2014 International Conference on Machine Learning and Cybernetics, ICMLC2014, Lanzhou, China, 13-16 July 2014

Combining classifiers based on GMM approach on ensemble data

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

1 - School of 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, Albany, New Zealand

Email: tienthanh.nguyen2@griffithuni.edu.au

Abstract. Combining multiple classifiers to achieve better performances 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 building a powerful model which is competitive with other well-known combining algorithms such as Decision Template, Multiple Response Linear Regression (MLR), SCANN and fixed combining rules. It is difference from the traditional approaches, here we use Gaussian Mixture Model (GMM) to model distribution of Level1 data and predict label of an interesting observation based on maximize of posterior probability through Bayes model. We also expand GMM-based approach in which before modeling distribution, Principle Component Analysis (PCA) method is applied to output of base classifiers to reduce its dimension; as a result, improve performance and availability of model based GMM. Experiments were evaluated on 21 UCI Machine Learning Repository demonstrate benefits of our framework compared with benchmarks.

Keywords: Gaussian Mixture Model (GMM), ensemble method, multiple classifier system, combining classifiers, classifier fusion, Stacking Algorithm, Principle Component Analysis (PCA).

1 INTRODUCTION

Traditionally, single learning algorithm is usually employed to solve classification problems by forming a classifier on a particular training set which contains hypothesis about the relationship between feature vectors and its class labels. A nature question is arisen that can we combine multiple algorithms in a system to achieve more effectiveness and higher performance than any single one? That is the idea to form a class of methods called ensemble method. Ensemble in literature review is a method that combines models to obtain lower error rate than using single model. Concept “model” in definition of ensemble methods is understood in a broad sense, including not only the implementation of many different learning algorithms, or the creation of more training set scheme for same learning algorithm, but also generating generic classifiers in combination to improve efficiency of classification task [23].

In this paper, we base on the strategy to build an ensemble system [1] where prediction framework is formed by combining outputs of different classifiers (called meta-data or Level1 data to distinguish with original data). It is expected that base classifiers should be significantly different to each other. Moreover, training set is shared among all base classifiers as it is trained by all base classifiers to give posterior probability corresponding to each class and classifier. Actually, several well-known algorithms were introduced and have been applied successfully to various data sources. They all have the same target by discovering knowledge from meta-data to construct the decision system. Here we continue exploiting data based on different perspective by studying related to statistical representation for meta-data. Specifically, we choose Gaussian-based mixture models to approximate distribution of Level1 data corresponding with each class. GMM as a linear combination of multiple Gaussian components can result a better representation for an arbitrary density function [14]. By using GMM to approximate likelihood function on meta-data, prediction framework based on posterior probability is introduced though Bayes model.

The rest of this paper is organized as followed. Section 2 gives the literature review on both some state-of-art combining classifiers methods based Staking algorithm and popular combining methods by using fixed rules. After that, several approaches in which GMM plays as a classifier are mentioned. In Section 3, the novel combining classifiers model is proposed by using GMM directly as well as applying PCA to Level1 data before. Experimental results conducted on 21 common UCI Machine Learning Repository datasets [21] are illustrated and discussed in Section 4. Finally, conclusion and future work are given in the last section.

2 RECENT WORK

2.1 Combining classifiers algorithms

There are several combining classifiers strategies proposed and Stacking-based approaches are one of the most popular ensemble methods. The Stacking algorithm was first introduced by Wolpert [2] and was further developed by Ting [4]. In this algorithm, training set is divided to several disjoint parts and then each plays as test set and the others are gathered in a new single training set; as a result, all observations will be tested one time. Output of Stacking family is Fuzzy Label [3] or in other words is posterior probability that each observation belongs to a class according to each classifier. Posterior probability set of all observations is called meta-data or Level1 data with the aim of distinguishing with original Level0 data.

Let’s denote N as number of observations, K as number of base classifiers and M as 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 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 detailed in form:

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

Level1 data of an observation X is defined:

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

Based on Stacking algorithm, various combining algorithms have been introduced with the purpose of reducing error rate of classification task. Stacking-based algorithms are called trainable algorithms since Level1 of training set is again exploited to discover latent knowledge as the second training process. Specifically, Ting [4] proposed Multiple Response Linear Regression algorithm (MLR) to combine posterior probabilities of each observation based on sum of weights calculated from K Linear Regression functions. Kuncheva [3] applied Fuzzy Relation to find a relationship between posterior probability matrix (eqn. 2) and Decision Template for each class computed on posterior probability of observations and its true class label. She also detailed 11 measurements between two fuzzy relations [3] so as to predict about label of interested observations. Metz [6] combined Stacking, Correspondence Analysis (CA) and K Nearest Neighbor in a single algorithm called SCANN in which CA method is used to analyze relationship between rows (include all observations) and columns (include outputs of Stacking in Crisp Label type [3] and true label of each observation) to form the new representation of outputs from base classifiers. After that, KNN is applied to that representation to have prediction for unlabeled observations. Recently, Zhang [10] used linear programming to find weight that each classifier puts on a particular class. Sen [11] introduced a model inspired by MLR by applying 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.

On the other hand, fixed rule is simple and effective combining classifiers method in practice. Kittler [5] presented six 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 those of base classifiers. Another benefit of fixed rules is that they only work on Level1 data of unlabeled observation; as a result; computational cost is saved significantly. Frequently, Sum and Vote rule are selected in combining strategy although issue related to fixed rules is that we cannot know what rule is appropriate for a specific data source.

2.2 GMM Classifier

Although Gaussian distribution is the widely approximation for density model, it has some remarkable limitations. One of the most significant problems with this distribution is intrinsically uni-model so it cannot be flexible to capture a wide range of distribution. Meanwhile, GMM as a linear combination of multiple Gaussian components is better approximation to a distribution than single Gaussian. GMM has been widely used in a variety of practical applications such as skin color extraction, speed recognition and image retrieval [16, 17]. Here we focus on GMM as a classifier. Li [18] proposed GMM-Markov Random Field classification: a GMM classifier based on low dimensional feature space for hyper-spectral image classification. Liu [13] showed that when dimension of data is high, the effectiveness of GMM approximation is reduced so as to deteriorate classification accuracy. Based on this address, he introduced a discriminative model selection for GMM for classification by applying result from Moghaddam [12] where a technique was proposed to reduce dimension of data for GMM given by:

$$\underbrace{P(\mathbf{x} \mid \Theta)}_{likelihood\ function} = P_F(\mathbf{x} \mid \Theta) \times P_{\overline{F}}(\mathbf{x} \mid \Theta) = P(\mathbf{y} \mid \Theta^*) \times P_{\overline{F}}(\mathbf{x} \mid \Theta) \qquad (3)$$

where Θ is model for input data and Θ^* is another model for projected data $\mathbf{y}$ which detailed below. Moghaddam [12] introduced a method to obtain basic function of Karhunen–Loève transformation (KLT) by solving eigenvalue problem $\mathbf{\Sigma} = \phi \mathbf{\Lambda} \phi^T$ in which $\mathbf{\Sigma}$ is covariance matrix computed from input data; ϕ are eigenvectors associated with eigenvalues $\{\lambda_i\}$ and $\mathbf{\Lambda} = diag(\lambda_i)$ is diagonal matrix with of eigenvalues. In Principle Component Analysis (PCA), a partial KLT is performed by using several eigenvectors corresponding with largest eigenvalues. Hence, vector $\mathbf{y}$ in eqn. 3 is projection of $\mathbf{x}$ on principle subspace that $\mathbf{y} = \phi_P^T \left(\mathbf{x} - \overline{\mathbf{x}} \right)$ in which $\overline{\mathbf{x}}$ is mean value of $\mathbf{x}$ and ϕ_P is submatrix of ϕ including P retained eigenvectors. F and $\overline{F}$ in turn are principal subspace $F = \{\phi_i\}_{i=\overline{1,P}}$ containing principal component and its orthogonal complement $F = \{\phi_i\}_{i=\overline{P+1,...}}$ respectively.

In our research, we propose applying GMM to model distribution of outputs of base classifiers. We note that several significant changes from original GMM classifier are needed to adapt with characteristics of Level1 data. Our aim is introducing a novel framework which is competitive with GMM on its counterpart Level0, best result from base classifiers, best result from fixed rules and other well-known combining algorithms like Decision Template, SCANN and MLR.

3 THE PROPOSED MODEL

3.1 Combining Classifiers based on GMM

In our knowledge, all GMM-based approaches are conducted on Level0 in which they suffer from remarkable limitations in modeling various datasets. Attributes in Level0 are frequently diverse, measurement unit and type; as a result, GMM cannot achieve effectiveness when it is selected to approximate distribution of Level0. Level1, otherwise, can be viewed as scaled result from feature domain to posterior domain where data is reshaped by posterior probability as all Level1 attributes have same real value domain [0, 1]. Observations belonged to same class may have nearly equal posterior probability values resulted by base classifier; as a result, may be located nearly in new coordination system. It is hoped that Level1 will have more discriminative than original data and therefore GMM on Level1 will be more effective than on Level0. Besides, in some situations, Level1 has lower dimension than Level0. It is well known in literature that the higher dimension of data is, the lower effectiveness of GMM approximation is. Hence, applying GMM on Level1 improves itself effectiveness.

As mentioned above, we pursue exploiting knowledge on meta-data to form hypothesis about the relationship between feature vector and its class label. This paper presents a technique for effectively addressing classifier fusion issue by applying GMM on meta-data. The novel combining classifiers model is illustrated in Figure 1. Firstly, training set is divided and classified by using Stacking Algorithm to generate its Level1 data (eqn. 1). Next, since label of each observation in training set is known so observations belonged to same class are grouped together. Here we put attention on building model for each class by the way that approximate distribution by GMM. Unlabeled observation, in turn, is classified by base classifiers with model generated on training set to output its Level1 data (eqn. 2). That Level1 is gone through M-GMMs as input data to obtain final prediction.

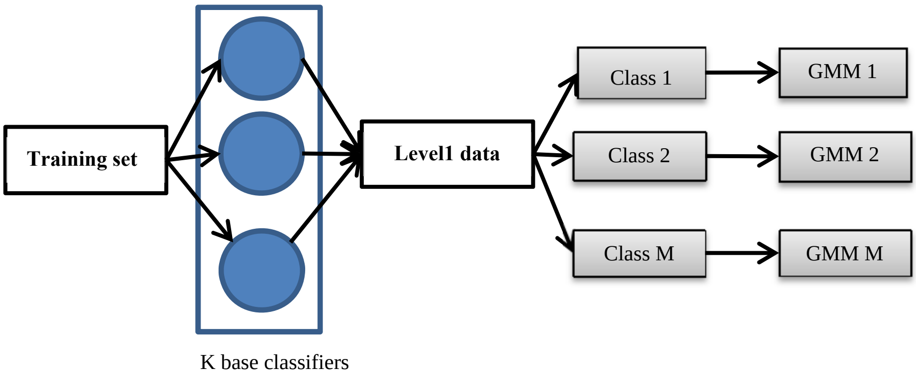


Fig. 1. GMM-based approach on Level1 data

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

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

Here likelihood function is GMM:

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

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 and $\boldsymbol{\mu}_{ip}$, $\boldsymbol{\Sigma}_{ip}$ are mean and covariance of p^{th} component in model for i^{th} class respectively. Prior probability in (eqn. 4) of i^{th} class is defined by:

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

where N_i is number of observations in i^{th} class.

Now with dataset $\mathbf{X} = \{\mathbf{x_i}\}$ $i = 1, N_i$ in which $\mathbf{x_i}$ has identity and independence distribution, logarithm of likelihood function is given by:

$$\ln P(\mathbf{X} \mid GMM_i) = \ln\left\{\prod_{i=1}^{N_i} P(\mathbf{x}_i \mid \boldsymbol{\mu}_{ip}, \boldsymbol{\Sigma}_{ip}, \omega_{ip})\right\} = \sum_{i=1}^{N_i} \ln\left\{\sum_{p=1}^{P_i} \omega_{ip} N\left(\mathbf{x}_i \mid \boldsymbol{\mu}_{ip}, \boldsymbol{\Sigma}_{ip}\right)\right. \tag{8}$$

It is worth noting that summation appears inside logarithm (eqn. 8) results in complicated expression for the maximum likelihood solution. Hence, 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 [14].

The other question related to GMM is how to find the number of components. Frequently, it is fixed by a specific number. Here, we propose applying Bayes Criterion Information (BIC) to find optimal model [14]. By assuming that we have a set of model $\{F_j\}$ (which are all available GMMs associated with number of component) with parameters $\boldsymbol{\theta}_j$ where $\boldsymbol{\theta}_j$ are denoted for all parameters of model (which are means, covariances of components and mixing coefficients in scenario of GMM). To find model by BIC, we compute:

$$\ln P(\mathbf{X} \mid F_j) \approx \ln P(\mathbf{X} \mid F_j, \boldsymbol{\theta}_{MAP}) - \frac{1}{2}\left|\boldsymbol{\theta}_j\right| \ln N_i \tag{9}$$

where $\boldsymbol{\theta}_{MAP}$ is corresponding with maximum of posterior distribution and $\left|\boldsymbol{\theta}_j\right|$ is number of parameters in $\boldsymbol{\theta}_j$. It is interesting to note that Level1 conveys posterior information from each classifier that how much supports by a classifier for an observation belonged to a class. Sometimes, there are several columns in Level1 data in which $\exists k, m$ such $P_k(W_m \mid X_i)$ is nearly constant for all i. Hence, covariance matrix may be singular and EM is unable to solve GMM. We propose a produce with the purpose of regularizing Level1 by before applying GMM to Level1, we check condition (eqn. 10) on all columns. If the condition is satisfied, few random elements are chosen in this column and increate their values by a small quantity. It called a regularization produce for Level1. The produce only adds small value on some random elements in a column so it does not affect the nature of interested posterior probability as well as covariance matrix. Moreover, it reduces singular situation of covariance matrix so helps overcome this situation.

$\left|x-\bar{x}\right| < \varepsilon \ \forall$ column vector x and small threshold value ε (10)

Where $\bar{x}$ is mean value of x

```
Algorithm 1: Regularize Level1
Input: Level1, extravalue, r
Output: Regularized Level1
For $i^{th}$ column of Level1
     If Condition(eqn. 10) = true
          Generate r random numbers in (1,size(column))
          Element(r) = Element(r)+ extravalue
     End if
End
Return Level1
```

```
Algorithm 2: GMM for combining classifiers

Training progress:
Input: Training set: L0, K base classifiers, PiMax: maximum number of Gaussian component for $i^{th}$ class.
Output: Suitable GMM for each class.
Step1: Applied Stacking algorithm to generate Level1 of L0.
Step2: Gather same labeled observations in M classes; compute $P(GMM_i)$ (eqn. 7), mean and covariance for each class.
Step3:For $i^{th}$ class
          Call Algorithm 1 to regularize Level1 of class
          For p=1 to PiMax
               Apply EM algorithm to find GMM corresponding with p components.
               Compute BIC.
          End
     Select Pi corresponding with max(BIC) and GMM with Pi components.
     End
Test progress:
Input: unlabeled observation XTest
Output: predicted label of XTest
Step1: Compute Level1 of XTest by model from K classifiers and training set.
Step2: For each $i^{th}$ class
        Compute $P(XTest\,|\,GMM_i)$ (eqn. 5) and posterior related to class (eqn. 4)
                $P(GMM_i\,|\,XTest) \sim P(XTest\,|\,GMM_i) \times P(GMM_i)$
        End
Step3:   Predict   label   of   XTest   based   on   $XTest \in W_t$   if
$t = \arg\max_{i=1,M} P(GMM_i \mid XTest)$
```

3.2 GMM-PCA model

Another problem related to GMM is small number of observations in a class. Through preliminary experiment conduced on Matlab2013a we have seen that when number of observations is smaller than dimension of data, GMM cannot be estimated by EM algorithm. Hence, dimension of data needs to be reduced to be availability for diversity of data sources. Putting attention on eqn.3 in which likelihood $P(\mathbf{x} \mid \Theta)$ is analyzed to $P(\mathbf{y} \mid \Theta^*) \times P_{\bar{F}}(\mathbf{x} \mid \Theta)$, Moghaddam [12] proposed $\hat{P}_{\bar{F}}(\mathbf{x} \mid \Theta)$ as an estimation of $P_{\bar{F}}(\mathbf{x} \mid \Theta)$ by using spherical Gaussian:

$$\hat{P}_{\bar{F}}(\mathbf{x} \mid \Theta) = \frac{1}{(2\pi\rho)^{\frac{d-k}{2}}} \exp\left\{ \frac{-\varepsilon^2(\mathbf{x})}{2\rho} \right\}$$

$$\rho = \frac{1}{d-k} \sum_{l=k+1}^{d} \lambda_l \qquad (11)$$

$$\varepsilon^2(\mathbf{x}) = \left\| \mathbf{x} - \bar{\mathbf{x}} \right\|^2 - \sum_{l=1}^{k} y_l^2 = \sum_{l=k+1}^{d} y_l^2$$

where d is dimension of Level0, k is number of selected elements and $\varepsilon^2(\mathbf{x})$ is square of all unselected principle component feature vector in eigenspace.

It is noted that Level1 may include several nearly equal 0 elements due to the fact that an observation is predicted not belonged to a class. Hence, when PCA is performed on Level1, few eigenvalues will be nearly equal 0, consequently, $\rho \to 0$ and results in $\hat{P}_{\bar{F}}(\mathbf{x} \mid \Theta) \to \infty$. Here we leave this strategy and instead eigenvectors are ranked based on their associated eigenvalue and only the first largest C eigenvalues which satisfies condition (eqn. 12) are retained while the others are discarded:

$$\frac{\sum_{c=1}^{C} \lambda_c}{\sum_{c=1}^{MK} \lambda_c} > 1 - \varepsilon \qquad (12)$$

When several components are reduced, (eqn. 3) simply becomes $P(\mathbf{x} \mid GMM_{\mathrm{i}}) \sim P(\mathbf{y} \mid GMM_{\mathrm{i}})$ where y is the projection of x on principle subspace which contains C selected eigenvectors. Now with unlabeled observation $XTest$, we predict its class label by:

$$XTest \in W_t \text{ if } t = \arg\max_{i=1,M} P(YTest \mid GMM_{\mathrm{i}}) \times P(GMM_{\mathrm{i}}) \qquad (13)$$

4 EXPERIMENTAL RESULTS

There are two circumstances related to training set and test set in experiments. Firstly, we only had a single dataset, and did not have individual test set. To solve this case, B-fold cross validation method was employed by the way that dataset is divided to nearly equal disjointed B parts. Each part plays as a test set one time while the others play as a training set. In our experiments, we set B=10 (10-fold cross validation). Secondly, we had both training set and test set separately. It is traditional scenarios because we can perform classification by classical approach. To ensure objectiveness, we tried to run the test 10 times so we had up to 100 error rates result for each file. Statistical Toolbox in Matlab2013a was chosen as environment to develop our model. Besides, 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 by the motivation that they all have different approach to solve classification problem so diversity of ensemble system is ensured. For comparison purpose, we used paired statistical t-test to compare two expectations (parameter α is set by 0.05)

In our assessment, we compared error rate of our model with each other among 7 methods: best result based on test set from base classifiers, best result based on test set from fixed rules, SCANN, MLR, Decision Template (with measure of similarity S_1 [3] 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), GMM on Level0, GMM [12] on Level0. 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. It is noted that combining algorithms like fixed rules, SCANN and MLR do not require any initialized parameters. Actually, we chose the benchmarks for our model due to three reasons:

- Since our model is an ensemble system so it is required to compare with all base classifiers.
- Since our model combines outputs form base classifiers to form the prediction for unlabeled observations so it is necessary to compare with other well-known trainable combining algorithms as well as simple fixed combining algorithms.
- Since our model is based on GMM classifier so it is important to compare with its counterpart on original data

We chose 21 common UCI Machine Learning Repository [21] data files from 2 classes (Bupa, Artificial, etc…) to 26 classes (Letter). Number of attributes also changes in a wide range from only 3 attributes (Haberman) to 60 attributes (Sonar). Number of observations in each file also varies considerably, from small files like Iris, Fertility to quite big file such as Skin&NonSkin (up to 245057 observations) (Table 1). Our purpose is conducting an objective experiment so as to prove advantage of novel model and algorithms on diverse data sources. Experimental results of all 21 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. By using paired t-test to compare with outcomes of GMM and GMMPCA on Leve1l, it is objective to assess that both GMM and GMMPCA-based approach on meta-data outperform any base classifiers. GMM posts 6 wins and only 3 losses while the pattern of GMMPCA is 7 wins and 3 losses. Consequently, the goal of building ensemble method which is better than any base classifiers has been achieved.

It is interesting to note that GMM and GMMPCA on Level1 perform better than GMM on Level0, posting up to 16 and 17 wins, respectively. Our model only loses GMM Level0 on Ring files (2.09%). It is not surprised because Ring dataset is withdrawn from multivariate Gaussian distributions [21] so GMM is the best to approximate Level0 distribution in that case. Clearly, GMM on Level0 and GMM-based approach [12] report higher error rates than those on Level1 as well as Rules, Decision Template, MLR and SCANN. GMM-based approach [12] is only better than ours on 3 files while up to 16 cases address outstanding performance of our approach (Figure 2).

Besides, our approach is competitive with best result selected from fixed rules (Figure 2). There are few cases reporting the outstanding performance namely Ring (11.31% vs. 21.22%), Vehicle (21.66% vs. 26.45%) and Skin&NonSkin (0.04% vs. 0.06%) while on 1 file, best result from fixed rules is better than proposed GMM Level1. Actually, we cannot know what the optimal rules are for specific data source so GMM on Level1 can be considered replacing fixed rules to combining algorithm since it is better than any popular fixed rules.

Next, we compare our model with Decision Template algorithm. Clearly, GMM on Level1 outperforms Decision Template among experimental files, posting 8 wins and only 2 losses. The remarkable 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 is also better than SCANN (5 wins and 1 loss). The same pattern is repeated when we compare GMMPCA with SCANN (5 wins and 2 losses). Unluckily, SCANN cannot be performed on 3 files Skin, Balance and Fertility because of existence equal column in indicator matrix so column masses will be singular [6]. Here we do not put these cases in comparison. Compared with MLR, both two our approaches are competitive since both GMM on Level1 and GMMPCA have 4 wins and 4 losses.

Finally, one interesting result is addressed when we compare two methods GMM and GMM-PCA. On few files, accuracy of GMM-PCA is better than GMM, for instant Fertility (1.25% vs. 18.5%) while in one Tae file, GMM is outstanding (43.65% vs. 51.32%). Besides, GMM-PCA is more availability with diverse data sources than GMM since when number of observations is smaller than dimension of data (eqn. 1), EM algorithm may not converge to a solution where one or more of the components have a singular covariance matrix. By applying PCA to GMM, dimension of data is reduced; as a result, that problem can be solved.

File name	Number of attributes	Attribute type (*)	Number of observations	Number of classes	Number 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	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

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

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

Table 2. Classifying error of base classifiers

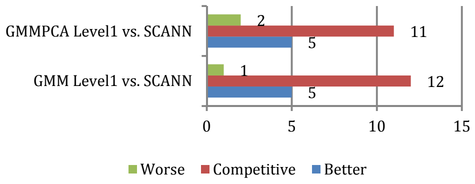
File name	MLR		Best result from 6 fixed rules		SCANN		Decision Template	
	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance
Bupa	0.3033	4.70E-03	0.2970	4.89E-03	0.3304	4.29E-03	0.3348	7.10E-03
Artificial	0.2426	2.20E-03	0.2193	2.05E-03	0.2374	2.12E-03	0.2433	1.60E-03
Pima	0.2432	2.30E-03	0.2365	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

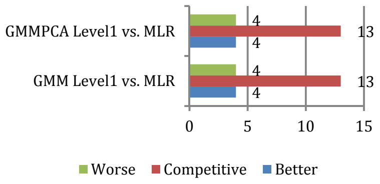
Heart	0.1607	4.70E-03	0.1570	4.64E-03	0.1637	4.14E-03	0.1541	4.00E-03
Phoneme	0.1136	1.75E-04	0.1407	1.95E-04	0.1229	6.53E-04	0.1462	2.00E-04
Haberman	0.2428	3.30E-03	0.2392	2.39E-03	0.2536	1.74E-03	0.2779	5.00E-03
Titanic	0.2169	4.00E-04	0.2167	5.00E-04	0.2216	6.29E-04	0.2167	6.00E-04
Balance	0.1225	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	0.0063	2.42E-05	0.1133	4.91E-05

Table 3. Classifying error of trainable combining algorithms

File name	GMM on Level0		GMM [12] on Level0		GMM on Level1		GMM PCA on Level1	
	Mean	Variance	Mean	Variance	Mean	Variance	Mean	Variance
Bupa	0.4419	5.80E-03	0.4064	8.00E-03	0.3022	5.31E-03	0.3176	5.49E-03
Artificial	0.4507	8.00E-03	0.4209	7.60E-03	0.2374	2.40E-03	0.2329	1.66E-03
Pima	0.2466	2.40E-03	0.3022	1.80E-03	0.2432	2.60E-03	0.2158	8.70E-03
Sonar	0.3193	1.26E-02	0.2000	7.90E-03	0.2009	6.20E-03	0.1974	6.90E-03
Heart	0.1715	7.30E-03	0.3367	6.50E-03	0.1559	4.51E-03	0.1600	5.43E-03
Phoneme	0.2400	4.00E-04	0.2136	3.00E-04	0.1165	2.01E-04	0.1161	1.72E-04
Haberman	0.2696	2.00E-03	0.2640	2.90E-03	0.2458	3.36E-03	0.2491	2.40E-03
Titanic	0.2904	2.01E-02	0.2353	2.70E-03	0.2167	5.91E-04	0.2183	7.83E-04
Balance	0.1214	1.10E-03	0.0899	1.40E-03	0.0839	1.21E-03	0.0783	1.10E-03
Fertility	0.3130	7.47E-02	0.1410	6.60E-03	0.1850	1.05E-02	0.1250	2.50E-03
Skin&NonSkin	0.0761	2.21E-06	0.0144	1.70E-05	4.10E-04	1.53E-08	0.0004	1.60E-08
Wdbc	0.0678	1.10E-03	0.0866	1.70E-03	0.0387	5.98E-04	0.0397	6.97E-04
Australian	0.1980	1.80E-03	0.3803	4.00E-03	0.1222	1.30E-03	0.1233	1.20E-03
Twonorm	0.0216	2.83E-05	0.0225	3.19E-05	0.0219	2.78E-05	0.0219	2.72E-05
Magic	0.2733	5.06E-05	0.2468	5.08E-05	0.1921	8.34E-05	0.1923	7.93E-05
Ring	0.0209	2.20E-05	0.0207	2.29E-05	0.1131	1.16E-04	0.1131	9.98E-05
Tae	0.5595	1.39E-02	0.4460	1.33E-02	0.4365	1.36E-02	0.5132	1.67E-02
Contraceptive	0.5306	1.80E-03	0.5099	2.10E-03	0.4667	1.30E-03	0.4671	1.70E-03
Vehicle	0.5424	2.40E-03	0.5124	2.20E-03	0.2166	1.40E-03	0.2132	1.80E-03
Iris	0.0453	2.50E-03	0.0287	1.60E-03	0.0360	2.10E-03	0.0400	3.02E-03
Letter	0.3573	9.82E-05	0.1302	5.53E-05	0.0797	3.03E-05	0.0834	2.98E-05

Table 4. Classifying error of GMM-based approaches





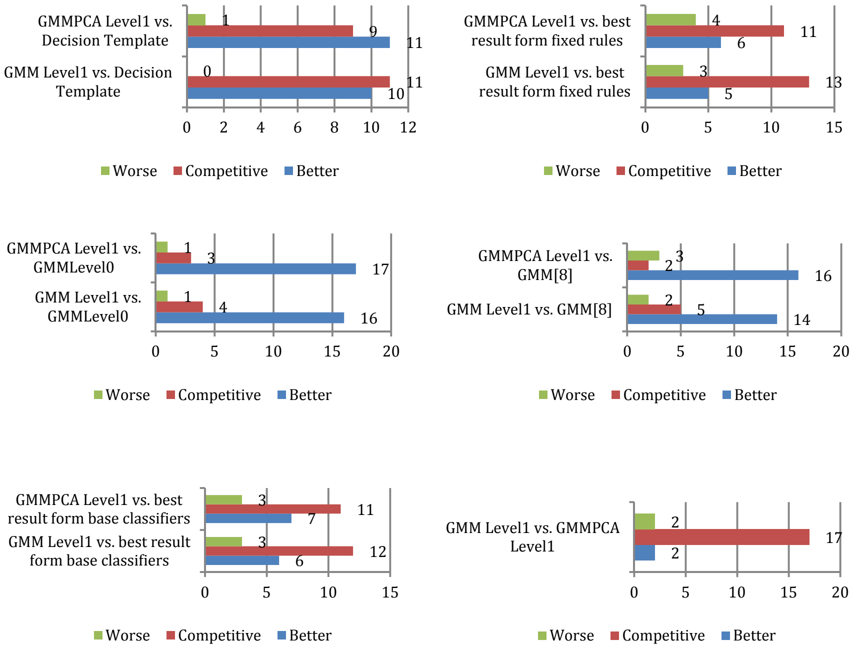


Fig. 2. Statistical test compare GMM, GMMPCA with the benchmarks

5 CONCLUSION AND FUTURE WORK

We have introduced a novel model which used GMM on Level1 data to combine results from base classifiers in multi classifier system. The experiments and assessments on 21 UCI files have illustrated the advantage of our method compared with popular state-of-art combining algorithms. Specifically, GMM-based approach on Level1 is better than on Level0 and Moghaddam approach [12] as well as Decision Template and SCANN, is competitive with best result from fixed rules and MLR. However, several potential limitations have also been addressed related to performance and effectiveness of our model. First, Level1 of training set is needed as input data for our framework (as well as Decision Template, MLR, SCANN) while fixed rules only employees Level1 of unlabeled observation; as a result, computational cost is gone up. Moreover, performance of GMM is time-consuming because of produce to find optimal number of components by BIC (eqn. 9). Due to this analysis, we are planning to improve performance of our model by studying new methods instead of using BIC. Besides, we intend applying classifier and feature selection methods to increase classifying accuracy of our model.

REFERENCES

[1] Robert P. W. Duin, The Combining Classifier: To Train or Not to Train? Proceedings. 16th International Conference on Pattern Recognition, Vol. 2, pp. 765-770, 2002.

[2] D.H. Wolpert, Stacked Generalization, Neural Networks, Vol. 5, No. 2, pp. 241-259, Pergamon Press, 1992.

[3] 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.

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

[5] 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.

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

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

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

[9] 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.

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

[11] 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.

[12] 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.

[13] 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.

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

[15] 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.

[16] Hayit Greenspan and Adi T.Pinhas, Medical Image Categorization and Retrieval for PACS Using the GMM-KL Framework, IEEE Transactions on Information Technology in Biomedicine, Vol. 11, No. 2, March 2007.

[17] G. J. McLachlan and D. Peel, Finite Mixture Models, New York Wiley, 2000

[18] Wei Li, Saurabh Prasad and James E.Fowler, Hyperspectral Image Clasfication Using Gaussian Mixture Models and Markov Random Fields, IEEE Geoscience and Remote Sensing Letter, Vol. 11, No. 1, January 2014.

[19] Zhiwu Lu and Horace H. S. I, Generalized Competitive Learning of Gaussian Mixture Models, IEEE Transactions on Systems, Man and Cybernetics – Part B: Cybernetics, Vol.39, No. 4, August 2009.

[20] 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.

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

[22] Alan Julian Izenman, Modern Multivariate Statistical Techniques: Regression, Classification and Manifold Learning, Springer Press, 2008.

[23] 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.