] R.L. Kennedy, R.F. Harrison, H.S. Fraser, S. Fletcher, A.M. Burton, L.N. McStay, and K.L. Woods, Prediction of Severe Complications and Mortality in Patients Admitted to a Coronary Care Unit, *Research Report No.* 6xx, Department of Automatic Control and Systems Engineering, University of Sheffield.

[33] J.A. Hanley, and B.J. McNeil, The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve, *Radiology* 143 (1982) 29-36.

[34] J.A. Hanley, and B.J. McNeil, A Method for Comparing the Areas under Receiver Operating Characteristic Curves Derived from the same cases, *Radiology* 148 (1983) 839-843.

[35] J.A. Swets, and R.M. Pickett, *Evaluation of Diagnostic Systems*, (Academic Press, London, 1982).

[36] M. Meistrell, Evaluation of Neural Network Performance by Receiver Operating Characteristic Analysis: Examples from the Biotechnology Domain, *Computer Methods and Programs in Biomedicine*, 32 (1990) 73-80.

[37] D.M. Green, and J.A. Swets, *Signal Detection Theory and Psychophysics*, (Wiley, New York, 1966). Reprinted by (Kreiger, New York, 1974).

[38] R.F. Harrison, C.P. Lim, and R.L. Kennedy, Autonomously Learning Neural Networks for Clinical Decision Support, *Proc. of the Int. Conf. on Neural Networks and Expert Systems in Medicine and Healthcare (NNESMED-94)* (1994) 15-22.

[39] *MINITAB Reference Manual, Release 9 for Windows* (MINITAB Inc., PA.: State College, 1993).

[40] D.W. Yates, M. Woodford, and S. Hollis, Preliminary Analysis of the Care of Injured Patients in 33 British Hospitals: First Report of the United Kingdom Major Trauma Outcome Study, *British Medical Journal* 305 (1992) 737-740.

[41] Preliminary Analysis of the Care of Injured Patients in Five Scottish Teaching Hospitals: First Report from the Scottish Trauma Audit Group (STAG), *Health Bulletin* 53 (1995) 55-65.

[42] C.R. Boyd, M.A. Tolson, and W.S. Copes, Evaluating Trauma Care: the TRISS Method. *Journal of Trauma* 27 (1987) 370-378.

[43] G.A. Carpenter, and A.H. Tan, Rule Extraction: From Neural Architecture to Symbolic Representation, *Connection Science* 7 (1995) 3-27.

[44] C.P. Lim, and R.F. Harrison, A Multiple Neural Network Architecture for Sequential Evidence Aggregation and Incomplete Data Classification, *Research Report No. 641*, Department of Automatic Control and Systems Engineering, University of Sheffield.

Appendix A

The Probabilistic Fuzzy ARTMAP Algorithm

Following the notation used in the FAM paper [15], 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 initialised 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 minimised, α_a should be initialised 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 initialisation procedure is also applicable to ART$_b$. In the map field, the vigilance parameter, ρ_{ab}, is also initialised 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, $\boldsymbol{a} \in [0,1]^M$, in F_0^a to a $2M$-dimensional vector in F_1^a as:

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

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

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

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

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

(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, $\boldsymbol{x}^{ab}$, obeys

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

which forms a prediction from the Jth ART_a category to the Kth ART_b target class via $\boldsymbol{w}_J^{ab}$.

(vi) Perform the map field vigilance test as

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

(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 centre 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 centre 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, 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

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

(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 [24]. 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.*,

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

This technique is, in fact, utilised for evidence gathering and uncertainty reasoning using the Bayesian framework in artificial intelligence.

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

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

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

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

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, Xu *et al* [24] proposes to use the following equation to approximate equation (B4)

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

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.*,

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

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, Huang and Suen [25] 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_2 \ e_1	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, $\boldsymbol{x}$, 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 $e_1(\boldsymbol{x})=3$ and $e_2(\boldsymbol{x})=4$. 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

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

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

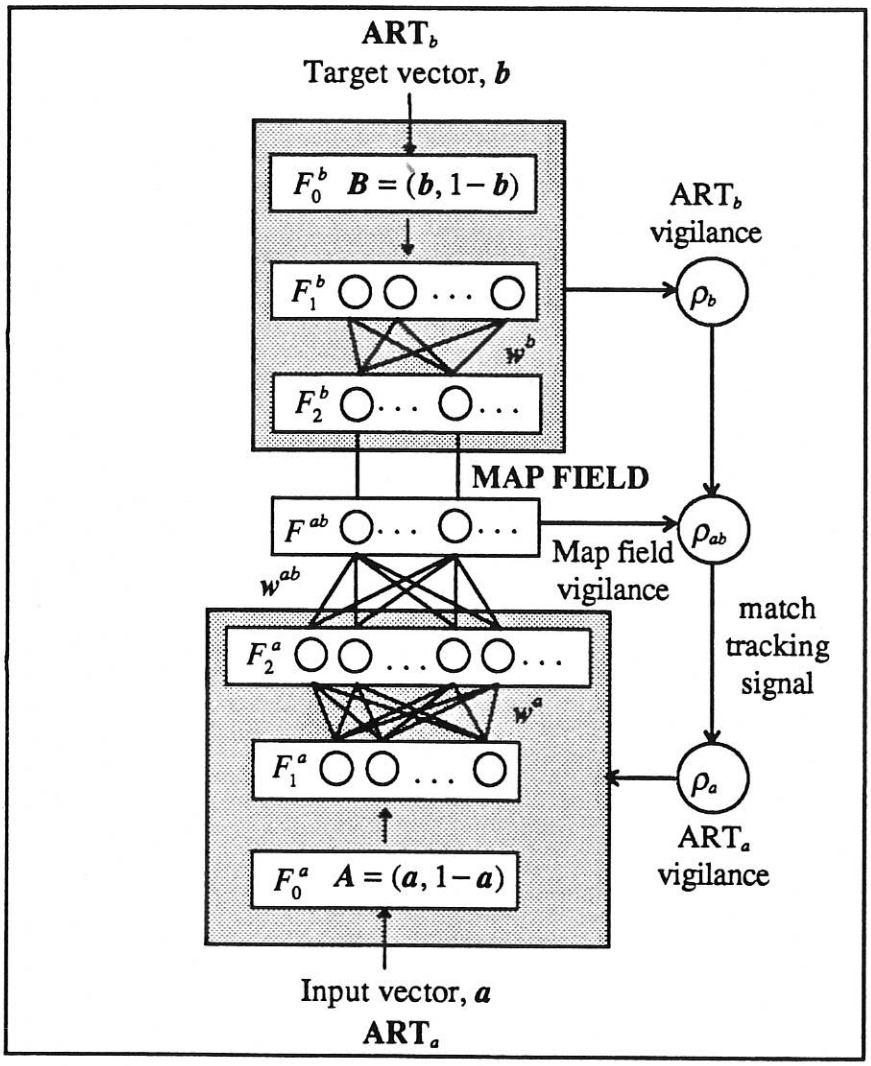


Figure 1 The Fuzzy ARTMAP network

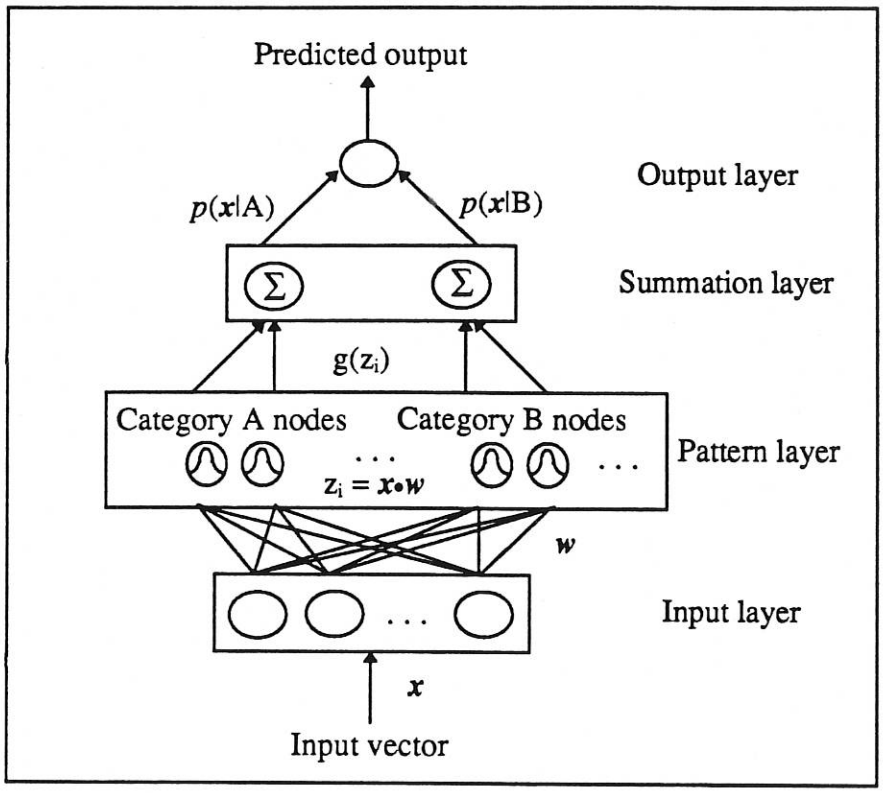


Figure 2 The Probabilistic Neural Network

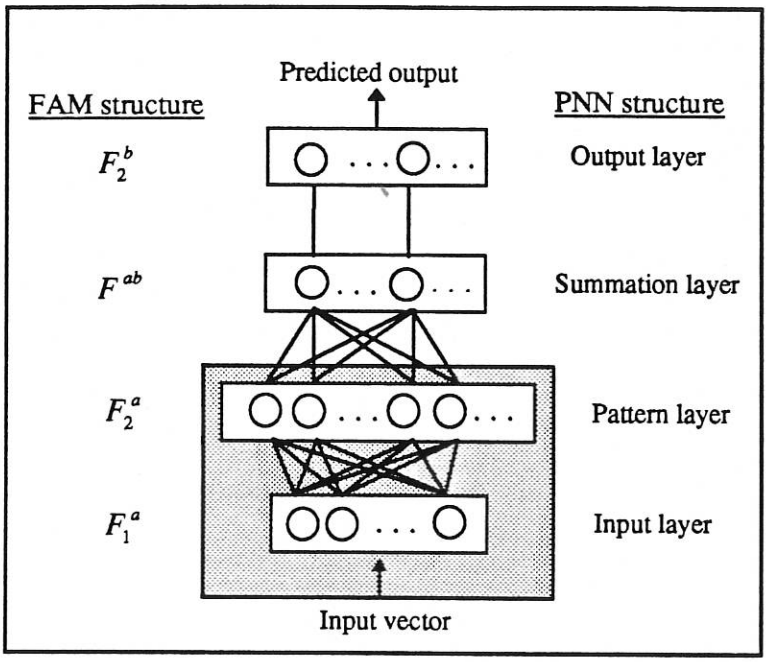


Figure 3 A schematic diagram depicting the similarity of the network structure between FAM and the PNN.

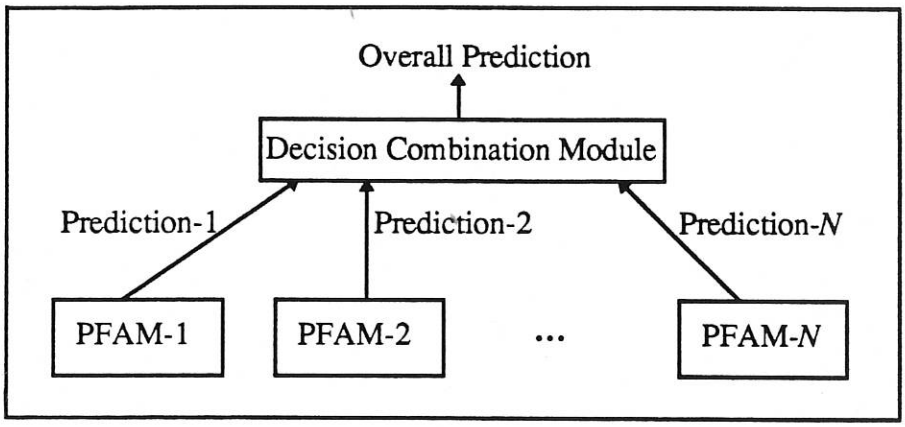
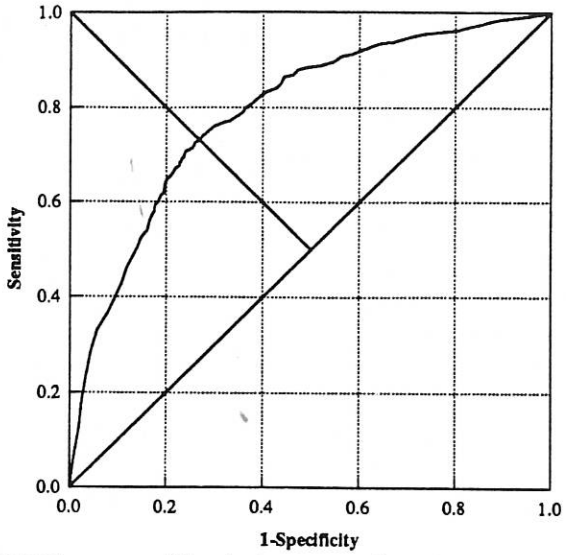
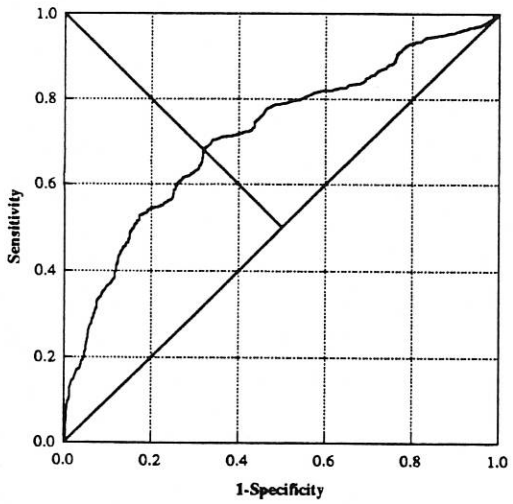


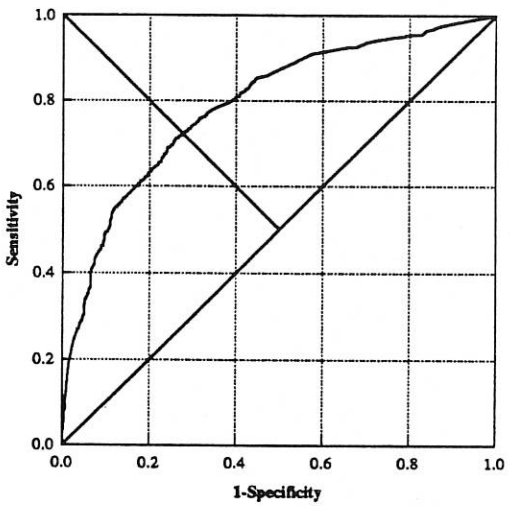
Figure 4 A schematic diagram of a multiple classifier system. There are N channels of independent PFAM classifiers. Predictions from these classifiers are combined using some decision combination algorithm to give an overall prediction.



(a) ROC curve of logistic regression (area=79.0%)



(b) ROC curve of PFAM (area=71.8%)



(c) ROC curve of PFAM (area=79.3%)

Figure 5 ROC plots of (a) logistic regression, as well as the (b) minimum and (c) maximum areas under the ROC curve achieved by PFAM for the database of CCU patients.

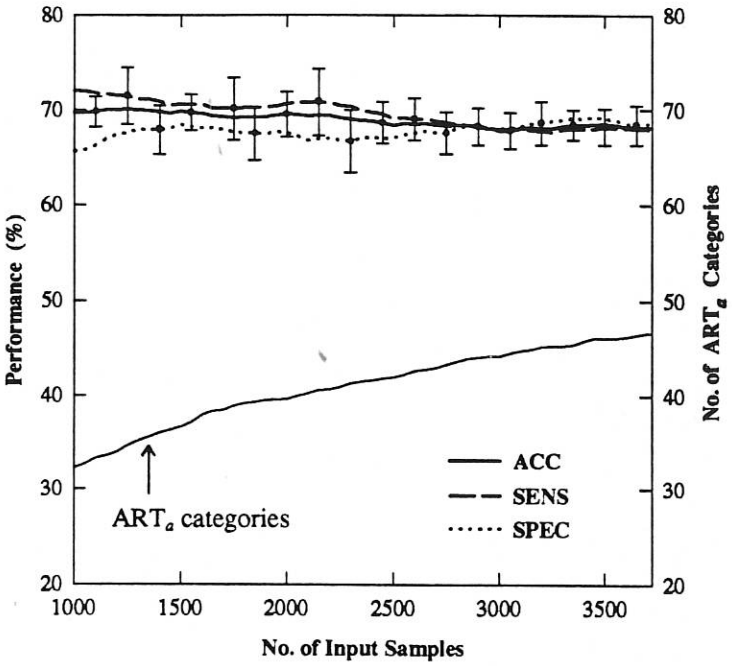
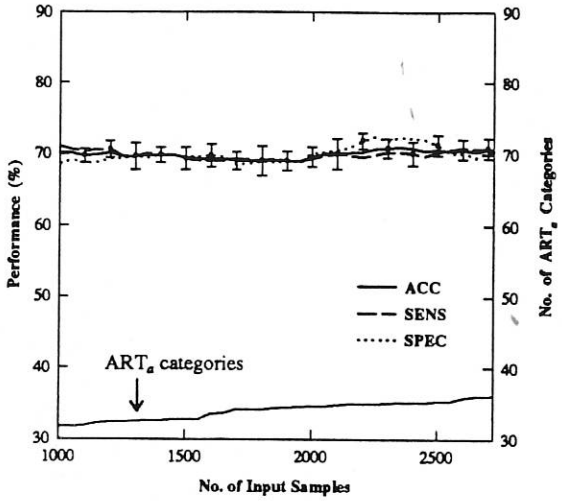
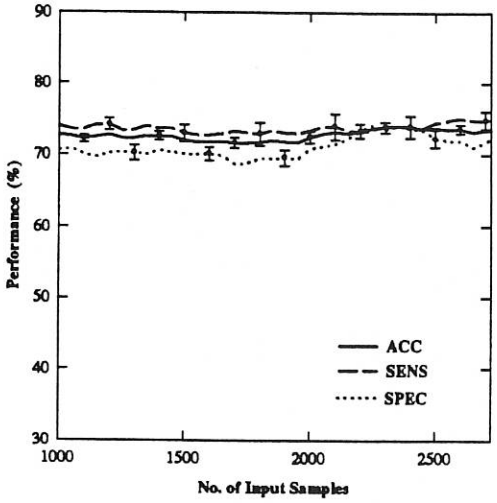


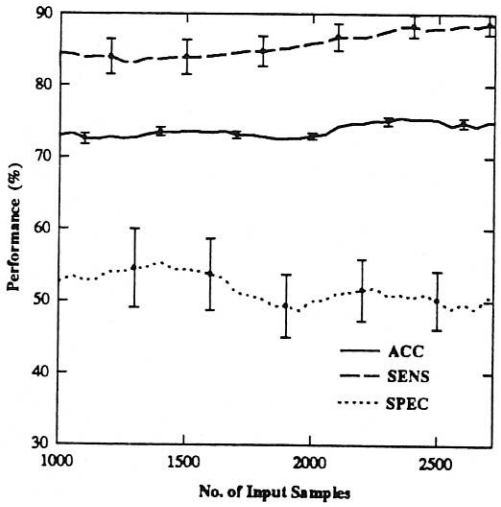
Figure 6 On-line plots of accuracy, sensitivity, specificity, and the growth pattern of ART_a categories for the database of CCU patients. The results are bounded by their respective 95% confidence intervals indicated as error bars.



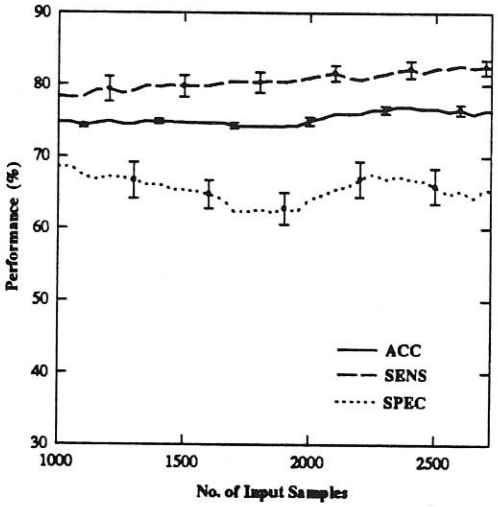
(a) average results of 25 individual classifiers, and the number of ART$_a$ categories



(b) average results of 12 multiple classifiers (voting)



(c) average results of 12 multiple classifiers (Bayesian)



(d) average results of 12 multiple classifiers (BKS)

Figure 7 Dual-mode learning results for the database of CCU patients. The error bars indicate the respective 95% confidence intervals for the performance indices.

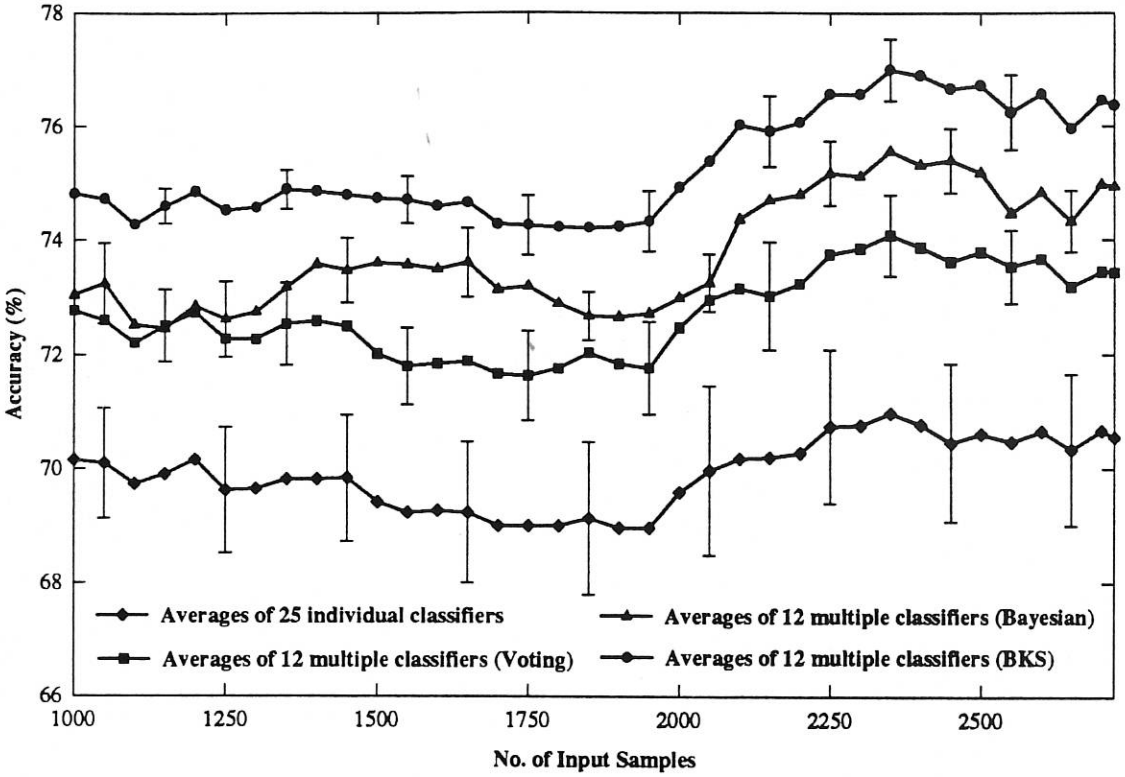
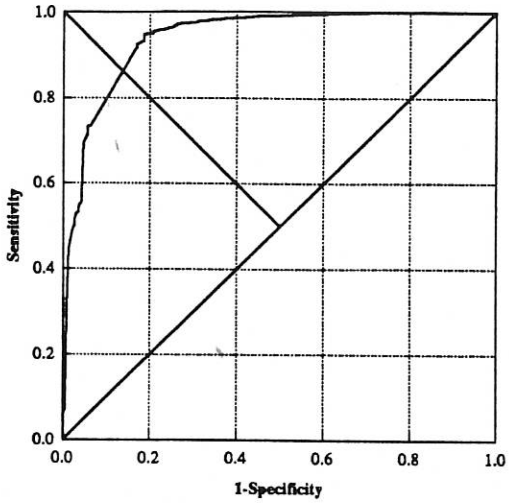
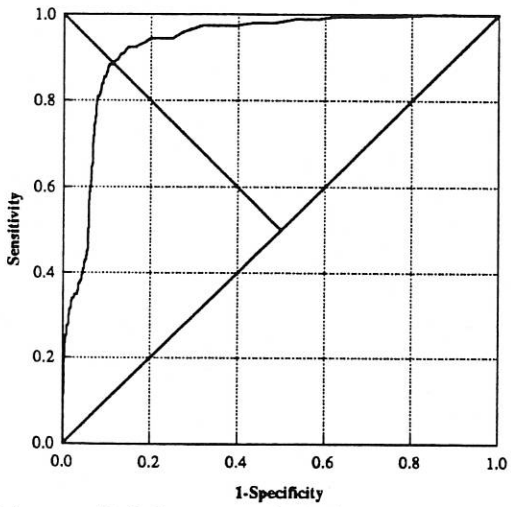


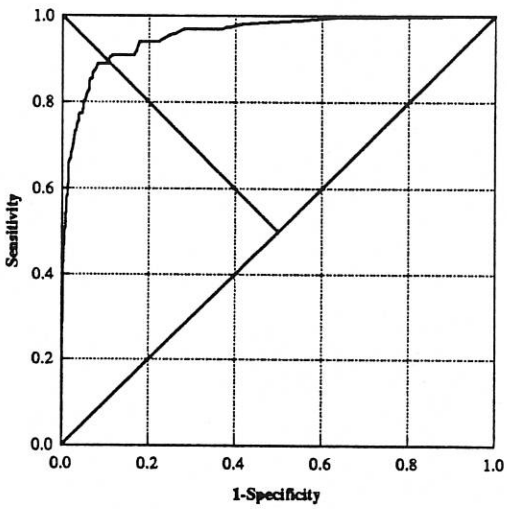
Figure 8 A comparison of classification accuracy of individual and multiple classifiers for the database of CCU patients. The error bars indicate the 95% confidence intervals.



(a) ROC curve of TRISS (area=94.1%)



(b) ROC curve of PFAM (area=93.2%)



(c)ROC curve of PFAM (area=95.7%)

Figure 9 ROC plots of (a) TRISS, as well as the (b) minimum and (c) maximum areas under the ROC curve achieved by PFAM for the database of trauma patients.

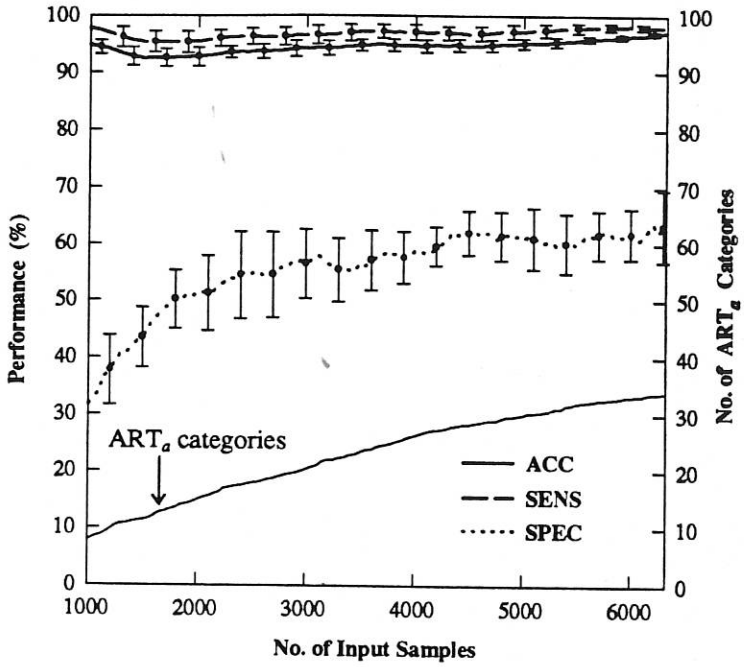
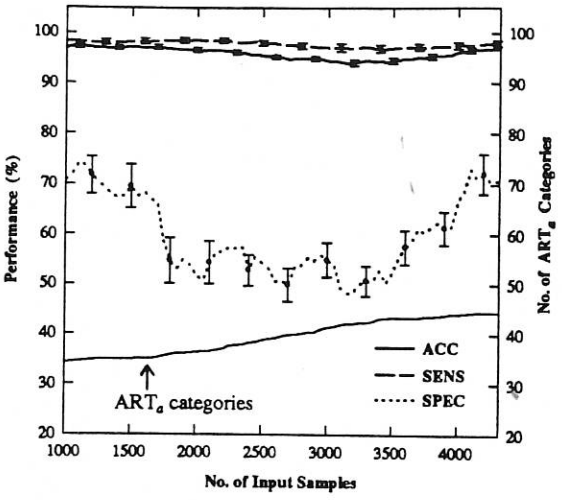
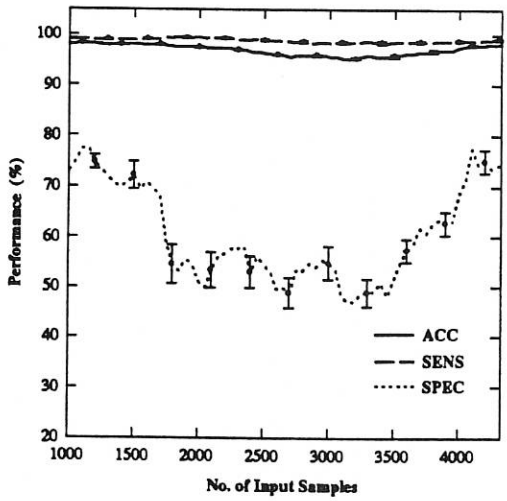


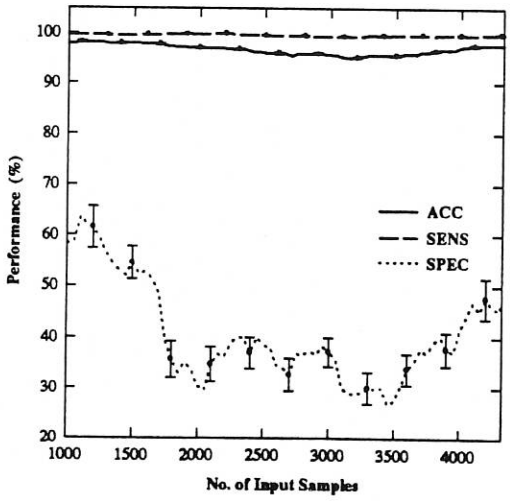
Figure 10 On-line plots of accuracy, sensitivity, specificity, and the growth pattern of ART$_a$ categories for the database of trauma patient. The results are bounded by their respective 95% confidence intervals indicated as error bars.



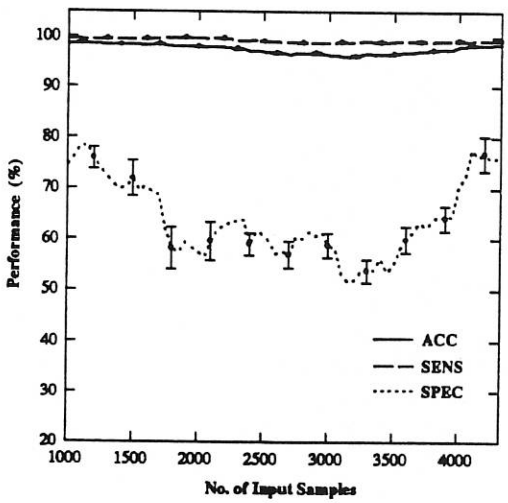
(a) average results of 25 individual classifiers, and the number of ART$_a$ categories



(b) average results of 12 multiple classifiers (voting)



(c) average results of 12 multiple classifiers (Bayesian)



(d) average results of 12 multiple classifiers (BKS)

Figure 11 Dual-mode learning results for the database of trauma patients. The error bars indicate the respective 95% confidence intervals for the performance indices.

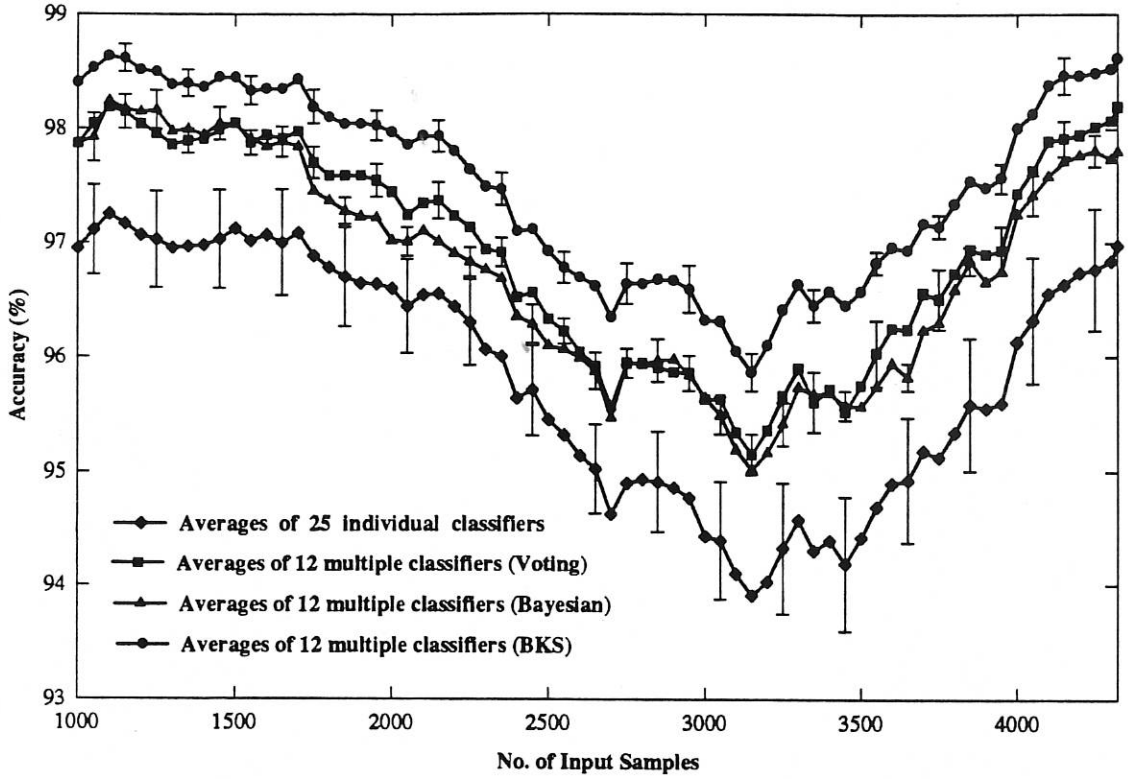


Figure 12 A comparison of classification accuracy of individual and multiple classifiers for the database of trauma patients. The error bars indicate the 95% confidence intervals.