Affect classification using genetic-optimized ensembles of fuzzy ARTMAPs

Wei Shiung Liew[a], Manjeevan Seera[b,*], Chu Kiong Loo[b], Einly Lim[a]

[a] Department of Biomedical Engineering, Faculty of Engineering, University Malaya, Kuala Lumpur, Malaysia
[b] Department of Artificial Intelligence, Faculty of Computer Science and Information Technology, University Malaya, Kuala Lumpur, Malaysia

ARTICLE INFO

Article history:
Received 23 January 2014
Received in revised form 26 July 2014
Accepted 27 October 2014
Available online 11 November 2014

Keywords:
Affect recognition
Classifier ensemble
Fuzzy ARTMAP
Genetic algorithm
Parameter optimization
Supervised learning

ABSTRACT

Training neural networks in distinguishing different emotions from physiological signals frequently involves fuzzy definitions of each affective state. In addition, manual design of classification tasks often uses sub-optimum classifier parameter settings, leading to average classification performance. In this study, an attempt to create a framework for multi-layered optimization of an ensemble of classifiers to maximize the system's ability to learn and classify affect, and to minimize human involvement in setting optimum parameters for the classification system is proposed. Using fuzzy adaptive resonance theory mapping (ARTMAP) as the classifier template, genetic algorithms (GAs) were employed to perform exhaustive search for the best combination of parameter settings for individual classifier performance. Speciation was implemented using subset selection of classification data attributes, as well as using an island model genetic algorithms method. Subsequently, the generated population of optimum classifier configurations was used as candidates to form an ensemble of classifiers. Another set of GAs were used to search for the combination of classifiers that would result in the best classification ensemble accuracy. The proposed methodology was tested using two affective data sets and was able to produce relatively small ensembles of fuzzy ARTMAPs with excellent affect recognition accuracy.

1. Introduction

Affective states indicate the general psychological state of a person in response to external or internal stimulus, or social and environmental factors. While psychological states can be difficult to measure, there have been several attempts to create consistent models to cover the entire emotion spectrum. Russell's Arousal-Valence scale [1] used two measures to describe most emotional states. The Arousal axis denotes the intensity of the experienced emotion, ranging from high excitability to lethargy. The Valence axis represents a more abstract concept of the polarity or pleasure derived from the emotion. Positive valence encompasses feel-good emotions such as joy and content, whereas negative valence includes psychologically disruptive emotions such as anger, fear, and sadness, with each emotion can be represented as a range of values.

The Positive and Negative Affect Schedule (PANAS) [2] utilizes a similar concept with two-dimensional measures of Positive Affect (PA) and Negative Affect (NA) for describing several distinct emotion states. The two measures are independent domains, with PA having correlations to social activity and diurnal variations, and NA related to self-perceived stress and no circadian patterns. Robert Plutchik [3] proposed an alternate emotion model utilizing a small number of basic emotions: joy, trust, fear, surprise, sadness, disgust, anger, and anticipation. Plutchik's model conveyed the complexity of the human emotion spectrum as combinations of any two basic emotions. In practice however, humans are capable of a variety of complex, nuanced emotions that are difficult to represent using two-dimensional representation. Most experiments involving human affect thus incorporate some form of self-assessment for users to describe their own emotions. For the sake of simplicity, the self-assessment tests rely on a smallest number of affect dimensions, such as the Arousal-Valence space, to describe the most number of emotions.

Affective machines are devices designed to recognize, interpret, and in some cases, simulate human emotions. Ambulatory monitoring systems may be designed to monitor physiological and behavioural cues from human subjects and respond accordingly. For example, in a hospice environment, a system may be implemented for 24-h monitoring and trigger alerts for medical intervention if any behavioural or physiological assessment showed signs of distress. Another application involves two-way

* Corresponding author. Tel.: +603 7967 6381.
E-mail address: mseera@gmail.com (M. Seera).

interaction, using human affect to determine the responses from the machine. Negative emotions for example, teaches the machine to suppress undesirable behaviour while positive emotions reinforces the machine's current behavioural parameters for future interactions.

For a machine to recognize human emotions, it has to be able to "read" a person's emotional cues and match the pattern to a knowledge base of emotional characteristics. Many studies utilize a multimodal approach, combining information from multiple physiological sources in parallel to form a comprehensive perception of affect [4,5]. The measures typically include electroencephalogram (EEG) [6] for observing neural responses of emotion stimuli, as well as heart rate variability (HRV) [7] and galvanic skin response (GSR) [8] to observe physical signs of excitation. The features derived from recorded physiological signals were then correlated to self-assessment ratings of affect [5], subsequently using a classifier system to predict the affective state of a given set of physiological features.

The accuracy of an affect classification system is dependent on several factors: the reliability of the affect training data, and the classifier's ability to identify and learn patterns of human affect. The focus of this study is to design a methodology that would maximize a classifier's ability to learn and recognize emotions from physiological signals. The practice for classification studies was to assign default settings or parameter constraints in order to obtain results that can be compared to prior art. In the case of classification systems with highly variable results, experiments were performed multiple times and averaged to obtain a single aggregated result.

The proposed system was developed for optimizing classifier parameters and classifier fusion combination. A two-step GA was proposed, first for generating a population of optimized classifiers, and the second step for choosing the best combination of classifiers for decision-level fusion. The flow of this paper is as follows. In the next section, a literature review is first presented. This is then followed by a detailed description of the FAM neural network, GA for FAM optimization and ensemble selection, negative correlation, and probabilistic voting in Section 3. The outline of an experimental study is given in Section 4, while the results are presented in Section 5 accompanied by a discussion. Concluding remarks are finally given in Section 6.

2. Literature review

In this section, literature review on various hybrid neural networks to multiple applications is presented. A hybrid neural network based on Self Organizing Maps (SOM) and Multilayer Perceptron (MLP) network for wind speed prediction in renewable energy systems is proposed in [9]. Experimental results show the hybrid network performs better in terms of minimization of errors [9]. A novel hybrid algorithm, SOM-based initialization for hybrid training, based on two-stage learning approach is presented in [10]. A structure learning scheme which includes adding hidden neurons is first done, followed by a fuzzy neighborhood-based hybrid learning scheme to adjust the network parameters [10]. In demonstrating the approach efficiency, four simulation examples are conducted and compared with other learning methods [10].

Hybrid algorithms based on Particle Swarm Optimization (PSO) are popular. A novel hybrid optimization algorithm, with simultaneous structure of Elman-type recurrent neural networks combining advantages of discrete PSO algorithm and improved PSO algorithm is proposed in [11]. The method is evaluated on a thermal system power plant, Mackey-Glass time series and CATS time series, with results indicate the hybrid approach has better prediction accuracy and generalization performance. An automatic search methodology for parameter and performance optimization of neural networks using a hybrid evolution strategies, PSO, and concepts from genetic algorithm (GA) is proposed in [12]. Experiments were performed with results proving the proposed method is better than other methods compared in literature [12].

In detection and characterizing of acoustic signals due to surface discharge activity and hence differentiate abnormal operating conditions from the normal ones, a hybrid algorithm combining regrouping PSO with wavelet Radial Basis Function (RBF) neural network is presented in [13]. The learning method is proven to be effective by applying the wavelet RBF based on the hybrid algorithm in classification of surface discharge fault data set, with test results indicating it is an efficient method [13]. In solving classification problems, a hybrid algorithm consisting of the PSO model for RBF networks is proposed in [14]. Various benchmark classification problems are tested with experimental results showing the proposed method outperforms standard methods compared in the literature [14].

In complex problem solving in fluid dynamics, a hybrid adaptive neural network with modified adaptive smoothing errors based on GA is proposed [15]. Simulation results indicate the proposed system works fast enough and stable, and it is able to predict an incompressible viscous fluid flow [15]. Granular-oriented self-organizing hybrid fuzzy polynomial neural networks, based on MLP with context-based polynomial neurons or polynomial neurons is presented in [16]. Good results were achieved using several data sets obtained from the UCI Machine Learning Repository. An approach combining the advantages of fuzzy sets, ant-based clustering and MLP neural networks for application in breast cancer imaging is introduced in [17]. The ability to classify breast cancer images to benign or malignant is obtained from the experimental results, where it is shown that the adaptive ant-based segmentation is superior to the classical ant-based clustering technique and the performance of the hybrid system is high [17]. In improving conceptual cost estimate precision, an evolutionary fuzzy hybrid neural network is proposed in [18]. The approach integrates neural networks and higher order neural networks into a hybrid neural network [18]. Results indicate the proposed hybrid neural network can be deployed to accurately estimate cost during early stages of construction projects [18].

Various time series applications have utilized hybrid models. In detecting temporal patterns for stock market prediction tasks, the effectiveness of hybrid neural networks for time series, such as the adaptive time delay neural networks and the time delay neural networks, with the GAs is studied in [19]. GA is applied to support optimization of number of time delays, with results showing the accuracy of the integrated approach is higher than that of standard neural networks [19]. A hybrid time series neural network model for time series forecasting is tested using the monthly stream flow data at Colorado River at Lees Ferry, USA is proposed in [20]. Results from the study indicates the strengths of the hybrid model in capturing non-linear nature of complex time series, while producing more accurate forecasts [20]. In a case study to accurately forecast enrolment in University of Alabama, a hybrid fuzzy time series approach is proposed [21]. Fuzzy c-means clustering method and artificial neural networks are employed in the hybrid approach, with results showing the most accurate forecasts obtained when the proposed hybrid fuzzy time series approach is being utilized [21].

A Bayesian neural network and learned by the hybrid Monte Carlo algorithm for a short term load forecasting model is presented in [22]. Forecast of the hourly load during spring, summer, autumn, and winter were done using the hybrid algorithm, with results indicating better performances using proposed hybrid algorithm as well as solving overfitting problems [22]. A hybrid fuzzy set-based polynomial neural networks, composed of

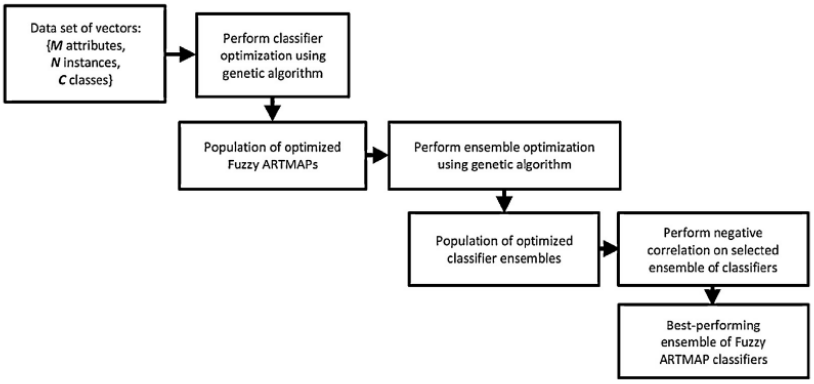
Data set of vectors: {M attributes, N instances, C classes}
Perform classifier optimization using genetic algorithm
Population of optimized Fuzzy ARTMAPs
Perform ensemble optimization using genetic algorithm
Population of optimized classifier ensembles
Perform negative correlation on selected ensemble of classifiers
Best-performing ensemble of Fuzzy ARTMAP classifiers

Fig. 1. Flowchart of the methodology to create an ensemble of optimized FAM classifiers.

heterogeneous feed-forward neural networks such as polynomial neural networks (PNNs) and fuzzy set-based PNNs is introduced in [23]. Using extensive experiments, the performance of the hybrid fuzzy neural network is quantified, and good results are achieved [23]. High order connections are developed and embedded into a back propagation network, which results in a hybrid high order neural network, applicable to linear and high order connections [24]. Two case studies were used in verifying the hybrid network performance with results showing that the hybrid network delivery better results compared to traditional back propagation network [24].

3. System framework

The proposed framework was developed as a means to generate ensembles of classifiers with consistently good classification accuracy, with minimal input from the user. For the base learner, the FAM was selected due to its popularity as well as having a significant number of literature research. The FAM was among the earliest design of ARTMAP-based neural networks, and is capable of learning patterns quickly while incorporating new knowledge incrementally without having to retrain using previous information. From the literature, a number of dependent variables that affect the performance of the FAM were identified. A search heuristic was utilized to determine the optimum combination of parameter settings to maximize the performance of the classifier. However, no classifier can hope to achieve perfect accuracy for any given problem. Dzeroski et al. [25] postulated that multiple sub-optimum classifiers can be combined into an ensemble that can outperform any single classifier. The key to an effective ensemble lies in selecting a particular combination of individuals that are sufficiently diverse to contribute complementary and conflicting information. A classifier combination system, operating on the assumption that multiple experts are less likely to make a mistake than a single expert, chooses the classification outcome that represents the consensus of the ensemble.

The design of the framework methodology thus hinges on several factors. First, a two-step searching process was required, one for optimizing the parameters of a single classifier, and the second for selecting a group of said optimized individuals to create an ensemble. GA was selected as the search heuristic for both tasks. As diversity between individuals is considered an important factor, a number of methods were applied to improve diversity. A Hierarchical Fair-Competition Parallel GA [26] was used to generate a more diverse population of FAM configurations. Feature subset selection [27] was another common method to improve diversity by training the classifier using only the partial set of training data. Finally, for creating classifier ensembles, a negative correlation method [28] was applied to identify and remove redundant classifiers. The overall framework is shown in Fig. 1.

3.1. *Fuzzy ARTMAP*

The Fuzzy ARTMAP [29] is a neural network capable of learning complex associations between multidimensional input objects and a set of discrete class labels through supervised learning. As shown in Fig. 2, a mapping field connects two ART modules, one for receiving input vectors and one for output vectors. The supervised learning method modifies the weighted connections between the input, output, and mapping fields to create a resonance between the current object in the input module and the corresponding output label. Given a set of labeled exemplars, supervised learning presents each exemplar to the FAM one at a time while the internal configuration of the weighted nodes shift incrementally in response.

The operations of the FAM neural network 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 0–1.

2 Vector a along with its complement, $a^C = 1 - a$, was encoded as a single input object A:

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

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:

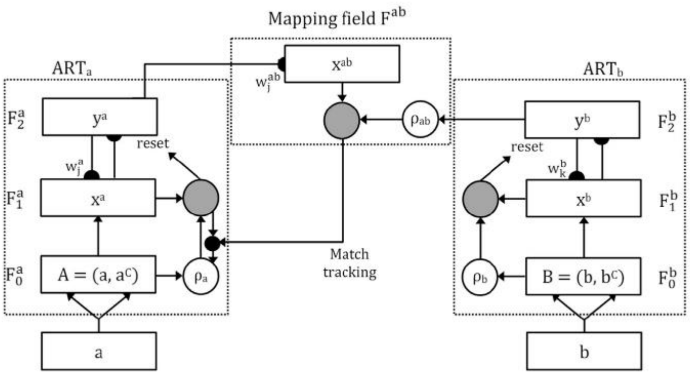

Fig. 2. Structure of the FAM neural network.

- Node J failed to meet the match criterion: $|x|/|A| < \rho_a$. Another node was chosen and Step 3 was repeated.
- Node J meets the match criterion: $|x|/|A| \geq \rho_a$. 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:

$$\frac{d\rho_a}{dt} = -(\rho - \overline{\rho}) + \Gamma R r^c \tag{5}$$

    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 $|x|/|A|$, 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:

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

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

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 [30] and ARTMAP parameter settings [31,32]. In this study, we will focus on using GA for optimizing the training order and the ARTMAP parameters, given as:

- Baseline vigilance, $\overline{\rho}$. Setting it to 0 allows a greater degree of generalization, while setting baseline vigilance close to 1 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 from the current presented pattern.
- Match tracking parameter, ϵ. Determines the rate in which current vigilance parameter returns to baseline value after each predictive error by the selected node.

3.2. Classifier optimization using genetic algorithms

GAs are search heuristics commonly used for multi-parameter optimization. Each parameter is encoded as a single gene. A collection of genes makes up a chromosome, which represents a single configuration of solutions for a given problem. In the context of this study, a chromosome represents the ARTMAP parameter values used to initialize the neural network, and the ordered subset of features of the data set to be used for training. Chromosomes were evaluated in terms of fitness, in this case, how well the neural network was able to classify patterns accurately. The GA starts off with a multitude of diverse chromosomes in a population. Over multiple generations, competitive eliminations and genetic reproduction serve to direct the evolution of the population by retaining high-fitness chromosomes to generate genetic variants as a method for exploring the solution space. When completed, the final population would consist of a number of chromosomes that yielded optimum configurations for generating the FAM.

Several studies have been performed in regards to optimizing FAMs using evolutionary algorithms. Al-Daraiseh et al. [33] proposed using GA to search for a Pareto-optimal solution between small network size and classification accuracy in order to reduce computation complexity without sacrificing performance. Palaniappan and Eswaran [30] used GA to select the optimum single-pass training sequence for a Simplified FAM. In comparison, the GA used in this study searches for combinations of parameters resulting in the highest classification rate, regardless of network size or complexity.

When developing a population of candidates for an ensemble of classifiers, two important criteria to observe are the individual classifier performance as well as the diversity of the population. Feature subset selection was implemented to improve diversity of the population, by having each classifier specializing in a narrow subset of features of the training data set. The practice of speciation using feature subset selection of training data attributes was introduced by Ho [34] and was implemented in a number of ensemble optimization studies as a measure to improve intra-population diversity [35,36].

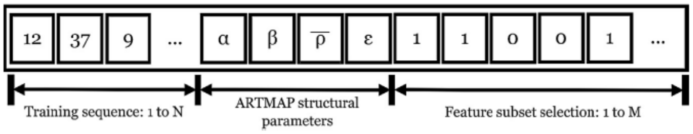
12
37
9
...
α
β
$\overline{\rho}$
ε
1
1
0
0
1
...
Training sequence: 1 to N
ARTMAP structural parameters
Feature subset selection: 1 to M

Fig. 3. Structure of a single chromosome for classifier optimization, representing a FAM constructed using the defined parameters and trained using a rearranged subset of the training data set.

In total, there are three factors to be accounted for by the FAM optimization algorithm, illustrated in Fig. 3. The parameters were:

1. Training sequence. Given a classification task with a data set consisting of N examples, the GA searches for a specific sequence of examples, which when presented to the FAM during supervised learning, will result in a trained FAM with the best classification accuracy. The training sequence was represented as a sequential series of integers from 1 to N.
2. ARTMAP structural parameters: choice parameter, learning rate, baseline vigilance, and match tracking parameter each represented as a single gene.
3. Feature subset selection. Given a data set with M attribute vectors, this section of the chromosome determines which attribute will be used for training the FAM. It was encoded as an M-length string of binary values, each representing a single attribute in the training data set, where a "0" or "1" determines which attribute to include or exclude from the training process.

A feature of GA is the tendency for the solutions in the population to converge on a single optimum point due to selection pressure [37]. A population of FAMs that are similar to each other is undesirable and unsuitable to be assembled into an ensemble. The Hierarchical Fair-Competition Parallel GA (HFCPGA) [26] mitigates the issue of convergence by implementing multiple subpopulations evolving in parallel. By restricting genetic operations to within each subpopulation, genetic convergence is spread across multiple areas.

At each generation, optimization was performed using the following steps:

1. Evaluation. Each chromosome was used to create a trained FAM classifier and tested using ten-fold crossvalidation. Fitness of the chromosome was the average recognition rate of the trained classifier.
2. Migration. For later generations, if a chromosome possessed a fitness value that was significantly higher or lower than the average fitness of the chromosomes in its current subpopulation, then it was allowed to shift to the next higher or lower subpopulation. Migration was limited to a small number per generation, so only the most radical chromosomes (i.e. largest difference in fitness between the chromosome and subpopulation average) were allowed to migrate. Migration was intended to shift chromosomes into a more competitive environment, thus allowing less-fit chromosomes to compete fairly.
3. Selection. A fraction of the total number of chromosomes was discarded from the global population, starting with the chromosomes with the worst fitness. Typically, this means that the lower subpopulations will have a higher turnover rate than the higher subpopulations.
4. Reproduction and Mutation. For each chromosome discarded in the previous step, a new chromosome was created via genetic reproduction. A roulette-wheel method was first employed to select a subpopulation. The probability for each subpopulation to be selected was inversely proportional to the average fitness of its chromosomes, thus giving higher priority to less-fit subpopulations. Two chromosomes were then selected at random from the subpopulation to reproduce and create a genetic offspring chromosome with traits from both parents.

For each offspring, a number of randomly selected genes underwent mutation to introduce a degree of randomization in the population. The number of mutated genes was defined as a fraction of the total chromosome length. The new chromosome was then added into the lowest subpopulation.

As each chromosome consisted of three different sections, reproduction was performed separately for each section, illustrated in Fig. 4.

- The training sequences for both parents were first compared for any common points, which were then carried over to the offspring. The remaining genes were distributed to the offspring while preserving the sequence as much as possible. A single mutation simply involved swapping the positions of any two random numbers within the sequence.
- For feature subset selection, logical AND was applied on each bit pair consisting of the two parent genes. However, in the case where one parent gene was '0' while the other was '1', the resultant offspring gene was randomly set to either '0' or '1'. Mutation was performed by flipping a gene.
- Each ARTMAP parameter was calculated as the average of the two parent chromosomes' parameters. Mutation was performed by adding or subtracting a small random number, no more than 10% of the maximum value of the parameter.

Two stopping criteria were introduced for ending the optimization process. The GA was terminated when a maximum number of generations have elapsed, or until the average and maximum fitness scores of the FAM population stopped improving for several consecutive generations.

3.3. *Ensemble optimization*

Given a population of optimized FAMs generated in the previous section, another GA step was used to search for the best combinations of classifiers to form an ensemble. Ensemble selection was determined by a binary string with length equal to the number of FAMs generated in the classifier optimization step. Each binary value in the string corresponded to one FAM, with '0' meaning the classifier was not selected, and '1' means that the classifier was selected for the ensemble. Fitness of a single chromosome was determined by the negative correlation index in Eq. (9). As the chromosomes were binary encoded, all genetic reproduction and mutation performed were similar to the methods outlined in the previous section for feature subset selection.

Classifier fusion was performed using two methods. The negative correlation method was employed to determine whether to accept or reject a classifier into the ensemble based on its contribution towards ensemble accuracy and/or diversity. Decision fusion for the multitude of ensemble members was decided using a probabilistic voting strategy.

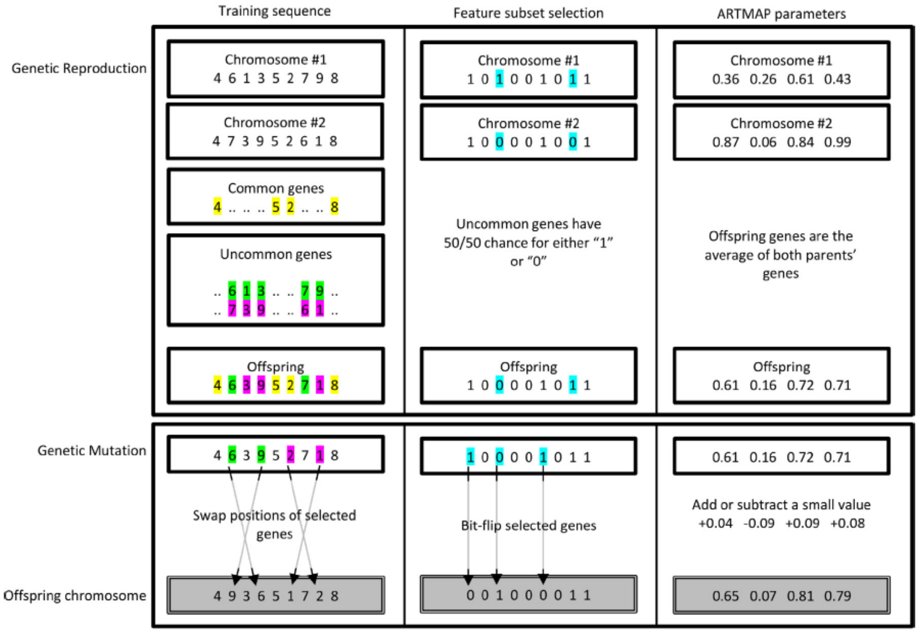

Fig. 4. Example of genetic reproduction and mutation for each section of the chromosome for FAM optimization.

3.3.1. Negative correlation

The GA optimization method proposed a group of FAMs to form an ensemble with no regard for correlation between individual FAMs. This may result in large ensembles with redundant FAMs. A negative correlation method [38] was used for building the ensemble by selective recruitment of FAMs.

Assuming a set of data pairs was given:

$$D = \{(x_n, c_n) | n = 1, \ldots, N\} \tag{7}$$

where x_n is an input vector and $c_n \in \{1, \ldots, C\}$ is its corresponding class label. Given an ensemble with J classifiers, the kth output for a given input object x_n is computed as:

$$\hat{f}^k(x_n) = \frac{1}{J} \sum_{j=1}^{J} f_j^k(x_n) \tag{8}$$

where $f_j = (f_j^1, \ldots, f_j^C)$ and $f_j^k \rightarrow [0, 1]$. Now, to define the generalization error for the jth neural network component

$$E_j = \sum_{n=1}^{N} \sum_{k=1}^{C} [(f_j^k(x_n) - y_n^k)^2 - \lambda(\hat{f}^k(x_n) - f_j^k(x_n))^2] \tag{9}$$

and the generalization error for the ensemble

$$\hat{E}_J = \sum_{n=1}^{N} \sum_{k=1}^{C} [(\hat{f}^k(x_n) - y_n^k)^2 - \lambda \sum_{r=1}^{J} (\hat{f}^k(x_n) - f_r^k(x_n))^2] \tag{10}$$

where λ is a controllable variable between the error and penalty terms, and $y_n = (y_n^1, \ldots, y_n^C)$ is a one-of-C representation of c_n:

$$y_n^k = \begin{cases} 1 & \text{if} \quad k = c_n \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

Eqs. (9) and (10) each states that the generalization error consists of the misclassification rate of the neural network's output, while the second part compares the similarity between the outputs of individual classifiers. The λ parameter controls the severity of the penalty when the compared classifiers are similar to each other in terms of classifier output. Substituting (10) into (12) yields the overall ensemble error which can be summarized as:

$$\hat{E}_J = (\frac{1}{J})^2 \sum_{j=1}^{J} \sum_{t=1}^{J} [C_{jt} - \lambda K_{jt}] \tag{12}$$

where the misclassification term is

$$C_{jt} = \sum_{n=1}^{N} \sum_{k=1}^{C} [(f_j^k(x_n) - y_n^k)(f_t^k(x_n) - y_n^k)] \tag{13}$$

and the similarity penalty is

$$K_{jt} = \sum_{n=1}^{N} \sum_{k=1}^{C} \sum_{r=1}^{J} [(f_j^k(x_n) - f_r^k(x_n)) \times (f_t^k(x_n) - f_r^k(x_n))] \tag{14}$$

An ensemble of FAMs was constructed using the following method:

1. Given a single chromosome representing a combination of selected FAMs, the classifier with the highest recognition rate was selected as the first member of the ensemble. The generalization error of the ensemble at that point was computed as $\hat{E}_1$.
2. When another FAM was added, the classification output of the ensemble was modified according to new information provided. The generalization error of the new ensemble was recalculated, $\hat{E}_2$.

Link to Full-Text Articles :

http://www.sciencedirect.com/science/article/pii/S1568494614005444