2014 IEEE Congress on Evolutionary Computation, CEC-2014, in WCCI2014, Beijing, China, 6-11 July, 2014.

A Novel Genetic Algorithm Approach for Simultaneous Feature and Classifier Selection in Multi Classifier System

Tien Thanh Nguyen, Alan Wee-Chung Liew
School of Information and Communication Technology
Griffith University
Gold Coast, QLD, Australia
tienthanh.nguyen2@griffithuni.edu.au,
a.liew@griffith.edu.au

Xuan Cuong Pham
School of Information and Communication Technology
Hanoi University of Science and Technology
Hanoi, Vietnam
cuong.phamxuan.hust@gmail.com

Minh Toan Tran
School of Applied Mathematics and Informatics
Hanoi University of Science and Technology
Hanoi, Vietnam

Mai Phuong Nguyen
College of Business, Massey University
Albany, New Zealand

***Abstract*—In this paper we introduce a novel approach for classifier and feature selection in a multi-classifier system using Genetic Algorithm (GA). Specifically, we propose a 2-part structure for each chromosome in which the first part is encoding for classifier and the second part is encoding for feature. Our structure is simple in the implementation of the crossover as well as the mutation stage of GA. We also study 8 different fitness functions for our GA algorithm to explore the optimal fitness functions for our model. Experiments are conducted on both 14 UCI Machine Learning Repository and CLEF2009 medical image database to demonstrate the benefit of our model on reducing classification error rate.**

Keywords—Genetic Algorithm, multi-classifier system, classifier fusion, classifier selection, feature selection, combining rules, combining classifiers algorithm.

I. INTRODUCTION

In recent years, ensemble learning is one of the most active research areas in supervised learning [1, 2]. Ensemble methods can be divided into two categories [2]:

- Mixture of experts: Using a fixed set of classifiers and making decision from outputs of these classifiers.
- Coverage: Generating generic classifiers, which are classifiers from the same family but have different parameters. The classifiers are combined to reach a final decision.

In this paper, we focus on the first type of ensemble methods where decision is formed by combining outputs of different base classifiers. There are several combining strategies and among them, Stacking-based approaches are one of the most popular ensemble methods. Stacking was first proposed by Wolpert [3] and was further developed by Ting and Witten [7]. In this model, the training set is divided into several equal disjoint parts. One part plays as the test set in turn and the rest play as the training set during training. The outputs of Stacking are posterior probabilities that observations belong to a class according to each base classifier. Posterior probabilities of all observations form meta-data or Level1 data. The original data is denoted as Level0 data.

We denote class set by $\{W_j\}$, N as the number of observations, K as the number of base classifiers and M as the number of classes. For an observation X_i, $P_k(\mathrm{W}_j \mid X_i)$ is the probability that X_i belongs to class W_j given by k^{th} classifier. Level1 data of all observations, a $N \times MK$ posterior probability matrix $\{P_k(\mathrm{W}_j \mid X_i)\}$ $j=\overline{1,M}$ $k=\overline{1,K}$ $i=\overline{1,N}$ is in the form:

$$\begin{bmatrix} P_1(\mathrm{W}_1 \mid X_1)... & P_1(\mathrm{W}_M \mid X_1)... & P_K(\mathrm{W}_1 \mid X_1)...P_K(\mathrm{W}_M \mid X_1) \\ P_1(\mathrm{W}_1 \mid X_2)... & P_1(\mathrm{W}_M \mid X_2)... & P_K(\mathrm{W}_1 \mid X_2)...P_K(\mathrm{W}_M \mid X_2) \\ \dots & & \dots \\ P_1(\mathrm{W}_1 \mid X_N)... & P_1(\mathrm{W}_M \mid X_N)... & P_K(\mathrm{W}_1 \mid X_N)...P_K(\mathrm{W}_M \mid X_N) \end{bmatrix} \quad (1)$$

Level1 data of an observation X is defined as:

$$Level1(X) := \begin{bmatrix} P_1(\mathrm{W}_1 \mid X) & \dots & P_1(\mathrm{W}_M \mid X) \\ \vdots & \ddots & \vdots \\ P_K(\mathrm{W}_1 \mid X) & \cdots & P_K(\mathrm{W}_M \mid X) \end{bmatrix} \quad (2)$$

GA model for multi-classifier system to improve classification performance has been shown to be an effective strategy [6, 10, 11]. Our aim here is to solve both the classifier selection and feature selection problem. Feature selection is an important problem in pattern recognition, data analysis and data mining [12]. Feature selection tries to reduce the number of features while maintaining accuracy at an acceptable value. Generally speaking, methods that transform features to a new domain with a reduction in the dimension of feature can be treated as feature selection. Therefore, strategy to solve this problem is very diverse, for instance, linear transformations, search techniques and GA. Classifier selection is an interesting topic in classifier fusion. The presence of some classifiers may actually degrade the accuracy of the system because of their wrong predictions. So removing them would reduce predicted error rate. Here, we propose a GA approach to not only search for the best feature subset but also explore the best classifier subset in a multi classifier system. Another aspect of our work is that we also empirically evaluate 8 different fitness functions for combining classifiers. As there are many state-of-the-art combining algorithms for classifier fusion, it is important to assess their performance. We conduct extensive experiments

on the UCI data files and CLEF 2009 medical image database to demonstrate the performance of our approach.

II. RECENT WORK

Based on Stacking model, a number of combining algorithms had been introduced with the purpose of reducing the error rate of classification task. Ting and Witten [7] proposed Multiple Response Linear Regression algorithm (MLR) to combine posterior probabilities of each observation based on the sum of weights calculated from K Linear Regression functions. Kuncheva et al. [4] applied Fuzzy Relation to find the relationship between posterior probability matrix of (2) and Decision Template for each class. These methods are trainable combining algorithms since Level1 data is used in training to form the prediction framework.

Besides, Kittler et al. [8] presented six fixed combining rules named Sum, Product, Vote, Min, Max and Average. These rules are simple in calculation and in several applications they give lower classifying error rate compared with that of base classifiers. The advantage of applying fixed rules for ensemble system is that no training based on Level1 data is needed.

On the other hand, GA approaches to improve accuracy of classifier fusion have also been proposed recently. Kuncheva and Jain [6] introduced two GA based algorithms in which features are selected by join and disjoin mechanism. In the former, features were encoded by $\{0,1,...,K\}$ where k $(1 \le k \le K)$ means that feature is only used by k^{th} classifier and 0 means that feature is not used by any classifiers. In the latter, classifier encoding was added in the same chromosome with feature encoding and both of them work independently in crossover and mutation stage. An encoding method was developed based on Venn diagram for feature encoding and integer values for classifier encoding. The first algorithm did not perform well according to their experiment while the second algorithm is quite hard to implement since Venn diagram becomes more complicate with many classifiers. Nanni et al. [10] employed GA to improve the SCANN algorithm [9] by building representations where each includes encoding of M classes. Gabrys and Ruta [11] tried to put classifier, feature and fixed rule encoding in a single chromosome as 3-dimensional cube. However, these two approaches are difficult to implement because of the complicated crossover stage. We address these issues with a new GA model for multi classifier system which achieves both effectiveness and easy implementation.

III. PROPOSED GA APPROACH FOR MULTI CLASSIFIER SYSTEM

A. Chromosome Encoding and Crossover

To implement GA for classifier fusion, first, we propose the structure of a chromosome as illustrated in Fig. 1. Each chromosome has two parts. The first part includes K genes based on the number of classifiers. We use two elements $\{0,1\}$ to encode for each gene in a chromosome in which:

$$Gene(k) = \begin{cases} 1 & \text{if } k^{th} \text{ classifier is selected} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

The second part is K feature encodings associated with the K classifiers. Each of them includes D genes according to the number of features. We again use $\{0,1\}$ to encode for a gene of feature encoding. For $k = \overline{1,K}$ and $d = \overline{1,D}$ we have:

$$Gene_k(d) = \begin{cases} 1 & \text{if } d^{th} \text{ feature is selected by } k^{th} \text{ classifier} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

Next, we discuss the 2-stage crossover process on two parents A and B. In the first stage, crossover is conducted on classifier encoding part. Here we employ single point splitter. Each classifier encoding exchanges its head with the other while retains its tail, and their feature encodings are swapped accordingly based on the classifier encoding on the first part (Fig. 2). This ensures the consistency between first part and second part of each chromosome. In the second stage, single point splitter is again applied since crossover is continued to be applied on feature encoding of i^{th} classifier of both A and B (Fig. 3) ($i = \overline{1,K}$). After crossover is performed on all pair of feature encodings, we have 2 new offspring chromosomes.

As GA search does not always resulted in better accuracy than the model on the supersets, we add a special chromosome to initialize the population in which all genes of this element are equal to 1. This encoding will be evaluated based on fitness value and replaced if its accuracy is inferior or retained if its error rate is competitive. In doing so, we ensure that our GA approach is always better than or equal to the original superset model.

Algorithm 1: Crossover
Input: Chromosome A and Chromosome B
Output: Two new offspring chromosomes

Step 1: Crossover on classifier encoding part of A and B by single point splitter as in Fig. 2
Step 2: Crossover each pair of feature encoding on second part of A and B respectively by single point splitter (Fig. 3)

B. Fitness Function

We introduce 8 fitness functions for our GA approach. The accuracy of the combining classifiers algorithm is use as fitness in evaluations. Our purpose is to explore which algorithm is suitable for our model. First, six fixed combining rules [5, 7], namely Sum, Product, Vote, Max, Min and Median, on Level1 data of unlabeled observation $XTest$ (2) are selected to predict class label. The details of the six fixed rules are given by:

- Sum rule:

$$XTest \in \mathrm{W}_t \text{ if } t = \arg\max_{m=1,M} \left\{ \sum_{k=1}^{K} P_k(\mathrm{W}_m \mid XTest) \right\} \quad (5)$$

- Product rule:

$$XTest \in \mathrm{W}_t \text{ if } t = \arg\max_{m=1,M} \left\{ \prod_{k=1}^{K} P_k(\mathrm{W}_m \mid XTest) \right\} \quad (6)$$

- Vote rule:

$$XTest \in \mathrm{W}_t \text{ if } t = \arg\max_{m=1,M} \left(\sum_{k=1}^{K} \Delta_{km} \right) \quad (7)$$

$$\Delta_{kj} = \begin{cases} 1 & if \quad j = \arg\max\limits_{m=1,M} P_k(W_m \mid XTest) \\ 0 & otherwise \end{cases}$$

- Max rule

$$XTest \in \mathrm{W}_t \text{ if } t = \arg\max_{m=1,M}\left\{\max_{k=1,K}\left\{P_k(\mathrm{W}_m \mid XTest)\right\}\right\} \quad (8)$$

- Min rule

$$XTest \in \mathrm{W}_t \text{ if } t = \arg\max_{m=1,M}\left\{\min_{k=1,K}\left\{P_k(\mathrm{W}_m \mid XTest)\right\}\right\} \quad (9)$$

- Median rule

$$XTest \in \mathrm{W}_t \text{ if } t = \arg\max_{m=1,M}\left\{median_{k=1,K}\left\{P_k(\mathrm{W}_m \mid XTest)\right\}\right\} \quad (10)$$

We also employ MLR algorithm [7] as fitness function. The idea of MLR is that each classifier put a different weight on each class and then combining algorithm is conducted based on posterior probability and its associated weight. Ting et al. [7] proposed solving M Linear Regression model corresponding with M class based on Level1 data (1) of training set to find these combining weights (Fig. 4). Here we denote the weight matrix by $\mathbf{\Psi} = \{\omega_{ij}\}$ in which ω_{ij} is the weight of i^{th} classifier on j^{th} class.

The regression model for m^{th} class is given by:

$$LR_m(X) = \sum_{k=1}^{K} \omega_{km} P_k(W_m \mid X) \quad (11)$$

and the predicted label of an observation $XTest$ is given by:

$$XTest \in W_t \quad \text{if} \quad t = \arg\max_{m=1,M}\{LR_m(XTest)\} \quad (12)$$

To find $\mathbf{\Psi} = \{\omega_{ij}\}$, we solve M Linear Regression model independently by minimizing M objective functions:

$$\sum_i (Y_m(X_i) - LR_m(X_i))^2 \to \min \quad (12)$$

where X_i is an observation in training set and Y_m $(m = \overline{1,M})$ is a crisp label vector of X_i given by

$$Y_m(X_i) = \begin{cases} 1 \ if \quad X_i \in \text{ class } \mathrm{W}_m \\ 0 \ otherwise \end{cases} \quad (13)$$

Finally, we also evaluate the Decision Template combining algorithm. Decision Template of i^{th} class as a $K \times M$ matrix defined in [4] is given by:

$$DT_i(k,m)_{k=\overline{1,K};\, m=\overline{1,M}} = \frac{\sum_{j=1}^{N} \mathbf{I}[Y_j = W_i] \times P_k(W_m \mid X_j)}{\sum_{j=1}^{N} \mathbf{I}[Y_j = W_i]} \quad (14)$$

where Y_j is the crisp label of observation X_j and

$$\mathbf{I}[Y_j = W_i] = \begin{cases} 1 & \text{if} \quad Y_j = W_i \\ 0 & \text{otherwise} \end{cases} \quad (15)$$

The Decision Profile of an observation X (denoted by $DP(X)$) is also defined as similar to Level1(X) (2). Actually, the Decision Template of i^{th} class is the average of Decision Profile of observations in training set where their labels are W_i.

Eleven measurements between $DP(X)$ and DT_i was proposed in [4]. Here we employ a similarity measurement S defined by:

$$S(DP(X), DT_i) = \frac{\|DP(X) \cap DT_i\|}{\|DP(X) \cup DT_i\|} \quad (16)$$

where $\|\alpha\|$ is the relative cardinality of the fuzzy set α. The class label for an observation $XTest$ is predicted by:

$$XTest \in W_t \quad \text{if} \quad t = \arg\max_{i=1,M}\{S(DP(X), DT_i(X))\} \quad (17)$$

The pseudo code of our GA algorithm for classifier fusion is given by:

Algorithm 2: Compute fitness value based on a specific chromosome encoding
Input: Training set Level0 data, chromosome c and combining classifier algorithm.
Output: fitness value for c

Step 1: Use Stacking to generate Level1 data from Level0 data by selected classifiers and their associated features based on encoding of c
Step 2: Run combining classifiers algorithm on Level1 data, return accuracy as fitness value.

Algorithm 3: GA approach for multi-classifier system
Training process:
Input: Training set Level0, K base classifiers, PMul: mutation probability, TMax: maximum number of generations, L: population size, combining classifiers algorithm
Output: selected subsets of classifiers and features and classification model corresponding with optimal solution.

Step 1: Initialize population with L chromosomes;
Step 2: For each chromosome, call Algorithm 2 to o btain fitness value
Step 3: Do

- Withdraw with replacement to generate L/2 pair of chromosome.
- Call Algorithm 1 to conduct crossover for each pair
- Perform mutation based on PMul
- Add new L offspring chromosomes to population
- Compute fitness function of all L offspring chromosomes (Algorithm 2), select L chromosomes with highest fitness

```
      values.
While(converge=true)  or  (the  number  of
generations > TMax)
Step 4: Choose optimal chromosome from final
population and generate classification model
corresponding with it.

Test process:
Input: Unlabeled observation XTest
Output: predicted label for XTest

Compute Level1 data of XTest (2) and then
predict its label based on classification
model obtained in training process.
```

IV. EXPERIMENTAL RESULTS

We conducted experiments on data from the UCI Machine Learning Repository data files and CLEF 2009 medical image dataset. For evaluation, 10-fold cross validation was performed on each dataset and the test was run 10 times. So in total we had 100 test outcomes.

To compute fitness value, as mentioned earlier we used 6 fixed combining rules (denoted by GA Sum, GA Product, GA Max, GA Min, GA Median and GA Majority Vote) and 2 well-known combining classifiers methods (denoted by GA Decision Template and GA MLR). Error rate of these approaches were compared. To initialize the parameters of GA, we set the mutation probability PMul=0.015, population size L=20, maximum number of generations TMax=50. We selected Linear Discriminant Analysis, Naïve Bayes and K Nearest Neighbor (with K set to 5, denoted as 5-NN) as our base classifiers. As these classifiers are different to each other in their approach, the diversity of the ensemble system is ensured. To assess statistical significance, we used paired t-test to compare two classification results (level of significance set to 0.05).

A. UCI Data

We chose 14 UCI files as input data for combining classifier system (Table I). Experimental results of all files are reported in Table II and III.

As can be seen, GA approach outperforms its counterparts with the same combining algorithm. For example, GAs on 5 fixed rules, namely Product, Max, Min, Median and Vote, all post more than 10 wins and 0 loss compared with the classical combining strategy. The results of GA on Sum Rule, Decision Template and MLR also show good performance although they are less remarkable than those of GA on the others (Table IV).

Next, we compare each GA approach with the best result from all GA approaches (called Select Best). Our objective is to discover which fitness function is best among the data. Table V illustrates result of statistical test in which GA Product achieves the best result, obtaining only 3 losses. GA Median is follows with 4 losses, and next is GA Sum, GA MLR and GA Max with 6 losses. GA Decision Template is the worst approach with up to 9 losses in comparison with Select Best.

In reality, it is difficult to choose a best GA approach for all data files. One may perform well on some files but poor on others. For example, GA Decision Template is the worse in our experiment but it obtains good result on Balance (8.32%). Based on our experiments, we suggest using fixed rules as fitness function for our GA approach since fixed rules are very simple in implementation. We only need to compute posterior probabilities of unlabeled observation to form the prediction; as a result, computation cost will be reduced. However, when GA with fixed rules do not achieve acceptable results, other complex combining classifier algorithms like Decision Template and MRL can be employed.

Another advantage of our GA approach is that it helps to select optimal subsets of classifiers and features which produce better result than their supersets. This is due to the fact that the proposed GA approach solves both classifier selection and feature selection problems by using a 2-part structure for chromosomes, the first for classifier encoding and the second for feature encoding.

B. CLEF 2009

The other experiment was evaluated on CLEF2009, a medical image database collected by Archen University, Germany. It is a large database containing 15363 images allocated in 193 hierarchical categories. Here, we chose 10 classes with different number of observations in each (Table VI). Histogram of Local Binary Pattern (HLBP) [14] was selected as feature vector of each image.

Once again, the benefit of our approach is evident from the results in Table VII and VIII. The reduction of error rate is significant on GA with fixed rules, for example, GA Min helps to reduce error rate by 9.63%. Error rate of GA MLR decreases by 4% compared with MLR while the decrease for GA Decision Template is 1.5%.

V. CONCLUSION AND FUTURE WORK

In this paper, we have introduced a method to solve both the feature and classifier selection problems in a multi-classifier system based on GA. Our aim is to find subset of classifier and feature set which have more discrimination ability and to search the optimal subset from base classifiers to achieve lower error rate than the original sets. We have conducted extensive experiments and compare our GA approach among 8 different fitness functions. Experimental results on UCI files and CLEF2009 dataset has demonstrated that our model considerably reduces error rates of classification by exploring optimal subsets of the original feature and classifier set. We suggest using fixed rules as fitness function for our model because it helps to save computation cost while achieving acceptable accuracy.

In the future, we plan to explore new fitness functions for our GA model as well as discover new effective encoding to boost GA performance.

REFERENCE

[1] L. Rokach, “Taxonomy for characterizing ensemble methods in classification tasks: A review and annotated bibliography”, Journal of Computational Statistics & Data Analysis, vol. 53, issue 12, pp. 4046 - 4072, October, 2009.

[2] T. K. H, J. J. Hull. and S.N. Srihari, “Decision Combination in Multiple Classifier Systems”, IEEE Transactions on Pattern Analysis and Machine Intelligent, vol. 16, no. 1, pp. 66-75, 1994.

[3] D. H. Wolpert, “Stacked Generalization”, Neural Networks, vol. 5, no. 2, pp. 241-259, Pergamon Press, 1992.

[4] L. I. Kuncheva, J. C. Bezdek and R. P. W. Duin, “Decision Templates for Multi Classifier Fusion: An Experimental Comparison”, Pattern Recognition, vol. 34, no. 2, pp.299-314, 2001.

[5] L. I. Kuncheva, “A theoretical Study on Six Classifier Fusion Strategies”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 2, February 2002.

[6] L. I. Kuncheva and L. C. Jain, "Designing Classifier Fusion Systems by Genetic Algorithms", IEEE Transactions on Evolution Computation, vol.4, no. 4, September 2000.

[7] K. M. Ting, I. H. Witten, "Issues in Stacked Generation", Journal of Artificial In Intelligence Research, vol 10, pp. 271-289, 1999.

[8] J. Kittler, M. Hatef, R. P. W. Duin and J. Matas, "On Combining Classifiers", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 3, March 1998.

[9] C. Merz, "Using Correspondence Analysis to Combine Classifiers", Machine Learning, vol. 36, pp. 33-58, 1999.

[10] L. Nanni and A. Lumini, "A genetic encoding approach for learning methods for combining classifiers", Expert Systems with Applications, vol. 36, pp. 7510-7514, 2009.

[11] B. Gabrys and D. Ruta, "Genetic Algorithms in classifier fusion", Applied Soft Computing, vol. 6, pp. 337-347, 2006.

[12] M. L. Raymer, W. F. Punch, E. D. Goodman, L. A. Kuhn and A. K. Jain, "Dimensionality Reduction using Genetic Algorithms", IEEE Transactions on Evolutionary Computation, vol. 4, no. 2, July 2000.

[13] A. K. Seeward, "How to Make Stacking Better and Faster While Also Taking Care of an Unknown Weakness", Proceedings of the Nineteenth International Conference on Machine Learning, pp. 554-561, 2002.

[14] B. C. Ko, S. H. Kim and J. Y. Nam, "X-ray Image Classification Using Random Forests with Local Wavelet-Based CS-Local Binary Pattern", J Digital Imaging, vol. 24, pp. 1141-1151, 2011.

[15] M. U. Sen, H. Erdogan, "Linear clssifiers combination and selection using group sparse regulation and hingle loss", Pattern Recognition Letter, vol. 34, pp. 265-274, 2013.

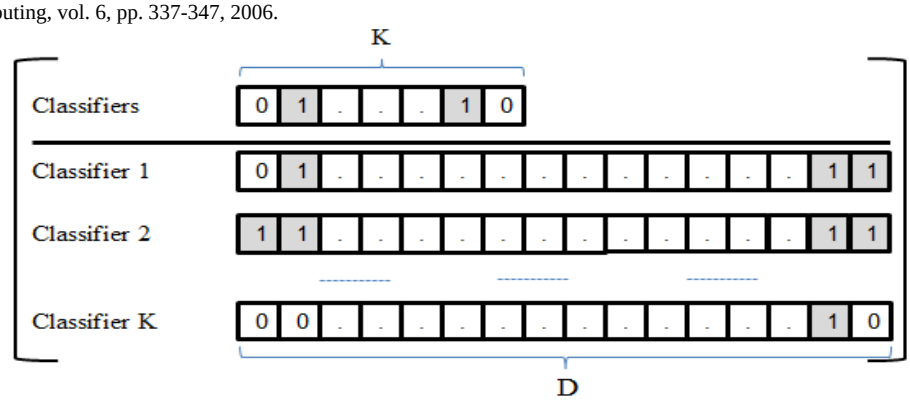


Fig. 1. Structure of a proposed chromosome

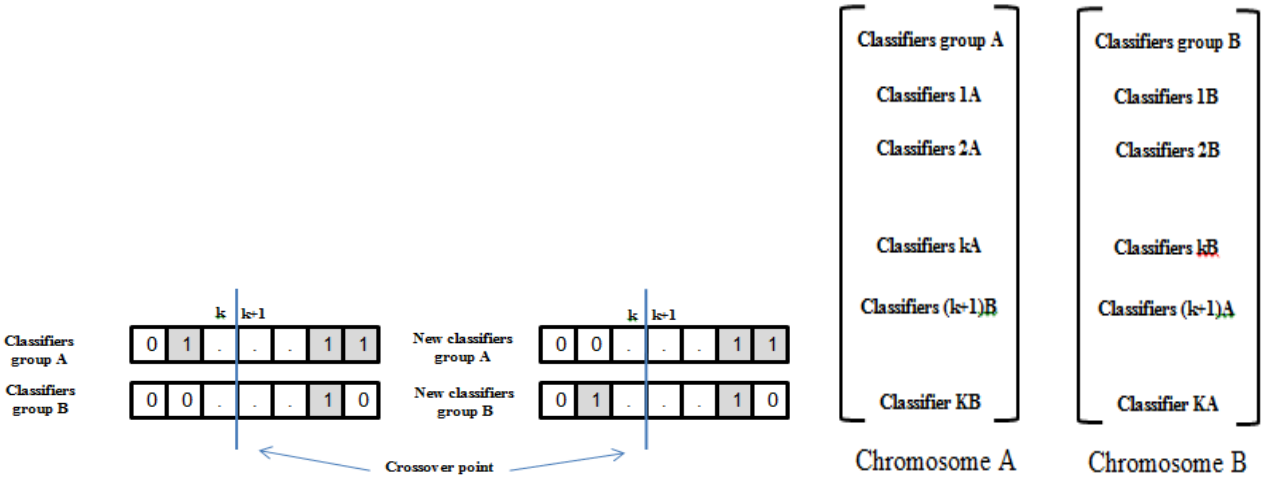


Fig. 2. Performing crossover on the first part and swapping second part accordingly

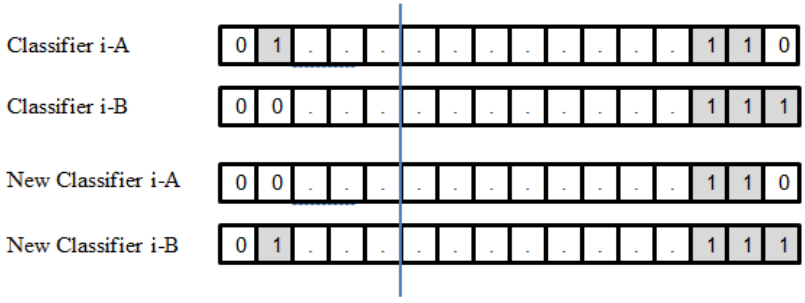


Fig. 3. Crossover on feature encoding of two classifiers

	Class 1	Class 2	…	Class j	...	Class M

Classifier 1	$P_1(W_1 \mid X)\ \omega_{11}$	$P_1(W_2 \mid X)\ \omega_{12}$	…	$P_1(W_j \mid X)\ \omega_{1j}$	…	$P_1(W_M \mid X)\ \omega_{1M}$
Classifier 2	$P_2(W_1 \mid X)\ \omega_{21}$	$P_2(W_2 \mid X)\ \omega_{22}$	…	$P_2(W_j \mid X)\ \omega_{2j}$	…	$P_2(W_M \mid X)\ \omega_{2M}$
…	…	…	…	…	…	…
Classifier i	$P_i(W_1 \mid X)\ \omega_{i1}$	$P_i(W_2 \mid X)\ \omega_{i2}$	…	$P_i(W_j \mid X)\ \omega_{ij}$	…	$P_i(W_M \mid X)\ \omega_{iM}$
…	…	…	…	…	…	…
Classifier K	$P_K(W_1 \mid X)\ \omega_{K1}$	$P_K(W_2 \mid X)\ \omega_{K1}$		$P_K(W_j \mid X)\ \omega_{Kj}$		$P_K(W_M \mid X)\ \omega_{KM}$
	⇩	⇩	…	⇩	…	⇩
	LR_1	LR_2		LR_j		LR_M

Fig. 4. MLR method on Level1 data

TABLE I. INFORMATION OF UCI FILES IN EXPERIMENTS

Number	File name	# of attributes	Attribute Type	# of observation	# of class	# of attribute on Level1 (3 classifiers)
1	Bupa	6	C,I,R	345	2	6
2	Artificial	10	R	700	2	6
3	Pima	6	R,I	768	2	6
4	Sonar	60	R	208	2	6
5	Heart	13	C,I,R	270	2	6
6	Haberman	3	I	306	2	6
7	Balance	4	C	625	3	9
8	Fertility	9	R	100	2	6
9	Wdbc	30	R	569	2	6
10	Australian	14	C,I,R	690	2	6
11	Tae	20	C,I	151	2	6
12	Contraceptive	9	C,I	1473	3	6
13	Vehicle	18	I	946	4	12
14	Iris	4	R	150	3	9

TABLE II. CLASSIFYING RESULTS ON UCI FILES BY 6 FIXED RULES, DECISION TEMPLATE AND MLR

File name	Sum rule		Product rule		Max rule		Min rule		Median rule		Majority vote		Decision Template		MLR	
	Mean	*Variance*	*Mean*	*Variance*	*Mean*	*Variance*	*Mean*	*Variance*	*Mean*	*Variance*	*Mean*	*Variance*	*Mean*	*Variance*	*Mean*	*Variance*
Bupa	0.3028	4.26E-03	0.3021	4.12E-03	0.2986	4.15E-03	0.2970	4.89E-03	0.3428	4.46E-03	0.3429	4.04E-03	0.3348	7.10E-03	0.3033	4.70E-03
Artificial	0.2230	2.06E-03	0.2193	2.05E-03	0.2450	2.57E-03	0.2453	2.90E-03	0.3089	1.36E-03	0.3073	1.03E-03	0.2433	1.60E-03	0.2426	2.20E-03
Pima	0.2405	1.62E-03	0.2419	1.63E-03	0.2411	1.69E-03	0.2449	2.02E-03	0.2376	1.69E-03	0.2365	2.10E-03	0.2482	2.00E-03	0.2432	2.30E-03
Sonar	0.2259	9.55E-03	0.2285	9.81E-03	0.2260	7.01E-03	0.2298	9.32E-03	0.2104	1.00E-02	0.2079	8.16E-03	0.2129	8.80E-03	0.1974	7.20E-03
Heart	0.1637	4.59E-03	0.1648	5.20E-03	0.1730	4.14E-03	0.1700	4.01E-03	0.1570	4.64E-03	0.1604	3.87E-03	0.1541	4.00E-03	0.1607	4.70E-03
Haberman	0.2392	2.39E-03	0.2424	3.08E-03	0.2457	3.18E-03	0.2461	2.47E-03	0.2524	1.67E-03	0.2504	1.76E-03	0.2779	5.00E-03	0.2428	3.30E-03
Balance	0.1113	5.55E-04	0.1131	4.95E-04	0.1112	4.82E-04	0.1232	4.99E-04	0.1155	4.93E-04	0.1261	4.63E-04	0.0988	1.40E-03	0.1225	8.00E-04
Fertility	0.1290	2.46E-03	0.1290	2.26E-03	0.1270	1.97E-03	0.1280	2.02E-03	0.1330	2.81E-03	0.1310	2.34E-03	0.452	3.41E-02	0.1250	2.28E-03
Wdbc	0.0401	7.07E-04	0.0517	8.19E-04	0.0485	8.03E-04	0.0522	7.71E-04	0.0395	5.03E-04	0.0406	6.47E-04	0.0385	5.00E-04	0.0399	7.00E-04
Australian	0.1281	1.78E-03	0.1594	1.91E-03	0.1604	1.95E-03	0.1609	1.80E-03	0.1270	1.56E-03	0.1262	1.37E-03	0.1346	1.50E-03	0.1268	1.80E-03
Tae	0.4625	1.36E-02	0.4622	1.14E-02	0.5191	1.11E-02	0.4868	1.40E-02	0.4443	1.46E-02	0.4435	1.70E-02	0.4643	1.21E-02	0.4652	1.24E-02
Contraceptive	0.4653	1.79E-03	0.4667	1.19E-03	0.4734	1.19E-03	0.4766	1.77E-03	0.4803	1.31E-03	0.4844	1.27E-03	0.4781	1.40E-03	0.4675	1.10E-03
Vehicle	0.2671	1.38E-03	0.2645	1.37E-03	0.2937	1.54E-03	0.2737	1.57E-03	0.2858	1.57E-03	0.3194	2.01E-03	0.2161	1.50E-03	0.2139	1.40E-03
Iris	0.0387	2.59E-03	0.0407	2.39E-03	0.0440	3.13E-03	0.0413	2.56E-03	0.0333	1.64E-03	0.0327	1.73E-03	0.04	2.50E-03	0.0220	1.87E-03

TABLE III. CLASSIFYING RESULTS ON UCI FILES BY GA APPROACH WITH 6 FIXED RULES, DECISION TEMPLATE AND MLR AS DIFFERENT FITNESS FUNCTIONS

File Name	GA Sum		GA Product		GA Max		GA Min		GA Median		GA Majority Vote		GA Decision Template		GA MLR	
	Mean	*Variance*	*Mean*	*Variance*	*Mean*	*Variance*	*Mean*	*Variance*	*Mean*	*Variance*	*Mean*	*Variance*	*Mean*	*Variance*	*Mean*	*Variance*
Bupa	0.2923	3.68E-03	0.3004	5.20E-03	0.3136	5.30E-03	0.3051	5.37E-03	0.2952	5.14E-03	0.3162	3.99E-03	0.3210	5.85E-03	0.3020	4.48E-03
Artificial	0.1917	2.04E-03	0.1804	1.48E-03	0.2043	1.81E-03	0.2007	1.91E-03	0.2053	1.64E-03	0.2019	1.64E-03	0.1980	2.20E-03	0.1941	1.80E-03
Pima	0.2241	1.27E-03	0.2190	1.93E-03	0.2231	2.12E-03	0.2151	2.00E-03	0.2199	1.69E-03	0.2216	1.62E-03	0.2274	1.29E-03	0.2191	1.30E-03
Sonar	0.1485	6.71E-03	0.1490	5.67E-03	0.1429	6.20E-03	0.1452	5.22E-03	0.1466	6.12E-03	0.1635	6.01E-03	0.1735	7.55E-03	0.1449	5.49E-03
Heart	0.1433	3.20E-03	0.1344	3.26E-03	0.1385	4.38E-03	0.1307	3.03E-03	0.1370	3.69E-03	0.1426	4.40E-03	0.1485	5.12E-03	0.1474	4.20E-03
Haberman	0.2343	2.04E-03	0.2360	2.29E-03	0.2413	2.56E-03	0.2520	1.94E-03	0.2422	1.96E-03	0.2507	2.23E-03	0.2483	5.03E-03	0.2412	2.03E-03
Balance	0.1126	4.12E-04	0.1142	5.01E-04	0.1107	3.62E-04	0.1123	6.07E-04	0.1126	4.83E-04	0.1249	4.44E-04	0.0832	1.23E-03	0.1110	5.89E-04

Fertility	0.1280	2.62E-03	0.0930	4.85E-03	0.1010	2.90E-03	0.0980	4.80E-03	0.0950	5.88E-03	0.1040	4.18E-03	0.2040	1.20E-02	0.1200	3.40E-03
Wdbc	0.0320	4.98E-04	0.0332	3.88E-04	0.0330	6.25E-04	0.0348	5.79E-04	0.0260	4.11E-04	0.0293	3.58E-04	0.0285	4.61E-04	0.0283	4.64E-04
Australian	0.1219	1.56E-03	0.1209	1.22E-03	0.1191	1.35E-03	0.1267	1.62E-03	0.1223	1.78E-03	0.1257	1.31E-03	0.1155	1.62E-03	0.1196	1.76E-03
TAE	0.4404	1.46E-02	0.3922	1.39E-02	0.4332	1.59E-02	0.4501	1.29E-02	0.4319	2.02E-02	0.4403	1.50E-02	0.4684	1.52E-02	0.4519	1.46E-02
Contraceptive	0.4586	1.32E-03	0.4599	1.65E-03	0.4640	1.07E-03	0.4606	1.46E-03	0.4569	1.26E-03	0.4623	1.42E-03	0.4758	1.51E-03	0.4601	1.16E-03
Vehicle	0.2266	1.18E-03	0.2152	1.23E-03	0.2203	1.88E-03	0.2190	1.71E-03	0.2201	1.39E-03	0.2164	1.24E-03	0.2062	1.38E-03	0.2154	1.37E-03
Iris	0.0200	1.20E-03	0.0200	1.02E-03	0.0260	1.32E-03	0.0200	1.02E-03	0.0200	1.29E-03	0.0200	1.47E-03	0.0233	1.37E-03	0.0213	1.32E-03

TABLE IV. STATISTICAL TEST COMPARING GA APPROACH WITH COMBINING CLASSIFIERS APPROACHES

	GA Sum vs. Sum	GA Product vs. Product	GA Max vs. Max	GA Min vs. Min	GA Median vs. Median	GA Majority Vote vs. Majority Vote	GA Decision Template vs. Decision Template	GA MLR vs. MLR
Better	7	10	11	12	11	10	10	5
Competitive	7	4	3	2	3	4	4	9
Worse	0	0	0	0	0	0	0	0

TABLE V. STATISTICAL TEST COMPARING EACH GA APPROACH WITH BEST RESULT SELECTED FROM ALL GA APPROACHES

	GA Sum vs. Select Best	GA Product vs. Select Best	GA Max vs. Select Best	GA Min vs. Select Best	GA Median vs. Select Best	GA Vote vs. Select Best	GA Decision Template vs. Select Best	GA MLR vs. Select Best
Competitive	8	11	8	7	10	6	5	8
Worse	6	3	6	7	4	8	9	6

TABLE VI. INFORMATION OF 10 CLASSES CHOSEN FROM CLEF2009 MEDICAL IMAGE DATABASE

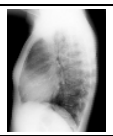
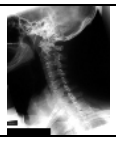
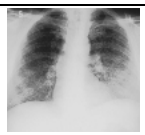
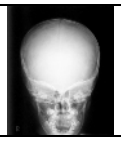
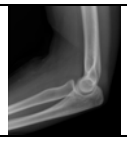
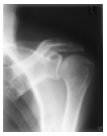
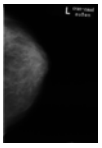
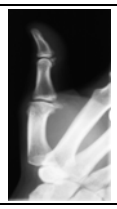
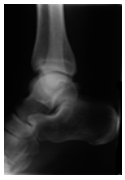
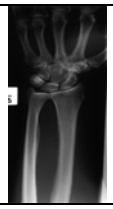
Image					
Description	Abdomen	Cervical	Chest	Facial cranium	Left Elbow
Number of observation	80	81	80	80	69
Image					
Description	Left Shoulder	Left Breast	Finger	Left Ankle Joint	Left Carpal Joint
Number of observation	80	80	66	80	80

TABLE VII. CLASSIFYING RESULTS ON CLEF2009 BY 6 FIXED RULES, DECISION TEMPLATE AND MLR

	Sum		Product		Max		Min	
	Mean	***Variance***	***Mean***	***Variance***	***Mean***	***Variance***	***Mean***	***Variance***
HLBP 10 classes	0.2023	1.85E-03	0.2300	2.05E-03	0.2211	1.90E-03	0.2450	2.02E-03
	Median		**Vote**		**Decision Template**		**MLR**	
	Mean	**Variance**	**Mean**	**Variance**	**Mean**	**Variance**	**Mean**	**Variance**
	0.2144	2.13E-03	0.2266	1.36E-03	0.1643	1.48E-03	0.1890	1.64E-03

TABLE VIII. CLASSIFYING RESULTS ON CLEF2009 BY GA APPROACH WITH 6 FIXED RULES, DECISION TEMPLATE AND MLR AS DIFFERENT FITNESS FUNCTIONS

	GA Sum		GA Product		GA Max		GA Min	
	Mean	***Variance***	***Mean***	***Variance***	***Mean***	***Variance***	***Mean***	***Variance***
HLBP 10 classes	0.1543	1.59E-03	0.1438	1.36E-03	0.1579	1.46E-03	0.1487	1.70E-03
	GA Median		**GA Majority Vote**		**GA Decision Template**		**GA MLR**	
	Mean	***Variance***	***Mean***	***Variance***	***Mean***	***Variance***	***Mean***	***Variance***
	0.1491	1.60E-03	0.1647	1.72E-03	0.1502	1.07E-03	0.1505	1.46E-03

TABLE IX. STATISTICAL TEST COMPARING GA APPROACH WITH NORMAL COMBINING APPROACH

Fitness Function	GA Sum vs. Sum	GA Product vs. Product	GA Max vs. Max	GA Min vs. Min	GA Median vs. Median	GA Majority Vote vs. Majority Vote	GA Decision Template vs. Decision Template	GA MLR vs. MLR
Better	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**

Competitive	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Worse	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**