Combining Classifiers Based on Gaussian Mixture Model Approach to Ensemble Data

Tien Thanh Nguyen[1(✉)], Alan Wee-Chung Liew[1], Minh Toan Tran[2], and Mai Phuong Nguyen[3]

[1] School of Information and Communication Technology, Griffith University, QLD, Australia
{a.liew,tienthanh.nguyen2}@griffithuni.edu.au
[2] School of Applied Mathematics and Informatics,
Hanoi University of Science and Technology, Hanoi, Vietnam
toan.tranminh@hust.edu.vn
[3] College of Business, Massey University, Albany, New Zealand
phuongnm0590@gmail.com

Abstract. Combining multiple classifiers to achieve better performance than any single classifier is one of the most important research areas in machine learning. In this paper, we focus on combining different classifiers to form an effective ensemble system. By introducing a novel framework operated on outputs of different classifiers, our aim is to build a powerful model which is competitive to other well-known combining algorithms such as Decision Template, Multiple Response Linear Regression (MLR), SCANN and fixed combining rules. Our approach is difference from the traditional approaches in that we use Gaussian Mixture Model (GMM) to model distribution of Level1 data and to predict the label of an observation based on maximizing the posterior probability realized through Bayes model. We also apply Principle Component Analysis (PCA) to output of base classifiers to reduce its dimension of what before GMM modeling. Experiments were evaluated on 21 datasets coming from University of California Irvine (UCI) Machine Learning Repository to demonstrate the benefits of our framework compared with several benchmark algorithms.

Keywords: Gaussian mixture model (GMM) · Ensemble method · Multi-classifier system · Combining classifiers · Classifier fusion · Stacking algorithm · Principle component analysis (PCA)

1 Introduction and Recent Work

Traditionally, single learning algorithm is usually employed to solve classification problems by training a classifier on a particular training set which contains hypothesis about the relationship between feature vectors and its class labels. A natural question that arises is: can we combine multiple classification algorithms to achieve higher performance than a single algorithm? This is the idea behind a class of methods called ensemble methods. Ensemble method is a method that combines models to obtain lower error rate than using a single model. "Models" in ensemble methods could include not only the implementation of many different learning algorithms, or the

X. Wang et al. (Eds.): ICMLC 2014, CCIS 481, pp. 3–12, 2014.
DOI: 10.1007/978-3-662-45652-1_1

creation of larger training set for the same learning algorithm, but also generating generic classifiers in combination to improve efficiency of classification task [13].

In this paper, we build an ensemble system where the prediction framework is formed by combining outputs of different classifiers (called meta-data or Level1 data). One of the most popular ensemble methods is based on Stacking [1-3]. Output of the Stacking proces is posterior probability that each observation belongs to a class according to each classifier. The set of posterior probability of all observations is called meta-data or Level1 data.

Le us denote N to be the number of observations, while K is a number of base classifiers and M stands for the number of classes. For an observation X_i, $P_k(\mathrm{W}_j \mid X_i)$ is 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 $\left\{P_k(\mathrm{W}_j \mid X_i)\right\}$ $j = \overline{1, M}$ $k = \overline{1, K}$ $i = \overline{1, N}$ is given by:

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

Level1 data of an observation X is defined in the form:

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

Based on stacking, various combining algorithms have been introduced. For example, Multiple Response Linear Regression (MLR) [3], Decision Template [2], SCANN [5] are well-known combining classifiers algorithm. Recently, Zhang and Zhou [6] used linear programming to find weight that each classifier puts wording on a particular class. Sen et al. [7] introduced a model inspired by MLR by applying hinge loss function to the combiner instead of using conventional least square loss. Stacking-based algorithms are called trainable algorithms since Level1data of training set is again exploited to discover latent knowledge during the second training process.

On the other hand, fixed rule is simple and effective combining classifiers method in practice. Kittler et al. [4] presented six rules named Sum, Product, Vote, Min, Max and Average. A benefit of fixed rules is that no training on Level1 data is required. Sum and Vote rules are popular combining strategy. An issue related to fixed rule is that we cannot know a priori what rule is appropriate for a specific data source.

In this work, we focus on GMM as a classifier for Level1 data. Li et al. [11] proposed a GMM classifier based on low dimensional feature space for hyper-spectral image classification. Liu et al. [9] showed that when dimension of data is high, the effectiveness of GMM approximation is reduced. To address this problem, the dimension of data for GMM is reduced by:

$$\underset{\text{likelihood function}}{P(\mathbf{x}\mid\Theta)} = P_F(\mathbf{x}\mid\Theta)\times P_{\overline{F}}(\mathbf{x}\mid\Theta) = P(\mathbf{y}\mid\Theta^*)\times P_{\overline{F}}(\mathbf{x}\mid\Theta) \tag{3}$$

where Θ is model for input data and Θ^* is another model for projected data $\mathbf{y}$ by applied PCA method on input data. F and $\overline{F}$ in turn are principal subspace containing principal component and its orthogonal complement, respectively.

The rest of this paper is organized as followed. Section 2 describes the proposed GMM based combining classifiers model that operates on Level1 data. Experimental results on 21 common UCI datasets [12] are presented in Section 3. Finally, conclusion and future work are given.

2 The Proposed Model

2.1 Combining Classifiers Based on GMM

In our knowledge, all GMM-based approaches are conducted on Level0 data (i.e. original data) in which they suffer from severe limitations in modeling various datasets. Attributes in Level0 data are frequently diverse in measurement unit and type. As a result, GMM may not approximate distribution of Level0 data well. Level1 data, on the other hand, can be viewed as scaled result from feature domain to probability domain. Observations belonging to the same class would have nearly equal posterior probability from the base classifier. As a result, they would be located close to each other. Besides, in some situations, Level1 data has lower dimension than Level0 data. It is well known that the higher the dimension of data is, the lower the effectiveness of GMM approximation. Hence, GMM on Level1 data is expected to have better performance.

This paper presents a classifier fusion technique that apply GMM on meta-data. The novel combining classifiers model is illustrated in Fig 1. First, Stacking is applied to the training set to generate Level1 data (1). Next, observations that belong to the same class are grouped together, and the class distribution is approximated by GMM.

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

$$\underset{\text{posteriror}}{P(GMM_i\mid\mathbf{x})} \sim \underset{\text{likelihood}}{P(\mathbf{x}\mid GMM_i)}\times \underset{\text{prior}}{P(GMM_i)} \tag{4}$$

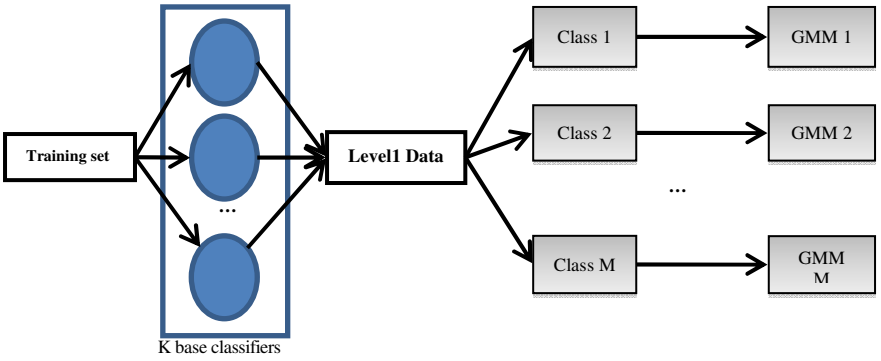

Fig. 1. GMM-based approach on Level1 data

Here likelihood function is GMM:

$$P(\mathbf{x} \mid GMM_i) = P(\mathbf{x} \mid \boldsymbol{\mu}_{ip}, \boldsymbol{\Sigma}_{ip}, \omega_{ip}) = \sum_{p=1}^{P_i} \omega_{ip} N\left(\mathbf{x} \mid \boldsymbol{\mu}_{ip}, \boldsymbol{\Sigma}_{ip}\right) \tag{5}$$

where

$$N\left(\mathbf{x} \mid \boldsymbol{\mu}_{ip}, \boldsymbol{\Sigma}_{ip}\right) = \frac{1}{(2\pi)^{MK/2} \left|\boldsymbol{\Sigma}_{ip}\right|^{1/2}} \exp\left\{-\frac{1}{2}\left(\mathbf{x}-\boldsymbol{\mu}_{ip}\right)^{\mathrm{T}} \boldsymbol{\Sigma}_{ip}^{-1}\left(\mathbf{x}-\boldsymbol{\mu}_{ip}\right)\right\} \tag{6}$$

P_i is number of Gaussian components in GMM_i model, ω_{ip}, $\boldsymbol{\mu}_{ip}$, $\boldsymbol{\Sigma}_{ip}$ are mixing coefficient, mean and covariance of p^{th} component in model for i^{th} class respectively. Prior probability in (4) of i^{th} class is defined by:

$$P(GMM_i) = N_i / N \tag{7}$$

where N_i is number of observations in i^{th} class. To find parameters of GMMs, we apply Expectation Maximization (EM) algorithm by maximize the likelihood function with respect to means, covariances of components and mixing coefficients [10].

A question related to GMM is how to find the number of components. Frequently, it is user defined. Here, we propose applying Bayes Criterion Information (BIC) to find the optimal model [10]. Given a set of GMMs $\{F_j\}$ with parameters $\boldsymbol{\theta}_j$ (which are means, covariances of components and mixing coefficients of GMM), to find best model by BIC, we compute:

$$\ln P(\mathbf{x} \mid F_j) \approx \ln P(\mathbf{x} \mid F_j, \boldsymbol{\theta}_{MAP}) - \frac{1}{2}\left|\boldsymbol{\theta}_j\right| \ln N_i \tag{8}$$

where $\boldsymbol{\theta}_{MAP}$ maximizes the posterior distribution and $|\boldsymbol{\theta}_j|$ is number of parameters in $\boldsymbol{\theta}_j$. Note that Level1 data conveys posterior information from each classifier about the amount of supports by a classifier for an observation belonging to a class. Sometimes, there are columns in Level1 data in which $\exists k,m$ such $P_k(W_m \mid X_i)$ is nearly constant for all i. In this case, the covariance matrix is singular and EM is unable to solve for GMM. To overcome this problem, we randomly choose several elements in such columns and perturb their values by a small quantity before EM is applied.

Given the GMM model for each class, the class label of an observation *XTest* is given by:

$$XTest \in W_t \text{ if } t = \arg\max_{i=\overline{1,M}} P(GMM_i \mid XTest) \tag{9}$$

Algorithm: GMM for combining classifiers

```
Training process:
Input: Training set: L0, K base classifiers,
PiMax: maximum number of Gaussian component for
i^th class.
Output: Optimum GMM for each class.
Step1: Applied Stacking to generate Level1 data.
Step2: Group Level1 data into M classes based
on class information. Compute P(GMM_i) (7),
mean, and covariance for each class.
Step3:For i^th class
   For p=1 to PiMax
      Apply EM algorithm to find GMM with p
components and compute BIC (8).
   End
   Select p that resulted in max(BIC) and
associated GMM.
End

Test process:
Input: Unlabeled observation XTest.
Output: Predicted label of XTest
Step1: Compute Level1 data of XTest
Step2: For each class
Compute P(XTest|GMM_i) (5) and then posterior
P(GMM_i|XTest)
End
Step3: Predict label of XTest from (9)
```

2.2 GMM-PCA Model

When the number of observations is smaller than the dimension of data, GMM cannot be estimated by EM algorithm. Hence, the dimension of data needs to be reduced. We perform PCA on Level1 data and retain only the C largest eigenvalues which satisfies (10):

$$\left(\sum_{c=1}^{C}\lambda_c\right)\left(\sum_{c=1}^{MK}\lambda_c\right)^{-1} > 1-\varepsilon \tag{10}$$

Then, $P(\mathbf{x}\,|\,GMM_{\mathbf{i}}) \sim P(\mathbf{y}\,|\,GMM_{\mathbf{i}})$ where y is the projection of x on principle subspace which contains the C selected eigenvectors. Given the GMM model for each class, the class label of an observation *XTest* is given by:

$$XTest \in W_t \quad \text{if} \quad t = \arg\max_{i=\overline{1,M}} P(YTest\,|\,GMM_{\mathbf{i}}) \times P(GMM_{\mathbf{i}}) \tag{11}$$

3 Experimental Results

In our experiments, we performed 10-fold cross validation on the dataset. Moreover, to ensure objectiveness, the test was run 10 times so we had 100 error rates result for each data file. Three base classifiers, namely Linear Discriminant Analysis (LDA), Naïve Bayes and K Nearest Neighbor (with K set to 5 denoted by 5-NN) were selected. 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 results (parameter α is set by 0.05)

In our assessment, we compared error rate of our model and five other methods: best result from the set of base classifiers, best result based on fixed combining rules, SCANN, MLR, Decision Template (with measure of similarity S_1 [2] defined as $S_1(Level1(X), DT_i) = \dfrac{\|Level1(X) \cap DT_i\|}{\|Level1(X) \cup DT_i\|}$ where DT_i is Decision Template of i^{th} class and $\|\alpha\|$ is the relative cardinality of the fuzzy set α), GMM on Level0 data,. Here we used 6 fixed rules namely Sum, Product, Min, Max, Vote, Median to choose the best result based on their outcome on test set. Experimental results of 21 UCI files are showed in Table 2, 3 and 4.

In Table 2, we reported error rate of all 3 base classifiers and chose best result based on their performance on test set. We see that both GMM and GMMPCA-based approach on meta-data outperform any base classifiers. GMM posts 6 wins and only 3 losses while GMMPCA has 7 wins and 3 losses (Table 5).

GMM and GMMPCA on Level1data are observed to perform better than GMM on Level0 data, posting up to 16 and 17 wins, respectively. Our model is only poorer than GMM on Level0 data on Ring files (2.09%). This is not surprising because the Ring dataset was drawn from multivariate Gaussian distributions [12], so GMM is the best to approximate Level0 distribution in that case. Clearly, GMM on Level0 reports higher error rates than those on Level1 data as well as Rules, Decision Template, MLR and SCANN.

Table 1. UCI data files used in our experiment (*) R: Real, C: Category, I: Integer

File name	# of attributes	Attribute type (*)	# of observations	# of classes	# of attributes on Level1
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	6
Vehicle	18	I	846	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

Table 2. Classifying error rate of base classifiers

File name	LDA		Naïve Bayes		5-NN		Best result from base classifiers	
	Mean	*Variance*	*Mean*	*Variance*	*Mean*	*Variance*	*Mean*	*Variance*
Bupa	0.3693	8.30E-03	0.4264	7.60E-03	0.3331	6.10E-03	0.3331	6.10E-03
Artificial	0.4511	1.40E-03	0.4521	1.40E-03	0.2496	2.40E-03	0.2496	2.40E-03
Pima	0.2396	2.40E-03	0.2668	2.00E-03	0.2864	2.30E-03	0.2396	2.40E-03
Sonar	0.2629	9.70E-03	0.3042	7.40E-03	0.1875	7.60E-03	0.1875	7.60E-03
Heart	0.1593	5.30E-03	0.1611	5.90E-03	0.3348	5.10E-03	0.1593	5.30E-03
Phoneme	0.2408	3.00E-04	0.2607	3.00E-04	0.1133	2.00E-04	0.1133	2.00E-04
Haberman	0.2669	4.50E-03	0.2596	4.40E-03	0.2829	3.80E-03	0.2596	4.40E-03
Titanic	0.2201	5.00E-04	0.2515	8.00E-04	0.2341	3.70E-03	0.2201	5.00E-04
Balance	0.2917	2.90E-03	0.2600	3.30E-03	0.1442	1.20E-03	0.1442	1.20E-03
Fertility	0.3460	2.01E-02	0.3770	2.08E-02	0.1550	4.50E-03	0.1550	4.50E-03
Skin&NonSkin	0.0659	2.74E-06	0.1785	6.61E-06	0.0005	1.68E-08	0.0005	1.68E-08
Wdbc	0.0397	7.00E-04	0.0587	1.20E-03	0.0666	8.00E-04	0.0397	7.00E-04
Australian	0.1416	1.55E-03	0.1297	1.71E-03	0.3457	2.11E-03	0.1297	1.71E-03
Twonorm	0.0217	3.12E-05	0.0217	3.13E-05	0.0312	3.96E-05	0.0217	3.12E-05
Magic	0.2053	6.85E-05	0.2255	7.33E-05	0.1915	4.81E-05	0.1915	4.81E-05
Ring	0.2381	2.27E-04	0.2374	2.23E-04	0.3088	1.30E-04	0.2374	2.23E-04
Tae	0.4612	1.21E-02	0.4505	1.22E-02	0.5908	1.37E-02	0.4505	1.22E-02
Contraceptive	0.4992	1.40E-03	0.5324	1.42E-03	0.4936	1.70E-03	0.4936	1.70E-03
Vehicle	0.2186	1.39E-03	0.5550	2.94E-03	0.3502	2.35E-03	0.2186	1.39E-03
Iris	0.0200	1.40E-03	0.0400	2.30E-03	0.0353	1.50E-03	0.0200	1.40E-03
Letter	0.2977	8.31E-05	0.4001	1.04E-04	0.0448	1.68E-05	0.0448	1.68E-05

Besides, our approach is competitive with best result selected from fixed rules (Table 5). The files in which we have superior performance are: Ring (11.31% vs. 21.22%), Vehicle (21.66% vs. 26.45%) and Skin&NonSkin (0.04% vs. 0.06%), while on 4 files, best result from fixed rules is better than our algorithm. Note, however, that the optimal rule for a particular data source is usually not known in advanced.

Table 3. Classifying error rate of trainable combining algorithms

File name	MLR		Best result from 6 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-04	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

Table 4. Classfying error rate of GMM-Based approaches

File name	GMM on Level0		GMM on Level1		GMM PCA on Level1	
	Mean	*Variance*	*Mean*	*Variance*	*Mean*	*Variance*
Bupa	0.4419	5.80E-03	0.3022	5.31E-03	0.3176	5.49E-03
Artificial	0.4507	8.00E-03	0.2374	2.40E-03	0.2329	1.66E-03
Pima	0.2466	2.40E-03	0.2432	2.60E-03	0.2158	8.70E-03
Sonar	0.3193	1.26E-02	0.2009	6.20E-03	0.1974	6.90E-03
Heart	0.1715	7.30E-03	0.1559	4.51E-03	0.1600	5.43E-03
Phoneme	0.2400	4.00E-04	0.1165	2.01E-04	0.1161	1.72E-04
Haberman	0.2696	2.00E-03	0.2458	3.36E-03	0.2491	2.40E-03
Titanic	0.2904	2.01E-02	0.2167	5.91E-04	0.2183	7.83E-04
Balance	0.1214	1.10E-03	0.0839	1.21E-03	0.0783	1.10E-03
Fertility	0.3130	7.47E-02	0.1850	1.05E-02	0.1250	2.50E-03
Skin&NonSkin	0.0761	2.21E-06	4.10E-04	1.53E-08	0.0004	1.60E-08
Wdbc	0.0678	1.10E-03	0.0387	5.98E-04	0.0397	6.97E-04
Australian	0.1980	1.80E-03	0.1222	1.30E-03	0.1233	1.20E-03
Twonorm	0.0216	2.83E-05	0.0219	2.78E-05	0.0219	2.72E-05
Magic	0.2733	5.06E-05	0.1921	8.34E-05	0.1923	7.93E-05
Ring	0.0209	2.20E-05	0.1131	1.16E-04	0.1131	9.98E-05
Tae	0.5595	1.39E-02	0.4365	1.36E-02	0.5132	1.67E-02
Contraceptive	0.5306	1.80E-03	0.4667	1.30E-03	0.4671	1.70E-03
Vehicle	0.5424	2.40E-03	0.2166	1.40E-03	0.2132	1.80E-03
Iris	0.0453	2.50E-03	0.0360	2.10E-03	0.0400	3.02E-03
Letter	0.3573	9.82E-05	0.0797	3.03E-05	0.0834	2.98E-05

Our results also show that GMM on Level1 data outperforms Decision Template, posting 10 wins and 0 loss. Superior results are reported on Bupa (30.22% vs. 33.48%), Haberman (24.58% vs. 27.79%), Fertility (18.5% vs. 45.2%), Skin&NonSkin (0.04% vs. 3.32%), Ring (11.31% vs. 18.94%) and Letter (7.97% vs. 11.33%).

GMM on Level1 data is also better than SCANN (5 wins and 1 loss). Likewise, GMMPCA is better than SCANN with 5 wins and 2 losses. We note that SCANN cannot be performed on 3 files, Skin, Balance and Fertility, due to the existence of equal column in the indicator matrix, which resulted in singular matrix [5]. Compared with MLR, both our approaches are equally competitive since both GMM on Level1 data and GMMPCA have 4 wins and 4 losses.

Table 5. Statistical tests compare GMM, GMMPCA with the Benchmarks

	Better	Competitive	Worse
GMM PCA Level1 vs. SCANN	5	11	2
GMM Level1 vs. SCANN	5	12	1
GMM PCA Level1 vs. Decision Template	11	9	1
GMM Level1 vs. Decision Template	10	11	0
GMM PCA Level1 vs. MLR	4	13	4
GMM Level1 vs. MLR	4	13	4
GMM PCA Level1 vs. best result from fixed rules	6	11	4
GMM Level1 vs. best result from fixed rules	5	13	3
GMM PCA Level1 vs. best result from base classifiers	7	11	3
GMM Level1 vs. best result from base classifiers	6	12	3
GMM PCA Level1 vs. GMM Level0	17	3	1
GMM Level1 vs. GMM Level0	16	4	1
GMM Level1 vs. GMM PCA Level1	2	17	2

Finally, we observe that the classification accuracy of GMM-PCA and GMM on Level1 data is comparable. However, since GMM-PCA can deal with situation when the number of observations is smaller than the dimension of data, it would be more applicable in many situations.

4 Conclusion and Future Work

We have introduced a novel approach which used GMM on Level1 data to combine results from base classifiers in a multi-classifier system. Experimental results on 21 UCI files have demonstrated the superiority of our method compared with several state-of-art combining algorithms. Future work will be to apply classifier and feature selection methods to further increase the classification accuracy of our approach.

References

1. Wolpert, D.H.: Stacked Generalization. Neural Networks **5**(2), 241–259 (1992)
2. Kuncheva, L.I., Bezdek, J.C., Duin, R.P.W.: Decision Templates for Multi Classifier Fusion: An Experimental Comparison. Pattern Recognition **34**(2), 299–314 (2001)
3. Ting, K.M., Witten, I.H.: Issues in Stacked Generation. Journal of Artificial In Intelligence Research **10**, 271–289 (1999)
4. Kittler, J., Hatef, M., Duin, R.P.W., Matas, J.: On Combining Classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence **20**(3) (1998)
5. Merz, C.: Using Correspondence Analysis to Combine Classifiers. Machine Learning **36**, 33–58 (1999)
6. Zhang, L., Zhou, W.D.: Sparse ensembles using weighted combination methods based on linear programming. Pattern Recognition **44**, 97–106 (2011)
7. Sen, M.U., Erdogan, H.: Linear classifier combination and selection using group sparse regularization and hinge loss. Pattern Recognition Letters **34**, 265–274 (2013)
8. Moghaddam, B., Pentland, A.: Probabilistic Visual Learning for Object Representation. IEEE Transactions on Pattern Analysis and Machine Intelligence **19**(7) (July1997)

9. Liu, X.H., Liu, C.L.: Discriminative Model Selection for Gaussian Mixture Models for Classification. In: First Asian Conference on Pattern Recognition (ACPR), pp. 62–66 (2011)
10. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer Press (2006)
11. Li, W., Prasad, S., Fowler, J.E.: Hyperspectral Image Clasfication Using Gaussian Mixture Models and Markov Random Fields. IEEE Geoscience and Remote Sensing Letter **11**(1) (January 2014)
12. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml/datasets.html
13. Rokach, L.: Taxonomy for characterizing ensemble methods in classification tasks: A review and annotated bibliography. Journal of Computational Statistics & Data Analysis **53**(12), 4046–4072 (2009)