2014 International Conference on Intelligent Computing, ICIC2014, Taiyuan, China. (LNCS, Vol. 8589, pp 56-67)

Combining Multi Classifiers based on A Genetic Algorithm - Gaussian Mixture Model framework

Tien Thanh Nguyen[1], Alan Wee-Chung Liew[1], Cuong To[1], Minh Toan Tran[2], Mai Phuong Nguyen[3]

[1]School of Communication and Information Technology, Griffith University, QLD, Australia.
[2]School of Applied Mathematics and Informatics, Hanoi University of Science and Technology, Hanoi Vietnam
[3]College of Business, Massey University, New Zealand

EMAIL: tienthanh.nguyen2@griffithuni.edu.au

Abstract. Combining outputs from different classifiers to achieve high accuracy in classification task is one of the most active research areas in ensemble method. Although many state-of-art approaches have been introduced, no method is outstanding compared with the others on numerous data sources. With the aim of introducing an effective classification model, we propose a Gaussian Mixture Model (GMM) based method that combines outputs of base classifiers (called meta-data or Level1 data) resulted from Stacking Algorithm. We further apply Genetic Algorithm (GA) to that data as feature selection strategy to explore an optimal subset of Level1 in which our GMM-based approach can achieve high accuracy. Two methods are combined in a single framework called GAGMM. Experiments implemented on 21 UCI Machine Learning Repository data files and CLEF2009 medical image database demonstrate the advantage of our framework compared with other well-known combining algorithms such as Decision Template, Multiple Response Linear Regression (MLR), SCANN and fixed combining rules as well as GMM-based approaches on original data.

Keywords: Stacking Algorithm, feature selection, Gaussian Mixture Model, Genetic Algorithm, multiple classifier system, classifier fusion, combining classifiers, ensemble method.

1 INTRODUCTION

In recent years, ensemble methods which have been studied extensively are one of the most active research areas in supervised learning. There are several taxonomies of ensemble methods introduced that focuses on the different factors and views on the ensemble [1]. Here we introduce a simple classification by Ho [2] where she simply divided ensemble methods into two types:

- Mixture of experts: Using a fixed set of classifiers and after that making decision from outputs of these classifiers.
- Coverage: Generating genetic classifiers, which are classifiers from same family but have different few factors; for instance, classifiers with different parameters. Next, classifiers are combined to have final decision.

In this paper, we focus on the first type of ensemble methods where decision is formed by combining outputs of different classifiers. There are several combining classifiers strategies proposed and among of them, Stacking-based approaches are one of the most popular ensemble methods. The Stacking was first proposed by Wolpert [3] and was further developed by Ting [7]. In this model, the training set is divided into nearly equal disjoint parts. One part plays as test set and the others play as training set so all observations will be tested once. The outputs of Stacking are posterior probabilities that observations belong to a class according to each classifier. Posterior probabilities of all observations are gathered in a group called meta-data or Level1 data to distinguish it from Level0 data which is the original data.

To apply Stacking to ensemble of classifiers, Ting [7] proposed Multiple Response Linear Regression algorithm (MLR) to combine posterior probabilities of each observation based on sum of weights calculated from linear regression functions. The idea of MLR is that each classifier sets a different weight on each class and combining algorithm is then conducted based on posterior probability and its associated weight. Kuncheva [4] defined Decision Template for each class computed on posterior probability of observations in training set and their true class label and then detailed eleven measurements between posterior probability of unlabeled observation and each Decision Template [4] to output the prediction Meanwhile, Metz [9] combined Stacking, Correspondence Analysis (CA) and K Nearest Neighbor (KNN) in a single algorithm called SCANN. His idea was to form representation on new space for outputs of base classifiers generated by applying Stacking plus true label of each observation. Finally, KNN is used as classifier on new space to obtain prediction for unlabeled observation. Recently, Szepannek [12] developed idea from pairwise combining by finding which classifier is best for a pair including class i and j ($i \neq j$) and used a pairwise coupling algorithm to combine outputs of all pairs to make posterior for each class. Zhang [13] used linear programming to find weight that each classifier puts on a particular class. Sen [14] introduced a method that was inspired by MLR used hinge loss function to the combiner instead of using conventional least square loss. By using new function with regularization, he proposed three different combination, namely weighted sum, dependent weighted sum and linear stacked generalization based on different regularizations with group sparsity.

Another popular and simple approach to combine outputs from base classifiers is using fixed rules. Kittler [8] presented six fixed rules namely Sum, Product, Vote, Min, Max and Average. The advantage of applying fixed rules for ensemble system is that it only needs Level1 data of unlabeled observation as input instead of Level1 data of training set. Consequently, computational cost is reduced.

Generally speaking, most of strategies have focused on discovering “secrets” of Level1 data so as to exploit strategies to form hypothesis about relationship between feature vector and its corresponding label. We adopt that strategy by proposing new framework operated on Level1 data that could be competitive with these other state-of-art combining algorithms. In this paper, a novel combining classifiers that operates on outputs of base classifiers is introduced with different perspective; we put attention on distribution model of Level1 data. The proposed classifier fusion is formed by combining Gaussian Mixture Model (GMM) as classifier and Genetic Algorithm (GA) as feature selection method in a single effective framework. The purpose is that we want to construct a model which is competitive with respect to other state-of-art ensemble methods such as fixed combining rules as well as trainable methods like Decision Template, SCANN and MLR. In the next section, we introduce the proposed framework in detail and then perform empirical evaluations on two popular datasets namely UCI Machine Learning Repository [23] and CLEF2009 medical image database. Finally, we summarize and propose several future improvements.

2 METHODOLOGY

2.1 Classifier fusion based on GMM

Let us denote class label 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:

$$\begin{bmatrix} P_1(\mathrm{W}_1 \mid X_1)...\ P_1(\mathrm{W}_M \mid X_1)...\ P_K(\mathrm{W}_1 \mid X_1)...P_K(\mathrm{W}_M \mid X_1) \\ P_1(\mathrm{W}_1 \mid X_2)...\ P_1(\mathrm{W}_M \mid X_2)...\ P_K(\mathrm{W}_1 \mid X_2)...P_K(\mathrm{W}_M \mid X_2) \\ \cdots \qquad\qquad\qquad\qquad \cdots \\ P_1(\mathrm{W}_1 \mid X_N)...P_1(\mathrm{W}_M \mid X_N)...P_K(\mathrm{W}_1 \mid X_N)...P_K(\mathrm{W}_M \mid X_N) \end{bmatrix} \tag{1}$$

Level1 data of an observation X is defined as:

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

In our proposed algorithm we employ Stacking Algorithm to generate Level1 data of original training set. Pseudo code of Stacking is given below:

```
Algorithm 1: Generate Level1 data (Stacking algorithm)
Input: Level0 data, K base classifiers.
Output: Level1 data(eqn. 1)
Divide Level0 to nearly equal disjoint B parts.
For i = 1 to B
        •  Denote Level0(i) as ith part, use Level0-Level0(i) as training
           set and Level0(i) as test set.
        •  Classify observations in Level0(i) with model formed by K base
           classifiers on Level0-Level0(i).
        •  Store Level1 data of observations belonged to Level0(i)
End
```

The most important distinction between our work and the previous work is we use GMM-based approach on Level1 to construct combining classifiers model. In our knowledge, all previous GMM-based approaches were conducted on Level0 in which they suffer from significant limitations in modeling real data sets. Attributes in Level0 are frequently different in nature, measurement unit, and type; as a result, GMM cannot perform well when it is selected to approximate distribution of Level0. Level1, on the other hand, can be viewed as scaled result from feature domain to posterior domain where data is reshaped to be all real values in [0, 1]. Observations belonged to the same class will have nearly equal posterior probabilities by prediction from a base classifier; as a result they may be located nearby in the new domain. It is likely that Level1 will be more discriminative than the original data and therefore GMM on Level1 will have been more effective than on Level0. Besides, it is well known in literature that the higher the dimension of the data, the lower the effectiveness of GMM approximation. Therefore applying GMM to Level1 not only results in a reduction in storage cost but also improves its effectiveness in scenarios that Level1 has lower dimension than Level0.

Our proposed model is illustrated in Fig 1. Here training set is classified by K base classifiers based on Stacking Algorithm to generate the Level1 data. Because the label of all observations in training set is known so we can group Level1 into M classes such that observations belonging to the same class are grouped together. Next, GMM is employed as a statistical representation model for each class and then the class label of an observation is predicted by posterior probability using Bayes model.

In testing produce, unlabeled observation is classified by K base classifiers with the model generated on training set to output Level1 data of that observation (eqn. 2). That meta-data is then gone through M-GMMs as input data and the final prediction is released through maximization among posterior probabilities corresponding with all classes.

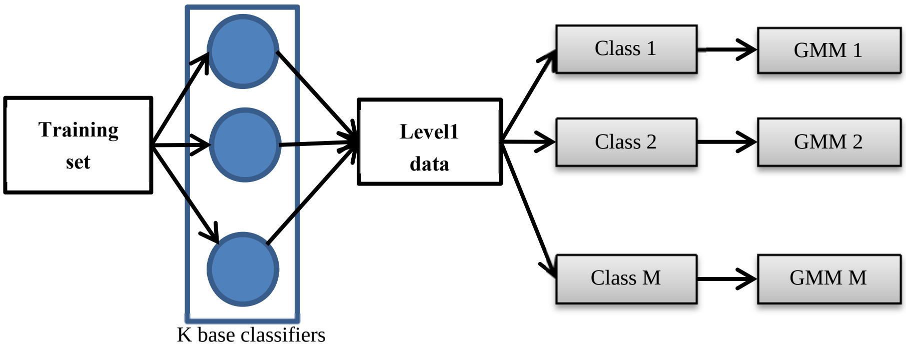


Fig. 1. GMM-based approach on Level1 data

For i^{th} class, we propose the prediction framework based on Bayes model

$$\underbrace{P(GMM_i \mid \mathbf{x})}_{posteriror} \sim \underbrace{P(\mathbf{x} \mid GMM_i)}_{likelihood} \times \underbrace{P(GMM_i)}_{prior} \tag{3}$$

Here the likelihood function is modeled by GMM:

$$P(\mathbf{x} \mid GMM_i) = P(\mathbf{x} \mid \boldsymbol{\mu}_{ip}, \boldsymbol{\Sigma}_{ip}, \omega_{ip}) = \sum_{p=1}^{P_i} \omega_{ip} \mathcal{N}\left(\mathbf{x} \mid \boldsymbol{\mu}_{ip}, \boldsymbol{\Sigma}_{ip}\right) \tag{4}$$

where

$$\mathcal{N}\left(\mathbf{x} \mid \boldsymbol{\mu}_{ip}, \boldsymbol{\Sigma}_{ip}\right) = \frac{1}{\left(2\pi\right)^{MK/2} \left|\boldsymbol{\Sigma}_{ip}\right|^{1/2}} \exp\left\{-\frac{1}{2}\left(\mathbf{x} - \boldsymbol{\mu}_{ip}\right)^{\mathbf{T}} \boldsymbol{\Sigma}_{ip}^{\mathbf{-1}} \left(\mathbf{x} - \boldsymbol{\mu}_{ip}\right)\right\} \tag{5}$$

P_i is the number of Gaussian components in GMM_i and $\boldsymbol{\mu}_{ip}, \boldsymbol{\Sigma}_{ip}$ are the mean and covariance of p^{th} component in the model of i^{th} class, respectively. The prior probability in (eqn. 3) of i^{th} class is defined by:

$$P(GMM_i) = \frac{n_i}{N} \tag{6}$$

where n_i is the number of observations in i^{th} class and N is the total number of observations in the training set. To find parameters of GMM, we apply Expectation Maximization (EM) algorithm by maximize the likelihood function with respect to the means, covariances of components, and mixing coefficients [21].

Now with dataset $\mathbf{X} = \{\mathbf{x_i}\}$ $i = 1, N_i$ corresponding with i^{th} class, the question is how to find the number of component for GMM. Frequently, it is fixed by a specific number. Here, we propose applying Bayes Criterion Information (BIC) to find the optimal model [21] with an assumption that we have a set of model $\{F_j\}$ with parameters $\boldsymbol{\theta}_j$ where $\boldsymbol{\theta}_j$ are denoted for all parameters of model. To find model by BIC, for i^{th} class ($i = \overline{1, M}$), we compute:

$$\ln P(\mathbf{X} \mid F_j) \approx \ln P(\mathbf{X} \mid F_j, \boldsymbol{\theta}_{MAP}) - \frac{1}{2}\left|\boldsymbol{\theta}_j\right| \ln n_i \tag{7}$$

where $\boldsymbol{\theta}_{MAP}$ is corresponding with the maximum of posterior distribution and $\left|\boldsymbol{\theta}_j\right|$ denotes the number of parameters in $\boldsymbol{\theta}_j$. In scenario of GMM, $\{F_j\}$ are group in which each element is a GMM and $\{\boldsymbol{\theta}_j\}$ include three parameters namely means, covariances of Gaussian components, and mixing coefficients in mixture model and n_i is the number of observations in i^{th} class.

It is noted that as Level1 data is obtained from K base classifier, it conveys the posterior information from each classifier about how much support a classifier has for an observation to belong to a class. In some cases, there are columns in Level1 data in which $\exists k, m$, $P_k(W_m \mid X_i)$ is nearly constant for all i. Hence, the covariance matrix may be singular and EM is unable to solve for GMM. To overcome this problem, we propose to regularize Level1 before applying GMM to Level1, by checking for condition in eqn. 8 on all columns. If the condition is satisfied, we choose several random elements in this column and increase its value by a small quantity. This procedure only adds small value to some random elements in a column so it does not affect the nature of the posterior probability as well as the covariance matrix.

$$\left|x - \overline{x}\right| < \varepsilon \;\; \forall \text{ column vector } x \text{ on Level1 data and small value } \varepsilon \tag{8}$$

where $\overline{x}$ is mean value of vector x.

```
Algorithm 2: Regularize Level1
Input: Level1, extravalue, r
Output: Regularized Level1
For $i^{th}$ column of Level1
    If Condition(eqn. 8) = true
        Generate r random number in (1,size(column))
```

```
              Element(r) = Element(r)+ extravalue
        End if
   End
   Return Level1
```

```
   Algorithm 3: GMM for combining classifiers

   Training progress:
   Input: Training set (Level0), K base classifiers, PiMax: maximum number of
Gaussian component for $i^{th}$ class.
   Output: GMM suitable with each class.
   Step1: Applied Algorithm 1 to generate Level1 of Level0.
   Step2: Gather same labeled observations in M classes; compute $P(GMM_i)$ (eqn.
6), mean and covariance for each class.
   Step3:For $i^{th}$ class
             Call Algorithm 2 to regularize Level1 of class
             For p=1 to PiMax
                  Apply EM algorithm to find GMM model corresponding with p
           components.
                  Compute BIC.
             End
        Select Pi corresponding with max(BIC) and GMM with Pi components.
        End
   Save GMMs

   Testing progress:
   Input: Unlabeled observation XTest
   Output: predicted label of XTest
   Step1: P Compute Level1 of XTest with model formed by base classifiers and
Level0.
   Step2: For each class
          Compute $P(XTest|GMM_i)$ (eqn. 4) and posterior related to class (eqn.
3) as $P(GMM_i | XTest) \sim P(XTest|GMM_i)\times P(GMM_i)$
        End
   Step3: Predict label of XTest due to $XTest \in W_t$ if $t = \arg\max_{i=1,M} P(GMM_i | XTest)$
   End
```

2.2 GAGMM Framework

Recently, several GA-based approaches have been proposed to improve the accuracy of classifier fusion by solving both the classifier and feature selection problems [6, 15 and 16]. For GMM, GA is only applied to improve EM algorithm [17]. Here, our novel idea is to employ GA as a feature selection technique on Level1 data. It means that if columns $P_k(\mathrm{W}_m | X_i)(i = \overline{1,n})$ for each m and k, which is posterior probability that X_i belong to class W_m given by k^{th} classifier, is not discriminative enough, its elimination from Level1 data could increase the classification accuracy of the combining algorithm.

To build the GAGMM framework on Level1, first, we propose the structure of chromosome in population. Each chromosome includes $M \times K$ genes due to dimension of Level1 (Fig 2). We use two values $\{0,1\}$ to encode for each gene in a chromosome in which:

$$Gene(i) = \begin{cases} 1 & \text{if } i^{th} \text{ feature is selected} \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

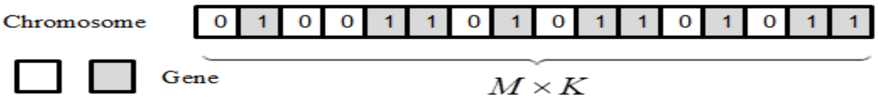


Fig. 2. Proposed chromosome structure

At crossover stage, we employ single splitter since a same single random point is selected on two parents. Each parent exchanges its head to the other while keeps its tail. After this stage, two new offspring chromosomes are created. Next, based on mutation probability, we select random genes from the offspring chromosomes in population and change their values by $0 \to 1$ or $1 \to 0$. Mutation helps GA reaches not only local extreme value but also global one. Here we use accuracy of combination by GMM-based approach on Level1 as fitness value of GA such that extreme value is maximized.

This framework can be viewed as mechanism to learn an optimal subset feature from Level1 data of training set by using GMM based classifiers and therefore it can be classified to wrapper type among feature selection methods [24]. Since only single training set is available to evaluate, cross validation method is used to divide Level1 data of training set to several disjoint parts in which each is selected as Level1 validation set while the others grouped in a single Level1 training set to build classification model with GMM classifiers. Attributes are selected corresponding with encoding of each chromosome, and therefore, GMM classifiers will work on reduced dimensional data. Fitness value which is accuracy of classifier is computed by averaging of all accuracy obtained from Level1 validation test. It is the single objective to select optimal subset based on GMM classifiers. It is worth to note that due to low dimension of Level1 data in several cases, GA-based approach can converge after a small number of interactions; as a result, training process plus GA is less time-consuming than that applied on original data.

GAGMM framework is detailed below:

```
Algorithm 4: GAGMM

Training process:
Input: Training set (Level0), K base classifiers, mutation probability (PMul), population size (L).
Output: optimal chromosome encoding as an optimal subset of Level1 feature and GMMs for classes associated with that chromosome.
Step1: Use Algorithm 1 to generate Level1 of training set.
Step2: Use v-Cross Validation (v-CV) strategy in Level1 of training set
Step3: Initialize a population with L random chromosomes
Step4: Compute fitness of each chromosome as averaged accuracy of classification tasks by calling Algorithm 3 with v-CV
Step5: Loop to select L best chromosomes.
While(not converge)
    • Withdraw with replacement L/2 pairs from population, conduct crossover and mutation (based on PMul) to generate new L chromosomes.
    • Compute fitness of each offspring chromosome as similarity to Step4.
    • Add new L chromosomes to population.
    • Select new population with L best fitness value chromosomes.
End
Step6: Select and save the best chromosome from final population and GMMs for classes associated with that chromosome.

Testing process:
Input: unlabeled observation XTest
Output: predicted label of XTest.
Call test process in Algorithm 3.
```

3 EXPERIMENTAL RESULTS

We empirically evaluated our framework on two data sources namely UCI dataset and CLEF2009 medical image database. In our assessment, we compared error rates of our model with each among 6 benchmarks: selecting best results from fixed rules based on outcomes on test set, Decision Template (measure of similarity S_1 [4] is defined as $S_1(DP(X), DT_i) = \frac{\|Level1(X) \cap DT_i\|}{\|Level1(X) \cup DT_i\|}$ where DT_i is Decision Template of i^{th} class and $\|\alpha\|$ is the relative cardinality of the fuzzy set α), MLR, SCANN, GMM on Level0 and GMM on Level1. The appearance of GMM Level0 and GMM Level1 was that we wanted to demonstrate the high performance ability of our model on Level 1 data compared with that on original data as well as the effectiveness of feature selection method. It is interesting to note that MLR, Decision Template and SCANN do not require any initialized parameters in their implementation. Three base classifiers namely Linear Discriminant Analysis, Naïve Bayes and K Nearest Neighbor (with K set to 5) were chosen. The motivation of choosing these base classifiers is that all they have different approaches to classification task; as a result, diversity of ensemble system is ensured. Parameters for GA were initialized by setting population size to 20 and mutation probability to 0.015.

To ensure objectiveness, we performed 10-fold validation and implemented the test 10 times so we had 100 error rates result for a file according to each combining algorithm. For comparison purpose, we used paired statistical t-test to compare the classification results of our approach and each benchmark (level of significance set to 0.05)

3.1 Experiment on UCI files

We chose 21 common UCI data files with number of classes ranging from 2 (Bupa, Artificial, etc…) to 26 (Letter). The number of attributes also changes in a wide range from only 3 attributes (Haberman) to 60 attributes (Sonar). The number of observations in each file also varies considerably, from small files like Iris, Fertility to big file such as Skin&NonSkin (up to 245057 observations) (Table 1). Experimental results of all 21 files are shown in Table 2 and 3.

From the paired statistical t-test in Table 4, it can be concluded that our framework is superior to compared methods. First, there are 6 cases in which GAGMM outperforms the best result from fixed rules while on the other 13 files, both methods are competitive. Besides, GAGMM is better than Decision Template in 14 cases and is not worse in any cases. Remarkable results are obtained on Skin&NonSkin (4.13e-04 vs. 0.033), Ring (0.1108 vs. 0.1894), Letter (0.0802 vs. 0.1133) and Phoneme (0.115 vs. 0.1462).

Next, GAGMM has 18 wins and 1 loss whereas GMM on Level1 has 16 wins and 1 loss with respect to GMM on Level0. This demonstrates GMM on Level1's ability to exceed the performance of GMM-based approaches on Level0.As mentioned in section 2, Level1 data is more uniformity than Level0 data in which all components of feature vector are real data type and being scaled in [0, 1]. As a result, GMM has better representation for distribution approximation on Level1 data than on original data and therefore, advantage of our model is demonstrated. On the other hand, regarding the one loss, we should point out that Ring is simulated data generated from multivariate Gaussian distribution [23], so GMM is expected to perform well on this dataset at Level0.

Comparing SCANN to GAGMM reveals that SCANN outperforms on just 1 file (Letter 0.063 vs. 0.0802) but has 8 losses. In our experiment, SCANN cannot be run on 3 files, namely Fertility, Balance and Skin&NonSkin, because the indicator matrix has columns with all 0 values posterior probability from the K base classifiers. As a result, its column mass will be singular and standardized residuals is not available. Here, we did not put these files in the comparison.

Moreover, we compare GAGMM and MLR. In general, GAGMM performs better than MLR, posting 6 wins and 2 losses. Significant improvements are on Ring (0.1108 vs. 0.17), Balance (0.0755 vs. 0.1225) and Tae (0.4313 vs. 0.4652). However on Letter, MLR performs well with error rate of only 0.0427 while our framework has twice the error rate.

Finally, GAGMM outperforms GMM on Level1, posting 5 wins and no loss. It is the consequence of feature selection method on Level1 data that causes not only reducing dimension of data but also improve performance of combining classifiers system. Fives datasets address the high accuracy of GAGMM compared with GMM Level1 are Pima (22.79% vs. 24.32%), Balance (7.55% vs. 8.39%), Fertility (12.7% vs. 18.5%), Wdbc (3.21% vs. 3.87%) and Iris (2.67% vs. 3.6%).

We also assess the dimension of data in Level0, Level1, and the data resulted from GAGMM. Fig 3 shows that the dimensions of data from GAGMM are significantly lower than those of Level0 and Level1. Remarkable results are observed on Fertility, Pima, Phoneme (with just 1 dimension) and Magic, Twonorm and Haberman (with just 2 dimensions). It is noted that we can use Level1 as feature of dataset instead of using features from Level0. Hence, our GAGMM framework helps reduce the dimension of data. Thus our framework not only lessens storage space but also improves performance of the classification system.

File name	Number of attributes	Attribute type	Number of observations	Number of classes	Number of attributes on Level1 (3 classifiers)
Bupa	6	C,I,R	345	2	6
Pima	6	R,I	768	2	6
Sonar	60	R	208	2	6
Heart	13	C,I,R	270	2	6
Phoneme	5	R	540	2	6
Haberman	3	I	306	2	6
Titanic	3	R,I	2201	2	6
Balance	4	C	625	3	9
Fertility	9	R	100	2	6
Wdbc	30	R	569	2	6
Australian	14	C,I,R	690	2	6
Twonorm (*)	20	R	7400	2	6
Magic	10	R	19020	2	6
Ring (*)	20	R	7400	2	6
Contraceptive	9	C,I	1473	3	9
Vehicle	18	I	946	4	12
Iris	4	R	150	3	9
Tae	20	C,I	151	2	6
Letter	16	I	20000	26	78
Skin&NonSkin	3	R	245057	2	6
Artificial	10	R	700	2	6

R: Real, C: Category, I: Integer (*) Simulated data

Table 1. UCI DATA FILES USED IN OUR EXPERIMENT

File name	Sum rule		Product rule		Max rule	
	Mean	Variance	Mean	Variance	Mean	Variance
Bupa	0.3028	4.26E-03	0.3021	4.12E-03	0.2986	4.15E-03
Artificial	0.2230	2.06E-03	**0.2193**	**2.05E-03**	0.2450	2.57E-03
Pima	0.2405	1.62E-03	0.2419	1.63E-03	0.2411	1.69E-03
Sonar	0.2259	9.55E-03	0.2285	9.81E-03	0.2260	7.01E-03
Heart	0.1637	4.59E-03	0.1648	5.20E-03	0.1730	4.14E-03
Phoneme	0.1713	1.90E-04	0.1518	2.87E-04	**0.1407**	**1.95E-04**
Haberman	**0.2392**	**2.39E-03**	0.2424	3.08E-03	0.2457	3.18E-03
Titanic	0.2167	6.01E-04	0.2167	5.65E-04	0.2167	6.59E-04
Balance	0.1113	5.55E-04	0.1131	4.95E-04	**0.1112**	**4.82E-04**
Fertility	0.1290	2.46E-03	0.1290	2.26E-03	**0.1270**	**1.97E-03**
Skin&NonSkin	0.0412	1.40E-06	0.0006	2.73E-08	0.0006	2.22E-08
Wdbc	0.0401	7.07E-04	0.0517	8.19E-04	0.0485	8.03E-04
Australian	0.1281	1.78E-03	0.1594	1.91E-03	0.1604	1.95E-03
Twonorm	0.0221	3.00E-05	0.0225	2.69E-05	0.0231	2.39E-05
Magic	0.1925	5.31E-05	0.1921	5.16E-05	0.1911	6.40E-05
Ring	**0.2122**	**1.62E-04**	0.2275	1.09E-04	0.2436	1.53E-04
Tae	0.4625	1.36E-02	0.4622	1.14E-02	0.5191	1.11E-02
Contraceptive	**0.4653**	**1.79E-03**	0.4667	1.19E-03	0.4734	1.19E-03
Vehicle	0.2671	1.38E-03	**0.2645**	**1.37E-03**	0.2937	1.54E-03
Iris	0.0387	2.59E-03	0.0407	2.39E-03	0.0440	3.13E-03
Letter	0.1388	6.50E-05	0.0856	3.10E-05	**0.0760**	**3.94E-05**

File name	Min rule		Median rule		Majority vote	
	Mean	Variance	Mean	Variance	Mean	Variance
Bupa	**0.2970**	**4.89E-03**	0.3428	4.46E-03	0.3429	4.04E-03
Artificial	0.2453	2.90E-03	0.3089	1.36E-03	0.3073	1.03E-03

Pima	0.2449	2.02E-03	0.2376	1.69E-03	**0.2365**	**2.10E-03**
Sonar	0.2298	9.32E-03	0.2104	1.00E-02	**0.2079**	**8.16E-03**
Heart	0.1700	4.01E-03	**0.1570**	**4.64E-03**	0.1604	3.87E-03
Phoneme	0.1417	2.10E-04	0.2254	2.71E-04	0.2257	2.73E-04
Haberman	0.2461	2.47E-03	0.2524	1.67E-03	0.2504	1.76E-03
Titanic	**0.2167**	**5.00E-04**	0.2216	5.25E-04	0.2217	4.61E-04
Balance	0.1232	4.99E-04	0.1155	4.93E-04	0.1261	4.63E-04
Fertility	0.1280	2.02E-03	0.1330	2.81E-03	0.1310	2.34E-03
Skin&NonSkin	**0.0006**	**2.13E-08**	0.0528	2.23E-06	0.0528	1.90E-06
Wdbc	0.0522	7.71E-04	**0.0395**	**5.03E-04**	0.0406	6.47E-04
Australian	0.1609	1.80E-03	0.1270	1.56E-03	**0.1262**	**1.37E-03**
Twonorm	0.0233	3.92E-05	0.0217	2.74E-05	**0.0216**	**2.82E-05**
Magic	**0.1905**	**5.72E-05**	0.2004	5.81E-05	0.2006	7.49E-05
Ring	0.2437	1.33E-04	0.2368	1.93E-04	0.2365	2.00E-04
Tae	0.4868	1.40E-02	0.4443	1.46E-02	**0.4435**	**1.70E-02**
Contraceptive	0.4766	1.77E-03	0.4803	1.31E-03	0.4844	1.27E-03
Vehicle	0.2737	1.57E-03	0.2858	1.57E-03	0.3194	2.01E-03
Iris	0.0413	2.56E-03	0.0333	1.64E-03	**0.0327**	**1.73E-03**
Letter	0.0941	4.42E-05	0.2451	8.22E-05	0.2390	7.68E-05

Table 2. ERROR RATES OF COMBINING CLASSIFIERS BASED ON FIXED RULES

File name	MLR		Best results from fixed Rules		SCANN		Decision Template	
	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance
Bupa	0.3033	4.70E-03	0.2970	4.89E-03	0.3304	4.29E-03	0.3348	7.10E-03
Artificial	0.2426	2.20E-03	0.2193	2.05E-03	0.2374	2.12E-03	0.2433	1.60E-03
Pima	0.2432	2.30E-03	0.2365	2.10E-03	0.2384	2.06E-03	0.2482	2.00E-03
Sonar	0.1974	7.20E-03	0.2079	8.16E-03	0.2128	8.01E-03	0.2129	8.80E-03
Heart	0.1607	4.70E-03	0.1570	4.64E-03	0.1637	4.14E-03	0.1541	4.00E-03
Phoneme	0.1136	1.75E-04	0.1407	1.95E-04	0.1229	6.53E-04	0.1462	2.00E-04
Haberman	0.2428	3.30E-03	0.2392	2.39E-03	0.2536	1.74E-03	0.2779	5.00E-03
Titanic	0.2169	4.00E-04	0.2167	5.00E-04	0.2216	6.29E-04	0.2167	6.00E-04
Balance	0.1225	8.00E-04	0.1112	4.82E-04	X	X	0.0988	1.40E-03
Fertility	0.1250	2.28E-03	0.1270	1.97E-03	X	X	0.4520	3.41E-02
Skin&NonSkin	4.79E-4	1.97E-08	0.0006	2.13E-08	X	X	0.0332	1.64E-06
Wdbc	0.0399	7.00E-04	0.0395	5.03E-04	0.0397	5.64E-04	0.0385	5.00E-04
Australian	0.1268	1.80E-03	0.1262	1.37E-03	0.1259	1.77E-03	0.1346	1.50E-03
Twonorm	0.0217	2.24E-05	0.0216	2.82E-05	0.0216	2.39E-05	0.0221	2.62E-05
Magic	0.1875	7.76E-05	0.1905	5.72E-05	0.2002	6.14E-05	0.1927	7.82E-05
Ring	0.1700	1.69E-04	0.2122	1.62E-04	0.2150	2.44E-04	0.1894	1.78E-04
Tae	0.4652	1.24E-02	0.4435	1.70E-02	0.4428	1.34E-02	0.4643	1.21E-02
Contraceptive	0.4675	1.10E-03	0.4653	1.79E-03	0.4869	1.80E-03	0.4781	1.40E-03
Vehicle	0.2139	1.40E-03	0.2645	1.37E-03	0.2224	1.54E-03	0.2161	1.50E-03
Iris	0.0220	1.87E-03	0.0327	1.73E-03	0.0320	2.00E-03	0.0400	2.50E-03
Letter	0.0427	1.63E-05	0.0760	3.94E-05	0.063	2.42E-05	0.1133	4.91E-05

File name	GMM on Level0		GMM on Level1		GAGMM	
	Mean	Variance	Mean	Variance	Mean	Variance
Bupa	0.4419	5.80E-03	0.3022	5.31E-03	0.2999	4.79E-03
Artificial	0.4507	8.00E-03	0.2374	2.40E-03	0.2423	2.64E-03
Pima	0.2466	2.40E-03	0.2432	2.60E-03	0.2279	1.95E-03
Sonar	0.3193	1.26E-02	0.2009	6.20E-03	0.2050	8.09E-03
Heart	0.1715	7.30E-03	0.1559	4.51E-03	0.1544	4.67E-03
Phoneme	0.2400	4.00E-04	0.1165	2.01E-04	0.1150	1.39E-04
Haberman	0.2696	2.00E-03	0.2458	3.36E-03	0.2461	3.96E-03
Titanic	0.2904	2.01E-02	0.2167	5.91E-04	0.2217	1.16E-03
Balance	0.1214	1.10E-03	0.0839	1.21E-03	0.0755	9.68E-04
Fertility	0.3130	7.47E-02	0.1850	1.05E-02	0.1270	2.40E-03
Skin&NonSkin	0.0761	2.21E-06	4.10E-4	1.53E-08	4.13E-4	1.98E-08
Wdbc	0.0678	1.10E-03	0.0387	5.98E-04	0.0321	5.25E-04
Australian	0.1980	1.80E-03	0.1222	1.30E-03	0.1210	1.60E-03
Twonorm	0.0216	2.83E-05	0.0219	2.78E-05	0.0220	2.72E-05
Magic	0.2733	5.06E-05	0.1921	8.34E-05	0.1918	6.03E-05
Ring	0.0209	2.20E-05	0.1131	1.16E-04	0.1108	1.09E-04
Tae	0.5595	1.39E-02	0.4365	1.36E-02	0.4313	1.58E-02
Contraceptive	0.5306	1.80E-03	0.4667	1.30E-03	0.4624	1.30E-03
Vehicle	0.5424	2.40E-03	0.2166	1.40E-03	0.2131	1.46E-03
Iris	0.0453	2.50E-03	0.0360	2.10E-03	0.0267	1.10E-03
Letter	0.3573	9.82E-05	0.0797	3.03E-05	0.0802	1.32E-05

Table 3. COMPARING ERROR RATES OF DIFFERENT COMBINING ALGORITHMS

	Better	Competitive	Worse
GAGMM vs. MLR	6	13	2
GAGMM vs. SCANN	8	9	1
GAGMM vs. Decision Template	14	7	0
GAGMM vs. SelectBest	6	13	2
GAGMM vs. GMM Level0	18	2	1
GAGMM vs. GMM Level1	5	16	0
GMM Level1 vs. GMM Level10	16	4	1

Table 4. COMPARE GAGMM WITH THE BENCHMARKS AMONG 21 UCI FILES

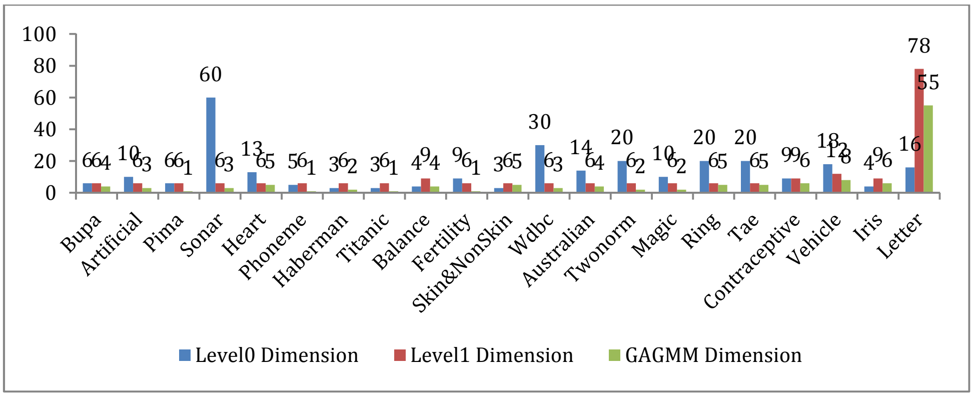


Fig. 3. Compare dimension of data among Level0, Level1 and GAGM (UCI datasets)

3.2 Experiment on CLEF2009 database

We also fulfilled on CLEF 2009 database, a large set of medical image collected by Archen University. It includes 15,363 images allocated in 193 hierarchical categories. In our experiment we chose 7 classes where each has different number of images (Table 5). Firstly, we performed necessary preprocessing techniques like histogram equation and then, Histogram of Local Binary Pattern (HLBP) [22] as feature vector is extracted. The results of the experiment on 7 classes are summarized in Table 6.

Table 7 illustrates the comparison between GAGMM and 6 benchmarks included best result from fixed rules, MLR, Decision Template, SCANN, GMM on Level0 and GMM on Level1. GAGMM posts all 6 wins on experiment and again, it is addressed advantage of our proposed framework since GAGMM is outstanding performance compared with other state-of-art combining algorithms. GMM on Level1, in turn, significantly outperforms than GMM on Level10 (14.69% vs. 24.53) but underperforms GAGMM (14.69% vs. 11.64%). Again, the benefit of proposed model and feature selection method on Level1 is reported.

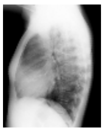
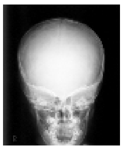
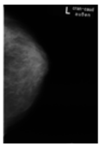
Image							
Description	Abdomen	Cervical	Chest	Facial cranium	Left Elbow	Left Shoulder	Left Breast
# of observation	80	81	80	80	69	80	80

Table 5. INFORMATION OF 7 CLASSES CLEF2009 IN OUR EXPERIMENT

Methods	**HLBP 7 classes**	
	Mean	**Variance**
Sum	0.1636	1.67E-03
Product	0.1831	2.18E-03

Max	0.1835	2.54E-03
Min	0.1978	2.35E-03
Median	0.1725	2.09E-03
Vote	0.1851	1.87E-03
MLR	0.1280	1.80E-03
Decision Template	0.1447	1.90E-03
Best result from fixed rules	0.1636	1.67E-03
SCANN	0.1455	2.62E-03
GMM Level0	0.2453	3.49E-03
GMM Level1	0.1469	2.00E-03
GAGMM	0.1164	2.10E-03

Table 6. ERROR RATES OF DIFFERENT COMBINING ALGORITHMS WITH CLEF2009

GAGMM vs. compared methods	
Better	6
Competitive	0
Worse	0
GMMLevel1 vs. GMM Level0	
Better	1
Competitive	0
Worse	0

Table 7. COMPARING GAGMM WITH 6 BENCHMARKS AND GMM LEVEL1 WITH GMM LEVEL0 RELATED TO CLEF2009

Data	**# of dimension**
Level0	32
Level1	21
Dimension from GAGMM	14

Table 8. COMPARING DIMENSION OF FEATURE RELATED TO CLEF2009

4 CONCLUSION AND FUTURE WORK

We have introduced a framework based on GA and GMM to combining classifiers in a multi classifier system. Our model is run on Level1 data which is the posterior probabilities obtained from applying Stacking Algorithm. Empirical evaluations have demonstrated the superiority of our framework compared with its rivals including best result from fixed rules, Decision Template, SCANN and MLR. Novel framework also remarkably outperforms that on original data due to uniformity characteristic of Level1 data. We reported lower classification error rate on both the 21 UCI files and CLEF2009 medical image database. Besides, the GA-based approach also helps to select the optimal subset of features from the original feature set; resulting in a significant reduction in the dimension of data during classification process. Although feature selection method is time-consuming task, we can conduct that process off-line so it is not a significant problem.

Three problems in our model warrant further research effort. First, GMM is time-consuming due to the determination of optimal number of components by BIC. Second, implementation on small data set is a problem with GMM since when the number of observations in a class is too small; EM algorithm has difficulty estimating the model parameters properly. Finally, the prior probability of eqn. 6 is less appropriate for small and imbalance data. These problems will be the future improvements of our framework.

REFERENCES

[1] Lior 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] Ho T. K., Hull J.J. and Srihari S.N., 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, James C. Bezdek and Robert 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 Lakhmi C. Jain, Designing Classifier Fusion Systems by Genetic Algorithms, IEEE Transactions on Evolution Computation. Vol.4, No.4, September 2000.

[7] Kai Ming Ting, Ian H. Witten, Issues in Stacked Generation, Journal of Artificial In Intelligence Research 10, pp. 271-289, 1999.

[8] Josef Kittler, Mohamad Hatef, Robert P. W. Duin and Jiri Matas, On Combining Classifiers, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.20, No.3, March 1998.

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

[10] Ljupco Todorovski, Saso Džeroski, Combining Classifiers with Meta Decision Trees, Machine Learning 50, pp. 223-249, 2003.

[11] Saso Džeroski, Bernard Ženko, Is Combining Classifiers with Stacking Better than Selecting the Best One? Machine Learning 54, pp. 255-273, 2004.

[12] G. Szepannek, B. Bischl and C. Weihs, On the combination of locally optimal pairwise classifiers, Engineering Applications of Artificial Intelligent, Vol. 22, pp. 79-85, 2009.

[13] Zhang L., Zhou W. D., Sparse ensembles using weighted combination methods based on linear programming, Pattern Recognition, Vol. 44, pp. 97-106, 2011.

[14] Mehmet Umut Sen, Hakan Erdogan, Linear classifier combination and selection using group sparse regularization and hinge loss, Pattern Recognition Letters 34, pp. 265-274, 2013.

[15] Micheal L.Raymer, William F.Punch, Erik D.Goodman, Leslie A.Kuhn and Anil K.Jain, Dimensionality Reduction using Genetic Algorithms, IEEE Transactions on Evolutionary Computation, Vol. 4, No. 2, July 2000.

[16] Bogdan Gabrys and Dymitr Ruta, Genetic Algorithms in Classifier Fusion, Applied Soft Computing 6, pp.337-347, 2006.

[17] Franz Pernkopf and Djamel Bouchaffra, Genetic-Based EM algorithm for Learning Gaussian Mixture Models, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, No. 8, August 2005.

[18] Mario A.T Figueiredo and Anil K. Jain, Unsupervised learning of finite mixture models, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No. 3, pp. 381-396, March 2002.

[19] Baback Moghaddam and Alex Pentland, Probabilistic Visual Learning for Object Representation, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 7, July 1997.

[20] Xiao Hua Liu and Cheng-Lin Liu, Discriminative Model Selection for Gaussian Mixture Models for Classification, First Asian Conference on Pattern Recognition (ACPR), pp. 62-66, 2011.

[21] Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer Press, 2006.

[22] Byoung Chul Ko, Seong Hoon Kim and Jea Yeal 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.

[23] UCI Machine Learning Repository, http://archive.ics.uci.edu/ml/datasets.html

[24] Saeys Y., Inza I. and Larrañaga P., A review of feature selection techniques in bioinformatics, Bioinformatics, Vol. 23, Issue 19, pp. 2507-2517, September 2007.