Probabilistic ensemble Fuzzy ARTMAP optimization using hierarchical parallel genetic algorithms

Chu Kiong Loo · Wei Shiung Liew ·
Manjeevan Seera · Einly Lim

Abstract In this study, a comprehensive methodology for overcoming the design problem of the Fuzzy ARTMAP neural network is proposed. The issues addressed are the sequence of training data for supervised learning and optimum parameter tuning for parameters such as baseline vigilance. A genetic algorithm search heuristic was chosen to solve this multi-objective optimization problem. To further augment the ARTMAP's pattern classification ability, multiple ARTMAPs were optimized via genetic algorithm and assembled into a classifier ensemble. An optimal ensemble was realized by the inter-classifier diversity of its constituents. This was achieved by mitigating convergence in the genetic algorithms by employing a hierarchical parallel architecture. The best-performing classifiers were then combined in an ensemble, using probabilistic voting for decision combination. This study also integrated the disparate methods to operate within a single framework, which is the proposed novel method for creating an optimum classifier ensemble configuration with minimum user intervention. The methodology was benchmarked using popular data sets from UCI machine learning repository.

C. K. Loo (✉) · W. S. Liew · M. Seera
Faculty of Computer Science and Information Technology,
University Malaya, Kuala Lumpur, Malaysia
e-mail: ckloo.um@gmail.com; ckloo.um@um.edu.my

W. S. Liew
e-mail: liew.wei.shiung@gmail.com

M. Seera
e-mail: mseera@um.edu.my

E. Lim
Faculty of Biomedical Engineering, University Malaya,
Kuala Lumpur, Malaysia
e-mail: einly_lim@um.edu.my

Keywords Ensembles · Fuzzy ARTMAP · Plurality voting · Parallel genetic algorithm · Feature selection

1 Introduction

Fuzzy ARTMAP (FAM) neural networks [1] are commonly used for pattern classification problems. Under a supervised learning condition, the FAM is able to form complex correlations between a given multi-dimensional input or exemplars and a multi-dimensional output or category labels. However, the classification accuracy is dependent on the parameter settings and the order in which exemplars were presented during supervised learning. Finding a combination of parameters that will yield the best classification performance is essentially an optimization task.

FAM have been used in a wide range of applications. In fault diagnosis of rolling element bearings, a FAM ensemble based on the improved Bayesian belief method is used [2]. A Gaussian ARTMAP model is proposed by Mokhtar et al. [3] for building better and more efficient energy management systems. Simplified FAM model is used in detecting faults of induction motor, with the inputs of transient current signals [4]. A FAM neural network is employed by Liang et al. [5] in classifying and grading yarn surface qualities with satisfactory performance. For fault detection and diagnosis in a power generation plant, a FAM network with evolutionary programming is proposed, with consistent experimental results [6].

The FAM optimization task is required to explore several different characteristics for optimum classifier performance. The parameter settings of the FAM control the fundamental architecture of the classifier and affect its ability to learn patterns during the training process.

In addition, the internal knowledge base of the FAM grows and changes shape with each additional exemplar presented to the neural network during training. A biased training sequence, or one which involves poorly defined exemplars, may affect the FAM's classification ability. The neural network is somewhat able to compensate for the sequencing problem if the exemplars are presented repeatedly during training [7], but the problem will be accumulative for FAM classifiers relying on online learning. A number of methods were proposed for overcoming the shortfalls of the FAM, such as parameter tuning [8–11], ensemble learning [12, 13], or by modifying the neural network architecture to create variants of the FAM [7, 14–17].

Using the FAM neural network, the proposed framework consisted of the following methods; given a classification task with a data set of pattern examples, parameter tuning was performed, using genetic algorithms (GA) to search for effective combinations of parameters and training sequence for creating a trained FAM classifier. The GA is able to converge onto optimal points in the search space through competitive eliminations and refining the search in between the surviving candidates. With the goal of developing an ensemble of classifiers, the role of the GA in this framework was to generate and iteratively evolve a population of FAMs not only for improved classification accuracy, but for diversity between individual classifiers. Traditional single-population GA have a tendency to converge on a single optimum point, mainly due to the elitist selection process. Convergence in this case is undesirable due to the intention to create a classifier ensemble.

Ensemble learning has recently gained much attention in various learning tasks such as classification, clustering, and regression problems [18]. A hybrid model consisting of Fuzzy Min–Max (FMM) neural network and the random forest (RF) model, comprising an ensemble of classification and regression trees (CART) for condition monitoring of induction motors is developed by Seera et al. [19]. Using a similar model, FMM, CART, and RF ensemble is used in three benchmark medical data sets, with positive outcome from the experiments [20]. An artificial neural network ensemble, composed using a multi-class classifier is used for traffic sign recognition and hand-written digit recognition, with good results [21].

The rationale for an ensemble is to combine complementary information from multiple diverse classifiers to achieve a classification accuracy higher than any individual classifier. A variant GA known as hierarchical fair-competition parallel genetic algorithms (HFCPGA) was proposed in [22], where genetic convergence was mitigated by distributing chromosomes across multiple subpopulations instead of a single population. The populations are independent of each other and are arranged in a hierarchy so that candidates will compete only against other candidates with similar levels of fitness.

Diversity was considered an important factor when designing ensembles of classifiers. Zenobi and Cunningham [23] discussed the possibility of having an ensemble of suboptimum classifiers outperforming an ensemble of optimum classifiers due to successful trade-off between diversity and accuracy. To ensure the GA generates a diverse set of FAM configurations, a number of methods were implemented in the framework such as the HFCPGA model and feature subset selection. Feature subset selection, wherein the FAM is trained using only a partial representation of the training patterns, would reduce the complexity of the resultant trained FAM, otherwise known as the problem of "overfitting" [24]. Thus, instead of having an ensemble of overfitted classifiers, the ensemble would consist of classifiers that were each trained on a small subset of the training data set. Several studies showed the advantages of an ensemble of diverse and specialized FAMs trained using feature-selected data sets [25–28]. For this framework, feature subsets generated by the GA would be used to filter the training data set before being used for training the FAM.

Having generated a diverse population of FAMs, an ensemble was created using classifiers with the best classification accuracy. The final ensemble will be integrated by means of a probabilistic voting strategy [29, 30] that assigns each classifier with a reliability index and skewing ensemble decisions in favor of high-reliability classifiers while reducing the voting weights of classifiers with poor accuracy. When classifying an object, the reliability of the classification was computed from the probabilistic output from each classifier. This adds a new dimension to classification output from the ensemble, in which reliability represents the "confidence" of the ensemble in its prediction.

Similar research has been conducted in the various literatures. Radtke et al. [31] proposed a similar two-step method for optimizing classifier training using feature subsets, and for crafting ensembles of classifiers, both using different evolutionary algorithm methodologies. Using an automated method for selecting classifiers for an ensemble may be considered for future development of our existing framework. Ishibuchi et al.'s methodology for generating ensembles of fuzzy rule-based classifiers [32] utilized a multi-objective GA for finding Pareto-optimum combinations of accuracy, and size and complexity of the fuzzy rule sets, whereas the method proposed in this study selected the FAMs which showcased the highest classification accuracy regardless of the internal configuration.

In this work, the FAM classifier requires training sequence and parameter optimization to perform with peak accuracy. A GA was suggested for this task, using a hierarchical and parallelized architecture to reduce homogeneity and to distribute convergence over multiple points.

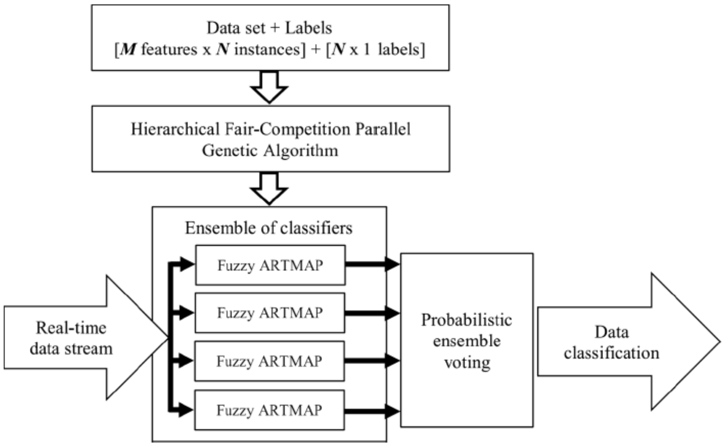
Data set + Labels
[*M* features x *N* instances] + [*N* x 1 labels]
Hierarchical Fair-Competition Parallel Genetic Algorithm
Ensemble of classifiers
Fuzzy ARTMAP
Fuzzy ARTMAP
Fuzzy ARTMAP
Fuzzy ARTMAP
Real-time data stream
Probabilistic ensemble voting
Data classification

Fig. 1 Chart of the proposed method for creating an ensemble of optimized Fuzzy ARTMAP classifiers

A population of optimal FAMs was generated, from which an ensemble of classifiers was created. Ensemble decision was based on probabilistic voting of the individual classifiers, weighted according to each FAMs reliability. The framework was proposed as an automated method for generating an ensemble of trained FAM classifiers for a given pattern classification task with consistently high accuracy. The workings of the FAM classifier, the genetic optimization methods, and the probabilistic voting scheme is detailed in Sect. 2. Section 3 details the experimental setup, where several data sets were used for benchmark. Results and discussions are included in Sect. 4. Concluding remarks is given in Sect. 5.

2 Description of framework

The proposed framework for generating an optimized ensemble of classifiers consist of three parts: classifier optimization, ensemble creation, and classifier combination, as illustrated in Fig. 1. Given a pattern classification task with a data set of labeled exemplars, the algorithm first generates a population of random candidates, or chromosomes, each of which can be used to create a single trained FAM classifier. The hierarchical parallel genetic optimization method searches for optimum chromosomes through genetic selection, crossover, and mutation, iterated across several generations from the initial starting population. Subsequently, the new population of candidates will consist of optimized chromosomes from which the best were selected to create an ensemble of FAM classifiers. Pattern classification was performed by the ensemble by having the constituent FAMs classify an unknown object. Individual classification decisions were combined using probabilistic voting to select the final ensemble decision.

The individual modules of the framework consist of the FAM pattern classifier, a HFCPGA for classifier optimization and a probabilistic voting step to combine classifier decision in an ensemble.

2.1 Fuzzy ARTMAP

Fuzzy ARTMAP neural networks operate on the principle of adaptive resonance theory for creating correlating connections between a given multi-dimensional input vector and the corresponding output representing its class category. The FAM neural network architecture is illustrated in Fig. 2. The system consists of two ART modules with a mapping field connecting both modules. Given a set of exemplars and their respective category labels, the input vectors are fed into the ART$_a$ input module while the labels go to the ART$_b$ output module. The supervised learning method modifies the weighted connections between F_2^a, F^{ab}, and F_2^b to achieve resonance between the input and output. Repeated over multiple times with a variety of exemplars, the FAM learns to recognize and classify similar input objects into the most likely class label based on its learning experiences.

Details of the FAM operations [1] is given in further details, as follows:

1. Given an object to be classified, in the form of a normalized vector a with M attributes in the range of 0 to 1.
2. Vector a along with its complement, $a^C = 1 - a$, was encoded as a single input object A:

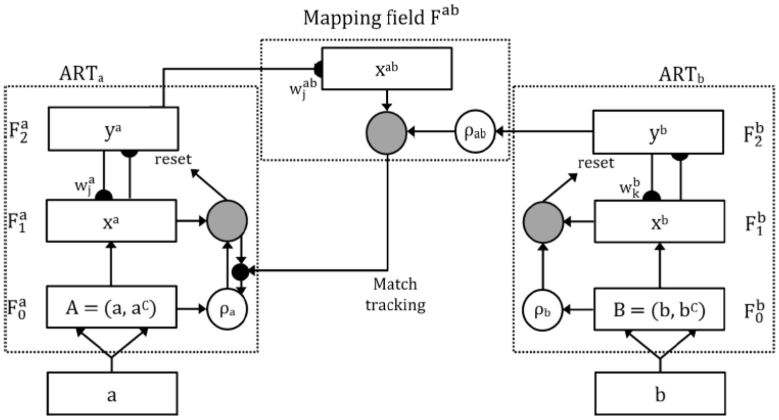

Fig. 2 Structure of the Fuzzy ARTMAP neural network

$$A = (a, a^C) \tag{1}$$

3. Among the nodes in F_2^a that have not been selected, a node J was selected with the maximum choice function:

$$T_j = |A \wedge w_j| + (1 - \alpha)(M - |w_j|) \tag{2}$$

Uncommitted nodes were initialized with all values of w_J set to 1.

4. The selected node was matched against the bottom-up input A. The field F_1^a represented the fuzzy intersection between the input vector A and the weights of the node, w_J. The vector representing the match between input vector A and the selected node weights w_J is represented as:

$$x = A \wedge w_J \tag{3}$$

where $\wedge$ denotes the component-wise minimum, or fuzzy intersection, of the bottom-up input vector A and the top-down expectation w_J. At this point, one of several cases may occur:

- Node J failed to meet the match criterion: $\frac{|x|}{|A|} < \rho_a$. Another node was chosen and Step 3 was repeated.
- Node J meets the match criterion: $\frac{|x|}{|A|} \geq \rho_a$. The node was used to make a classification prediction for object A.
  - The object A was transmitted along the weighted connections between F_2^a and the mapping field F^{ab}. A successful map between the input A and output B was determined by the map field match criterion:

$$|x^{ab}| \geq \rho_{ab}|y_b| \tag{4}$$

- The match tracking equation was designed to trigger a mismatch reset if the selected node J makes a wrong prediction:

$$\frac{\mathrm{d}\rho_a}{\mathrm{d}t} = -(\rho - \bar{\rho}) + \Gamma R r^{\epsilon} \tag{5}$$

In the case of an incorrect prediction, the predictive error parameter R is set to 1 and the current vigilance parameter ρ_a was incremented according to Eq. 5 until ρ_a was larger than the match value $\frac{|x|}{|A|}$, thus failing the match criterion. The algorithm then loops back to select a new node J and repeat Step 3. In the meantime, ρ_a decays by the match tracking parameter, ϵ before the next node J was selected. This mechanism was designed to minimize predictive errors by stimulating search between nodes, while maximizing the network's generalization ability through manipulating the current vigilance parameter.

- In the case where object a was successfully mapped to class b, or if the selected node J was uncommitted, the system learns by incorporating the input object A into node J:

$$w_J^{\mathrm{new}} = (1 - \beta)w_J^{\mathrm{old}} + \beta(w_J^{\mathrm{old}} \wedge A) \tag{6}$$

- The algorithm loops back to Step 1 for the next object to be classified.

The pseudocode of the process is given below.

```
TRAIN($DataSet, \alpha, \beta, \epsilon, \bar{\rho}$)
for $i \leftarrow 1$ to $N$ objects in DataSet
    do
        $A = [a_i, a_i^C]$
        $\rho_a = \bar{\rho}$
        for $j \leftarrow 1$ to $J$
            do
                while $(T_j = |A \wedge w_j| + (1 - \alpha)(M - |w_j|) \geq T_{max})$ and $(\frac{|A \wedge w_j|}{|A|} \leq \rho_a)$
                    do
                        if $J$ is uncommitted
                            then $w_J = \{w_1^J, w_2^J, ..., w_{2M}^J\} = \{1, 1, ..., 1\}$
                        if ($J$ is uncommitted) or $(|x^{ab}| \geq \rho_{ab}|y_b|)$
                            then and $w_J^{new} = (1 - \beta)w_J^{old} + \beta(w_J^{old} \wedge A)$
                            else
                                while $|x^{ab}| < \rho_{ab}|y_b|$
                                    do
                                        $\frac{d\rho_a}{dt} = -(\rho - \bar{\rho}) + \mathit{\Gamma} Rr^c$
                $\rho_a = \rho_a - \epsilon$
```

The ART-based neural network was dependent on its internal configuration of node weights, which in turn were affected by a number of factors such as the ARTMAP parameter settings and the training data used for learning. A number of approaches for ARTMAP optimization used evolutionary algorithms to search for the optimum training sequence [11] and ARTMAP parameter settings [33]. Kaylani et al. [16], however, proposed using multi-objective GA for optimizing directly the ARTMAP's internal topology, thus bypassing the need to determine the parameters.

In this study, however, we will focus on using GA for optimizing the training order and the ARTMAP parameters, given as:

- Baseline vigilance, $\bar{\rho}$. Setting zero vigilance allows a greater degree of generalization, while setting high vigilance only permits learning from highly specific exemplars.
- Choice parameter, α. Influences the degree of uniqueness of each committed node.
- Learning rate, β. Determines how quickly the nodes adapt and learn the given data.
- Match tracking parameter, ϵ. Determines the rate in which current vigilance returns to baseline after each predictive error by the selected node.

2.2 Hierarchical fair-competition parallel genetic algorithm

Genetic algorithms (GAs) are search heuristics employing the principles of natural evolution to search for and refine solutions for a given optimization task. A candidate solution to the problem is encoded in the form of a string of genes, each representing one variable, constituting into a single chromosome. A population of chromosomes are maintained over the course of several consecutive generations, during which the chromosomes are subject to competitive eliminations followed by genetic reproduction to create new chromosomes variants from the remaining survivors.

GAs have been used in the previous literature for optimizing classifier performance, such as generic fuzzy rule-based classifiers [32], radial basis function networks [34], and simple vector regression [35]. In all cases, optimizing a classifier requires the GA to search for an optimum combination of parameters, with each classifier having different parameters than the others.

One problem usually encountered especially with GAs with high turnover rate, is that of population convergence. With each successive round of eliminations and reproduction, the number of chromosomes sharing common genetic traits will increase, leading to the population clustering around a relatively small area in the solution space. Therein exists the possibility of premature convergence around a local optimum. Furthermore, in the context of this study, single-point convergence is counterproductive as it results in the final group of chromosomes to be homogeneous. For classifier ensembles, inter-classifier diversity is considered an important trait in order for the ensemble to perform better than its constituent classifiers. The default GA methodology is therefore unsuitable for our purpose to optimize individual FAMs for an ensemble of classifiers.

The HFCPGA [22] was conceived as a technique for mitigating the issue of premature convergence. The HFCPGA employs multiple populations of chromosomes, each evolving independently of each other. Distribution of chromosomes in each population is arranged in a hierarchy, ensuring that the chromosomes in each population face fair competition against chromosomes with similar fitness. The multitude of

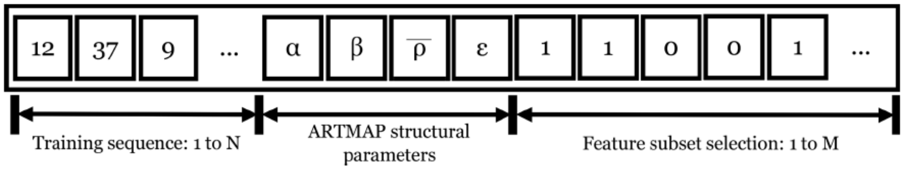
12
37
9
...
α
β
$\overline{\rho}$
ε
1
1
0
0
1
...
Training sequence: 1 to N
ARTMAP structural parameters
Feature subset selection: 1 to M

Fig. 3 A single chromosome consisting of three sections comprising all the parameters for optimizing FAMs for the given pattern classification task

communities improves global diversity by splitting the genetic convergence across multiple populations. Periodic immigration lets high-fitness chromosomes to move to a higher hierarchy and vice-versa, thus injecting new diversity into the converging populations.

A single chromosome contains all of the variables to be optimized, encoded as a single numerical string as shown in Fig. 3. In this case, a chromosome consists of three parts required to generate an initial FAM architecture and train it to recognize and classify patterns.

- Training sequence. Given a data set with N exemplars, this string is a random sequence of numbers from 1 to N representing the order in which the exemplars will be presented to the FAM during supervised learning.
- ARTMAP structural parameters α, β, ϵ, and $\bar{\rho}$, randomized according to the parameter range defined in Table 1.
- Feature subset selection. Given a data set of exemplars with M attributes, this M-length binary string decides which attribute to be excluded from the training data set. A user-defined parameter can be implemented to ensure a minimum and/or maximum limit to the number of attributes in the subset.

Figure 4 shows the flowchart of the genetic optimization process, elaborated as follows:

1. A population of random chromosomes was generated, each representing a single configuration for creating and training a FAM.
2. Each chromosome was used to generate a single trained FAM classifier.
3. Each FAM was fitness-tested using tenfold cross-validation. The training data set was divided evenly into ten folds. For a single iteration, onefold was set aside while the remaining folds were used for training the classifier. The trained classifier was then used for classifying the remaining fold. Repeat ten times, each time using a different fold for testing.
4. The fitness of the tested chromosome was computed from the mean recognition rate of the resulting trained FAM classifier.

Table 1 Variables and parameters in the FAM architecture

Variable	Description
a	Input object to be classified. Formatted as a numerical vector ranging from 0 to 1
a^C	Complement-coded vector of a. Each element in the vector is $a_i^C = 1 - a_i$
α	Choice parameter. Determines node selection for signal function T_j. Range from [0, 1], default $\alpha = 0.01$
A	Combination of original input vector a and its complement a^C
b	The true class category in which object a belongs to
β	Learning rate, or rate in which node weights were updated. Range from [0, 1], default $\beta = 1.0$ for fast learning
Γ	Fraction additive in vigilance parameter
ϵ	Match tracking parameter. Rate in which vigilance decays to baseline. Range from $[-1, +1]$, default ϵ =-0.001
J	Selected critical feature pattern. Encoded as a vector of weights w_{ij}
r	Mismatch parameter. $r = 1$ if $\rho\|A\| - \|x\| > 0$. Default $r = 0$
R	Predictive error parameter. $R = 1$ if node J makes a predictive error. Default $R = 0$
ρ	Current vigilance parameter. Range from [0, 1]
$\bar{\rho}$	Baseline vigilance parameter. Range from [0, 1]
T_j	Choice-by-difference signal function. Nodes in F_2^a are selected in order of highest to lowest signal function
x^a	Intersection between J and A: $x^a = \|A \wedge w_J\|$

5. In the first generation, chromosomes were grouped evenly into subpopulations according to similar fitness. For subsequent generations, a selected chromosome was immigrated to an adjacent subpopulation if it possessed a significantly higher or lower fitness than the average of the subpopulation.
6. Genetic selection, reproduction, and mutation were performed.
   - A number of chromosomes with the least fitness were discarded.
   - Offspring chromosomes were generated to replace discarded chromosomes. Reproduction was performed to generate an offspring which inherited the

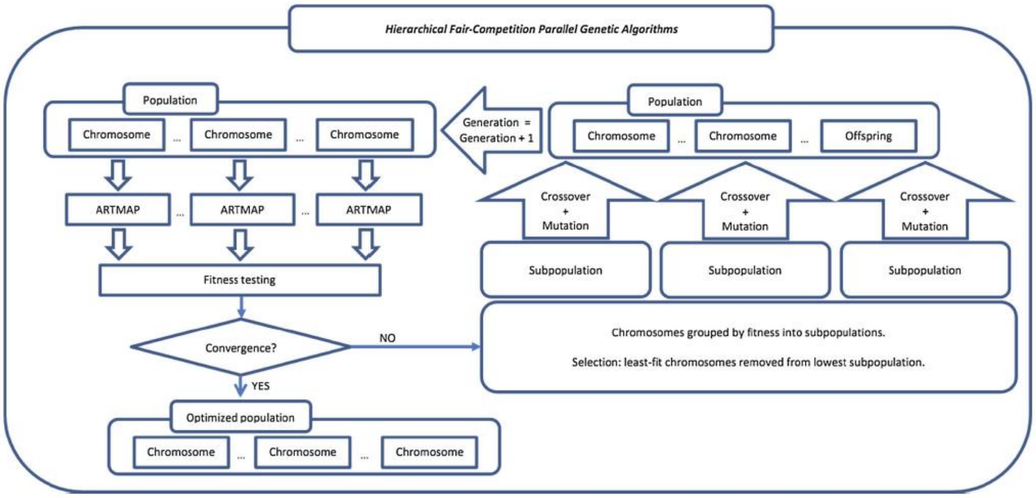

Fig. 4 Flowchart of the hierarchical parallel genetic algorithms

traits of parent chromosomes, followed by a mutation to add a random element to any selected gene in the chromosome. The reproduction and mutation operations, outlined in Fig. 5, were performed slightly differently for each segment of the chromosome.

- The newly generated offspring chromosome was added into the lowest subpopulation until they were given a chance to migrate after the next round of fitness testing.

7. The generation counter was incremented, and steps 2–5 were repeated until convergence was achieved by reaching a maximum generation limit.

The following Table 2 summarizes the key differences between the methodology of the HFCPGA and a simple GA.

2.3 Probabilistic voting

Classifier ensembles operate on the assumption that each member of the ensemble functions as an independent expert, and that combining the complementary information from every classifier, the ensemble is able to score a higher classification accuracy than any of its constituent classifiers. The decision combination method used in this study is similar the probabilistic voting system developed by Loo and Rao [30] based on the work by Lin et al. [29].

Given an ensemble with L classifiers $\{E_1, E_2, \ldots, E_L\}$ for classifying an input object X into one of k class categories $\{C_1, C_2, \ldots, C_k\}$.

$$P(E_i(X) = C(X)) = p_i \tag{7}$$

$E_i(X)$ is the class prediction of classifier E_i for the given object X, and $C(X)$ is the object's true class. Thus, each classifier has a constant recognition rate p_i to classify an object correctly. In the case of an incorrect recognition, it was assumed that all residual classes have equal probability of being selected:

$$P(E_i(X) = C_j) = (1 - p_i)/(k - 1) = e_i \tag{8}$$

where $j = 1, 2, \ldots, k$ and $C_j \neq C(X)$. Also, assuming classifier independence:

$$P(E_i(X), E_2(X), \ldots, E_L(X)|C(X) = C_j) = \prod_{i=1}^{L} P(E_i(X)|C(X) = C_j) \tag{9}$$

To minimize the error rate of the combination system, the class C_j with the largest *a posteriori* probability should be selected according to the Bayes' rule:

$$P(C(X) = C_j|E_1(X), E_2(X), \ldots, E_L(X)) = \frac{\left[\prod_{i=1}^{L} P(E_i(X)|C(X) = C_j)\right] \times P(C(X) = C_j)}{P(E_1(X), E_2(X), \ldots, E_L(X))} \tag{10}$$

$$P(E_i(X)|C(X) = C_j) = \begin{cases} p_i & \text{if } E_i(X) = C_j \\ e_i & \text{if } E_i(X) \neq C_j \end{cases} = e_i \left(\frac{p_i}{e_i}\right)^{\delta_{ij}(X)} \tag{11}$$

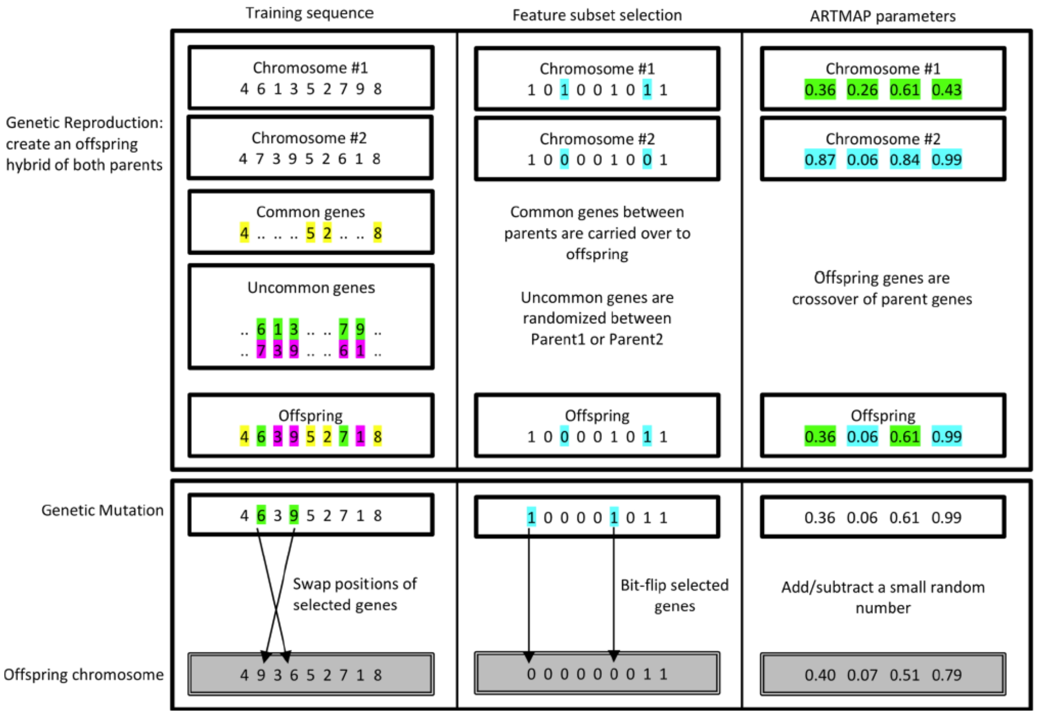

Fig. 5 Examples of genetic reproduction and mutation for the different segments of the chromosome

Table 2 Methodological differences between simple GA and HFCPGA

Simple GA	HFCPGA
Single population of chromosomes	Chromosomes divided into multiple subpopulations
All chromosomes placed in the same population	Chromosomes were grouped into subpopulations according to fitness
For reproduction, two parent	Roulette-wheel selection chooses a subpopulation
chromosomes were selected at random	Two parent chromosomes were chosen at random from the same subpopulation
No migration	Two chromosomes from different subpopulations swap positions if a high-fitness chromosome was located in a low-fitness subpopulation and vice-versa

where

$$\delta_{ij}(x) = \begin{cases} 1 & \text{if } E_i(X) = C_j \\ 0 & \text{if } E_i(X) \neq C_j \end{cases}$$

$$\begin{aligned} P(C(X) &= C_j | E_1(X), E_2(X), \ldots, E_L(X)) \\ &= \frac{\left[\prod_{i=1}^{L} e_i (\frac{p_i}{e_i})^{\delta_{ij}(X)}\right] \times P(C(X) = C_j)}{P(E_1(X), E_2(X), \ldots, E_L(X))} \end{aligned} \quad (12)$$

Let

$$Y(X) = \frac{(\prod_{i=1}^{L} e_i)}{P(E_1(X), E_2(X), \ldots, E_L(X))},$$

$$\begin{aligned} P(C(X) &= C_j | E_1(X), E_2(X), \ldots, E_L(X)) \\ &= Y(X) \left[P(C(X) = C_j) \prod_{i=1}^{L} \left(\frac{p_i}{e_i}\right)^{\delta_{ij}(X)} \right] \end{aligned} \quad (13)$$

As $Y(X)$ is the same for every class category, the effective decision function is the second part:

$$\begin{aligned} D_j(X) &= \ln P(C(X) = C_j) + \sum_{i=1}^{L} \ln\left(\frac{p_i}{e_i}\right) \delta_{ij}(X) \\ &= \ln P(C(X) = C_j) + \sum_{i=1}^{L} \ln\left(\frac{(k-1)p_i}{(1-p_i)}\right) \delta_{ij}(X) \end{aligned} \quad (14)$$

Link to Full-Text Articles :

http://link.springer.com/article/10.1007/s00521-014-1632-y