Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

A Novel Dynamic Fusion Method Using Localized Generalization Error Model

Daniel S. Yeung
School of Computer Science and Engineering,
South China University of Technology,
Guangzhou 510006, China
danyeung@ieee.org

Patrick P. K. Chan
Department of Computing,
The Hong Kong Polytechnic University,
Hung Hom, Kowloon, Hong Kong
patrickchan@ieee.org

***Abstract*—A new dynamic classifier fusion method named L-GEM Fusion Method (LFM) for Multiple Classifier Systems (MCSs) is proposed. The localized generalization error upper bound for the neighborhood of a testing sample is calculated and used to estimate the local competence of base classifiers in MCSs. Different from the recent dynamic classifier selection methods, the proposed method consider not only the training error but also the sensitivity of the base classifier. Experimental results show that the MCSs using LFM has more accurate than other popular dynamic fusion methods.**

***Keywords*—Multiple Classifier Systems (MCSs); Localized Generalization Error Model (L-GEM); Sensitivity; Dynamic Fusion Method**

I. INTRODUCTION

One of the most important steps in MCSs is the combination of base classifiers. The fusion methods have been under active research and many different approaches are studied [1-4]. The strategies in combining base classifiers in MCSs can be categorized into two types. In static fusion method, the combination method is decided during the training phase and it will not change for any testing samples. Majority Vote [5] and Weighted Average [6] are samples of static fusion. While in dynamic fusion method, e.g. Mixture of Experts [7] and Dynamic Integration [8], the combination method is changed according to the characteristics of testing samples and base classifiers in each classification phase.

A drawback of static fusion method is that it assumes base classifiers having the same performance in the whole input space. A more accurate base classifier is offered a larger value of the weight; vice versa. A base classifier may perform poorly in average but has a good performance in a certain region of the input space. The contribution in that region may be ignored since a small weight is assigned to that base classifier. On the contrast, in the dynamic fusion method, the weight is assigned to each base classifier according to their performance on the local region which the testing sample is located. Each base classifier can contribute to the MCS according to its local competence. Many studies showed that dynamic fusion methods outperform static fusion method [4, 8-10].

The weight assignment mechanism in dynamic fusion method is called the oracle. The oracle decides the value of weight for each base classifier when classifying a testing sample. The information considered by the oracle can be categorized into two types. The first type is the classification accuracy of base classifiers. Usually, the performance on the training samples or validation samples of testing sample is calculated [3, 8, 10]. Another type of information considered during weight assignment is the distance between testing sample from the training samples. It usually acts as a punishment to a base classifier if it classifies wrongly a sample near to the testing sample.

The current methods estimate the local competence of base classifiers on the testing sample using only the information provided by training samples. By this observation, the Localized Generalization Error Model (L-GEM) has been applied to dynamic fusion method. L-GEM developed by Yeung et al [11] estimated the error bound on the unseen samples located within a neighborhood of the training samples. This information may be useful to evaluate the performance of base classifiers to predict the testing sample.

A review of Dynamic Fusion Methods is presented in Section II. In Section III, he new Dynamic Fusion Method is presented. Experimental results are shown in Section IV. Section V concludes the paper.

II. DYNAMIC FUSION METHOD

Consider a population of L base classifiers trained by a given training set $D = \{(x_i, y_i)\}_{i=1}^{N}$, where N is the number of training samples. x_i is a n dimension vector denoted the i^{th} training sample, $[x_{i1}, x_{i2}, \ldots, x_{in}]^T$, n is the number of features and the superscript T is the vector transpose. y_i represents the true class ID of x_i and $y_i = \{\omega_c \mid c = 1..C\}$, where C is the number of classes. For each class, the MCSs using Dynamic Fusion Method is defined by:

$$f_c^{mcs}(x) = \sum_{l=1}^{L} w_c^l(x) f_c^l(x) \tag{1}$$

where x is a sample, $w_c^l(\,)$ is the weight assigned to the l^{th} base classifier calculated by the oracle and $f_c^l(\,)$ is the output of the l^{th} base classifier. When the output of base classifier is label output, $f_c^l(\,)$ is equal to 1 if the classifier predict x belongs to ω_c, otherwise it is equal to 0. While the output is the probability

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

of the base classifier decides that the sample belongs to ω_c, $f_c^l(\ )$ is a continuous value. The class ID estimated by MCS (y^{mcs}) of the sample x is defined by:

$$y^{mcs} = \arg\max_c f_c^{mcs}(x) \tag{2}$$

Figure 1 shows the architecture of MCSs using Dynamic Fusion Method. When a testing sample comes for classification, base classifiers make its own decision on it. On the other side, the oracle assigns a weight for each base classifier according to the testing sample. Finally, the final decision is calculated by using formula 1 and 2.

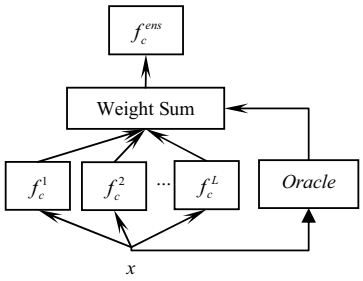

Figure 1. Architecture of Multiple Classifier Systems using Dynamic Fusion Method

In dynamic integration [13, 14], the base classifier's performance on the nearest K training samples of a testing sample is estimated using cross validation. The weight is the product of this local accuracy information and the distance between the corresponding training samples and the testing sample. The dynamic weight has been applied to Dynamic Selection (the best classifier is used), Dynamic Voting (weighted Voting) and Dynamic Voting with Selection (half of the best base classifier are combined by weighted Voting). This method has been applied to many different base classifier construction methods [12-14]. The experimental results showed that this dynamic weight fusion method outperforms the static method. The main drawback is that the cross validation is time consumed.

K-nearest-oracles (KNORA) dynamic selection method was proposed [3]. The validation set is used to estimate the error. For a testing sample, the weight of a base classifier is calculated according to its performance on the nearest K neighbors in the validation set. In KNORA-ELMINATE, only the outputs of the base classifiers which classify the nearest K validation samples correctly are used to decide the final decision. If no base classifier satisfies this requirement, K will be decrease till there is one classifier can perfectly classify the samples. Weight voting method is also applied and this method is called KNORA-UNION. A base classifier can classify nearest i validation samples has be larger weight than the one classify nearest j validation samples, where $1 \le j < i \le K$.

Dynamic Weight Update was proposed and applied to Learn++ in [15]. Every base classifier is trained by using different random training dataset. The weight of a classifier is determined by the Mahalanobis distance between a testing sample and the training dataset of that classifier. Classifiers trained with datasets closer to a testing sample are given larger weight value.

[16] proposed a dynamic fusion method which assigns a large weight to a base classifier with higher confidence about its output. The continuous-valued output can be interpreted as the probability of a sample in a class. When outputs of classifiers are near to 0 or 1, it means the classifiers have confidence about the decision.

$$w_c^l(x) = \begin{cases} f_c^l(x) & f_c^l(x) \ge 0.5 \\ 1 - f_c^l(x) & otherwise \end{cases} \tag{3}$$

An additional training process is required in some dynamic fusion methods. In Mixture of Experts [7], the weights of base classifiers are calculated by a neural network called the gating network. The gating network is trained using the training samples to minimize the following objective function:

$$E = \sum_{l=1}^{L}\sum_{i=1}^{N} w_c^l(x_i)\left(f_c^l(x_i) - F_c(x_i)\right)^2 \tag{5}$$

where $\sum_{l=1}^{L} w_c^l(x) = 1$ and $F_c(x)$ is the target output of sample x of class c. Obviously, this fusion method requested additional training are more time-consumed since additional training is required.

Classifier selection method [4, 17] is a special case of dynamic fusion method. Rather than assigning different weights to base classifiers, the oracle only select the best base classifier. The most common selection method estimates a prior and a posterior probability of base classifier classifying the testing sample correctly using on the K-nearest neighbors.

III. DYNAMIC FUSION METHOD USING L-GEM

The algorithm of the L-GEM Fusion Method (LFM) is introduced in this section. The idea of LFM is using the L-GEM as oracle to calculate the weights for base classifiers. When classifying a new sample, the oracle estimates the local generalization error bound of the local region where the sample located by using L-GEM. The weight to each base classifier is assigned according to the estimated local error bound. The outputs issued by the base classifiers are then combined using weighted averages.

1. Calculate distance between training sample and testing sample
2. Build nearest K neighborhoods set (X^K)
3. Calculate the q value
4. Calculate the weight for each base classifier using $R'_{Q^K}(f, X^K, q)$
5. Normalized the weight if necessary
6. Combine the decisions for each class
7. Make the final decision

Figure 2. The algorithm to classify a sample using LFM

L-GEM has been proposed by Yeung et al [11, 18]. As there is no information about unseen samples which are very different from the training set, a classifier cannot learn this part

of the input space and subsequently the error of the classifier in that region is expected to be high. Therefore, it may be counter-productive to assess the generalization performance of the classifier on unseen samples very different from the training set. Hence it will be more sensible to develop a generalization error model for unseen samples located within a neighborhood of the training samples. The Localized Generalization Error Bound (R^*_Q) is an upper bound of the mean square error (MSE) of unseen samples in a neighborhood of the training samples.

Figure 2 gives a description of the classification algorithm of LFM. Before classifying a testing sample (x_{test}), L base classifiers are trained by a particular base classifier construction method. To classify a sample, the distances (*dist*) between each training samples (x_i) and the testing sample are measured, where $i = 1 \ldots N$ and N is the number of training samples. The K training samples with the smallest distance are chosen and are formed K neighborhoods set (X^K). The largest value of the distance between the testing sample and samples in X^K is used as the value of q for L-GEM. Choosing the largest value of the distance ensures that every local neighborhood (Q_i^K) for each sample in X^K can cover the testing sample. The local region of the testing sample point (Q^K) is defined as the union of all local neighborhoods:

$$Q^K = \bigcup_{i=1}^{K} Q_i^K \quad \text{and} \quad Q_i^K = \left\{ x \mid x = x_i^K + \Delta x; |\Delta x| \le q \right\} \tag{6}$$

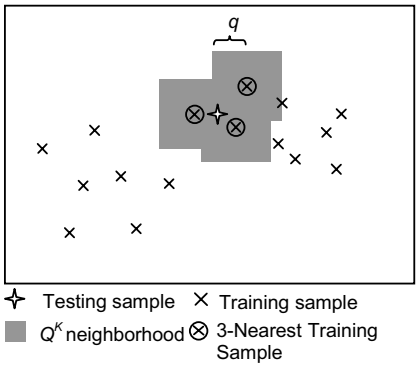

Figure 3. Q neighborhood for a testing sample with nearest 3 neighborhoods

Figure 3 illustrates an example with nearest 3 training samples of a testing sample in an artificial dataset. According to L-GEM, the local generalization error bound of Q^K region ($R^*_{Q^K}$) is:

$$R^*_{Q^K} = \left(\sqrt{\frac{1}{K}\sum_{i}^{K} err(f, x_i)} + \sqrt{\frac{1}{K}\sum_{i}^{K} \underset{Q_i^K}{E}\left((\Delta Y(f, x_i, q))^2 \right)} + \sqrt{A} \right)^2 + \varepsilon \tag{7}$$

where $err(f, x_i) = f(x_i) - F(x_i)$, $\Delta Y(f, x_i, q) = f(x_i + \Delta x) - f(x_i)$, $|\Delta x| \le q$ and $\varepsilon = B\sqrt{\ln \eta / (-2N)} \cdot E_Q\left((\Delta Y_b)^2 \right)$, η, A and B represent the sensitivity measure, the confidence level of the bound, the difference between the maximum and minimum value of the target outputs, and the maximum value of the MSE respectively. The value of A and B are fixed for a given dataset. Since A and ε will not affect the value of R^*_Q, these two parameters are ignored and the new function is now defined as:

$$R'_{Q^K}(f, X^K, q) = \sqrt{\frac{1}{K}\sum_{i}^{K} err(f, x_i)} + \sqrt{\frac{1}{K}\sum_{i}^{K} \underset{Q_i^K}{E}\left((\Delta Y(f, x_i, q))^2 \right)} \tag{8}$$

The sensitivity term is defined as follows. When $\|\Delta x\|$ is small, the classifier output can be approximated by

$$f(x + \Delta x) \approx f(x) + \left(\frac{\partial f}{\partial x} \right)^T \Delta x + \frac{1}{2} \Delta x^T H \Delta x \tag{9}$$

where H denotes the Hessian matrix with element $h_{ij} = \partial^2 f / (\partial x_i \partial x_j)$ and H is assumed to zero approximately for the surfaces with small curvature. As a result, the sensitivity term in Q_i is:

$$\begin{aligned} \underset{Q_i}{E}\left((\Delta Y(f, x_i, q))^2 \right) &= \underset{Q_i}{E}\left((f(x_i + \Delta x) - f(x_i))^2 \right) \\ &\approx \frac{q^2}{3} \left(\frac{\partial f}{\partial x_i} \right)^T \left(\frac{\partial f}{\partial x_i} \right) \end{aligned} \tag{10}$$

Δx is assumed to be a uniform distribution in Q_i. Hence, Δx is zero mean and uncorrelated. The mean of $\Delta x \Delta x^T$ in Q_i is equal to $\delta^2 I$, where δ^2 is equal to $q^2/3$. The sensitivity term can be applied to any classifier, e.g. MLP Neural Network and RBF Neural Network, which is differentiable by a sample x.

A base classifier with smaller local generalization error bound is more preferred. Thus, the weight for base classifier is the inverse of R'_{Q^K}. If necessary, the weight can be normalized from 0 to 1. Finally, the outputs of base classifiers are combined using weighted averaging method. The class which has the largest value is assigned to the testing sample.

Before classifying a testing sample, L base classifiers should be trained using the given training set. Each base classifier must be different from the others. Otherwise it is not necessary to combine those same base classifiers. Thus, the objective of MCSs construction method is to build a set of diverse base classifiers for MCSs. There are many different kinds of MCSs construction methods, e.g. Bagging and Boosting. The LFM can be applied to any of these methods. The only requirement of the LFM is the sensitivity term must be defined and calculated for a base classifier. In this paper, the sensitivity term is defined as a differentiation of the base classifier function. Hence, any differentiable base classifier, e.g. MLP Neural Network, RBF Network and SVM, can be applied.

RBF Network is applied to LFM in this paper. The definition of sensitivity of RBF Network is:

$$\frac{\partial f^{RBF}}{\partial x_i} = \left[\frac{df^{RBF}}{dx_{i_1}}, \frac{df^{RBF}}{dx_{i_2}}, \ldots, \frac{df^{RBF}}{dx_{i_n}} \right]^T \tag{11}$$

$$\frac{df^{RBF}}{dx_{i_t}} = -\sum_{m=1}^{M} \frac{\alpha_m \left(x_{i_t} - u_{m_t}\right)}{v_{m_t}^2} \exp\left(-\sum_{j=1}^{n} \left(\frac{x_{i_j} - u_{m_j}}{\sqrt{2} v_{m_j}} \right)^2 \right) \quad (12)$$

where M denotes the number of hidden neurons, α_m denotes weight of the m^{th} Gaussian output function, u_m denotes the peak position of the m^{th} center and u_{mj} is the j feature of u_m, v_m denotes the width of the m^{th} center and v_{mj} is the j feature of v_m and x_i denotes the sample i and x_{ij} is the j feature of x_i.

One of the reasons this proposed method may be better than recent dynamic fusion methods is that it considers not only the error but also the sensitivity of the base classifiers. For example, classifier A and classifier B have same accuracy on each of training samples. However, the output of classifier A is stable while classifier B is fluctuant. Intuitively, classifier A is expected to have better performance to recognize the unseen samples since its output is more stable. However, the current dynamic fusion methods assign the same value of weight to these two classifiers due to the same error in the training samples. On the other side, the proposed method measures the stability of the classifier output. The fluctuant classifier will be punished and a smaller weight will be assigned comparing with the stable one.

IV. EXPERIMENT

In this section, the performance of the LFM is evaluated experimentally. Ten datasets shown in Table I from UCI machine learning repository [19] and Intelligent Data Analysis Group [20] have been used. They cover a wide range of applications involving two-class and multi-class problems. Each dataset is equally divided into three parts: training (35%), validation (15%) and testing (50%) set. The experiment generates thirty independent runs for each dataset. Only samples in the training set are used during training. Some fusion methods require a validation set in classifying samples. The validation set will be used by some particular fusion methods only. The testing set is reserved to evaluate the performance of the trained MCSs. The inputs of all samples are normalized to [0, 1] to eliminate the effect of a large range of values.

TABLE I. TEN DATASETS

Dataset	Short Name	# Class	# Sample	# Feature
Breast Cancer Wisconsin	Canc	2	569	30
Glass Identification	Glass	2	214	10
Connectionist	Conn	4	208	60
Credit Approval	Cred	2	690	15
Dermatology	Derm	2	366	34
Spambase	Spam	7	4601	57
Thyroid	Thy	2	215	5
Tic-Tac-Toe Endgame	TTT	2	958	9
Waveform	Wave	2	5000	21
Wine	Wine	3	178	13

RBF Neural Network is used as the base classifiers. The number of neuron of RBF Neural Network is also selected randomly from two to fifty. The center and width of neuron is determined by K-mean [21] and the K-nearest-neighbor algorithm [22] respectively. The weight is calculated using Singular Value Decomposition (SVD) method [23]. To diversify a set of base classifiers in a MCS, Bagging method [24] is applied. Each base classifier is assigned a different training set which randomly selecting from the original training set with replacement.

LFM is compared with well known dynamic selection methods mentioned in Section II: Dynamic Selection (DS), Dynamic Voting (DV), Dynamic Voting with Selection (DVS), K-nearest-oracles Union (KU), K-nearest-oracles Eliminate (KE), Mahalanobis Distance method (MD), Confidence Measure Method (CM), Mixture of Experts (ME), a priori (Pri) and a posteriori (Post) method. The best value of K for LFM, DS, DV, DVS, KU, KE, Pri and Post methods are selected using cross-validation. MLP Neural Network is acted as gating network in ME methods.

The dynamic fusion methods are applied to combine the same set of base classifiers to form MCSs. The only difference is the weight of each base classifier. The performance of different sizes of MCSs (L = 10 and 30) are evaluated.

In Table II, the Win-Tie-Loss gives the number of datasets for which the MCS with LFM has been better/same/worse compared to the one with other fusion methods. Each column represents different number of base classifiers are contained in MCSs. The row represents different fusion methods. The last column is the average of all fusion methods.

TABLE II. LFM VS OTHER FUSION METHODS: WIN-TIE-LOSS COMPARISON OVER TWENTY DATASET

	L = 10	L = 30
DS	10-0-0	10-0-0
DV	6-1-3	9-0-1
DVS	10-0-0	10-0-0
KU	9-0-1	6-1-3
KE	9-0-1	6-1-3
CM	8-1-1	10-0-0
MD	10-0-0	10-0-0
ME	7-1-2	9-0-1
Pri	7-0-3	8-0-2
Post	7-0-3	8-0-2
Average	**8.3-0.3-1.4**	**8.6-0.2-1.2**

The table III and IV shows the MCS with LFM outperforms other methods in most of datasets and wins more than 8 datasets in average. Especially comparing with DS, DVS, CM, the MCS with LFM is more accurate in all datasets. The performance of the MCS with KU, KE, Pri and Post is the closest to the one with LFM. However, LFM still performs better in most datasets. From the average value of Win-Tie-Loss, there is no significant difference between the MCSs using 10 or 30 base classifiers. LFM is consistency better than other fusion methods in around 8.45 datasets and only worse in around 1.3 datasets.

The average percentage of classification accuracy of the testing sets of MCSs using different fusion methods over 30 independent runs are shown in Table III and IV. Each table shows the performance of different size MCSs. A column

represents a fusion method while a dataset is represented by a row. The first value and second value in a cell are the average classification testing accuracy and its variance respectively. The student's t-test is applied to examine the statistical significance of the improvement made by LFM. When the absolute t-value is larger than 2.00 in each experiment, a difference between two means is significant at the 95% probability level. The value is bold and underline in the cell if the performance of MCSs with LFM is better than the one with other methods significantly.

TABLE III. LFM VS OTHER FUSION METHODS: AVERAGE CLASSIFICATION ACCURACY AND VARIANCE OF TESTING SET OF MCSS WITH 10 BASE CLASSIFIERS OVER THIRTY INDEPENDENT RUNS

L=10	LFM	DS	DV	DVS	KU	KE	CM	MD	ME	Pri	Post
Canc	96.61 ±0.00	**96.02 ±0.00**	**96.09 ±0.00**	**95.90 ±0.00**	**95.84 ±0.00**	**95.86 ±0.00**	**96.06 ±0.00**	**96.06 ±0.00**	**96.06 ±0.00**	**96.11 ±0.00**	**95.89 ±0.00**
Glass	86.49 ±0.07	**85.05 ±0.09**	86.93 ±0.09	**81.56 ±0.09**	86.15 ±0.13	86.25 ±0.13	86.67 ±0.08	85.97 ±0.09	86.67 ±0.11	86.71 ±0.10	87.06 ±0.10
Conn	82.17 ±0.09	**79.41 ±0.07**	81.82 ±0.04	**75.82 ±0.12**	84.29 ±0.08	83.98 ±0.08	**80.67 ±0.05**	**79.96 ±0.06**	**81.02 ±0.06**	82.84 ±0.06	82.40 ±0.05
Cred	85.52 ±0.02	85.30 ±0.01	85.36 ±0.02	85.20 ±0.01	**81.95 ±0.01**	**82.21 ±0.01**	**85.25 ±0.02**	**85.41 ±0.01**	85.44 ±0.02	**84.98 ±0.01**	**84.85 ±0.01**
Derm	97.05 ±0.00	96.95 ±0.00	97.05 ±0.00	96.85 ±0.00	**96.60 ±0.01**	**96.67 ±0.01**	96.95 ±0.00	96.95 ±0.00	96.85 ±0.00	**96.70 ±0.00**	**96.75 ±0.00**
Spam	86.72 ±0.01	**86.23 ±0.00**	86.90 ±0.01	**85.23 ±0.00**	**86.19 ±0.01**	**86.10 ±0.01**	86.54 ±0.00	86.54 ±0.00	86.85 ±0.01	87.47 ±0.01	87.11 ±0.00
Thy	95.75 ±0.01	**94.65 ±0.03**	94.99 ±0.02	**94.14 ±0.05**	**95.07 ±0.01**	**94.99 ±0.01**	**94.65 ±0.03**	**94.65 ±0.02**	**94.90 ±0.02**	**94.77 ±0.02**	**95.03 ±0.02**
TTT	83.13 ±0.04	**77.43 ±0.04**	82.56 ±0.04	**71.17 ±0.02**	**80.13 ±0.02**	**79.90 ±0.02**	**78.57 ±0.04**	**78.44 ±0.05**	**80.72 ±0.06**	82.94 ±0.03	**80.87 ±0.04**
Wave	86.72 ±0.00	**86.48 ±0.00**	86.73 ±0.00	**86.01 ±0.00**	**81.02 ±0.00**	**80.96 ±0.00**	86.72 ±0.00	86.70 ±0.00	86.72 ±0.00	**86.43 ±0.00**	**86.41 ±0.00**
Wine	97.42 ±0.02	**96.66 ±0.03**	96.90 ±0.02	**96.18 ±0.03**	97.00 ±0.02	96.97 ±0.02	**96.69 ±0.02**	96.90 ±0.02	**96.80 ±0.02**	**96.60 ±0.02**	**96.81 ±0.02**

TABLE IV. LFM VS OTHER FUSION METHODS: AVERAGE CLASSIFICATION ACCURACY AND VARIANCE OF TESTING SET OF MCSS WITH 30 BASE CLASSIFIERS OVER THIRTY INDEPENDENT RUNS

L=30	LFM	DS	DV	DVS	KU	KE	CM	MD	ME	Pri	Post
Canc	96.38 ±0.00	**95.97 ±0.00**	**96.09 ±0.00**	**95.77 ±0.00**	**95.65 ±0.00**	**95.69 ±0.00**	**96.03 ±0.00**	**96.03 ±0.00**	**96.09 ±0.00**	**95.99 ±0.00**	**95.86 ±0.00**
Glass	86.84 ±0.09	**84.79 ±0.10**	86.06 ±0.10	**82.60 ±0.08**	86.58 ±0.16	86.77 ±0.17	85.63 ±0.09	**85.19 ±0.10**	86.15 ±0.07	86.28 ±0.15	86.71 ±0.13
Conn	81.20 ±0.09	**78.05 ±0.08**	80.49 ±0.09	**74.93 ±0.10**	85.61 ±0.05	85.31 ±0.05	**78.55 ±0.05**	**78.02 ±0.06**	**79.52 ±0.08**	82.67 ±0.05	81.96 ±0.07
Cred	85.65 ±0.01	85.24 ±0.02	85.20 ±0.02	85.44 ±0.02	**82.21 ±0.01**	**81.80 ±0.01**	**85.10 ±0.01**	**85.10 ±0.01**	85.20 ±0.02	**84.87 ±0.02**	**84.69 ±0.01**
Derm	97.10 ±0.00	**96.87 ±0.00**	**96.85 ±0.00**	**96.85 ±0.00**	**96.80 ±0.01**	**96.82 ±0.00**	**96.80 ±0.00**	**96.80 ±0.00**	**96.55 ±0.01**	**96.60 ±0.00**	**96.65 ±0.00**
Spam	86.66 ±0.01	**86.10 ±0.01**	86.83 ±0.01	**85.00 ±0.00**	86.94 ±0.01	86.71 ±0.01	86.47 ±0.01	86.46 ±0.01	86.77 ±0.01	87.76 ±0.01	87.21 ±0.01
Thy	95.33 ±0.01	**94.70 ±0.02**	94.90 ±0.02	**94.39 ±0.02**	95.50 ±0.01	95.39 ±0.01	**94.73 ±0.02**	**94.82 ±0.02**	95.07 ±0.02	**94.69 ±0.02**	94.86 ±0.02
TTT	84.68 ±0.04	**79.22 ±0.03**	84.67 ±0.03	**72.96 ±0.04**	**83.34 ±0.03**	**83.44 ±0.03**	**80.00 ±0.02**	**79.75 ±0.02**	**82.41 ±0.03**	84.59 ±0.04	**83.55 ±0.02**
Wave	86.87 ±0.00	**86.63 ±0.00**	86.84 ±0.00	**86.22 ±0.00**	**82.23 ±0.00**	**81.80 ±0.00**	86.83 ±0.00	86.82 ±0.00	86.79 ±0.00	**86.56 ±0.00**	**86.56 ±0.00**
Wine	97.52 ±0.02	97.07 ±0.02	97.31 ±0.02	**96.69 ±0.02**	97.52 ±0.01	97.52 ±0.01	97.21 ±0.02	97.31 ±0.02	**96.69 ±0.02**	97.01 ±0.02	97.01 ±0.02

Generally, the MCS using FLM has higher testing accuracy than other fusion methods in most cases with 95% confidence. In Tic-Tac-Toe Endgame dataset, FLM is 3.9% and 3.3% more accurate than other methods in average when L = 10 and 30 respectively. In Credit Approval and Thyriod, FLM also perform well and yield approximately 1% more than other method in average.

Similar to previous observation, the difference between MCS with 10 and 30 base classifiers are insignificant. The average accuracy of MCSs with different fusion methods is 88.8% and 89.0% when 10 and 30 base classifiers are used respectively. The performance of MCS with 30 base classifiers using LFM, DS, DVS, KU, KE, Pri and Post is better than the one using 10 base classifiers. However, the improvement is not significant.

Figure 4 and 5 show the average training and testing time of MCSs with 30 base classifiers with different dynamic fusion methods on twenty datasets over thirty independent runs. Each bar represents the time of MCS using a fusion method.

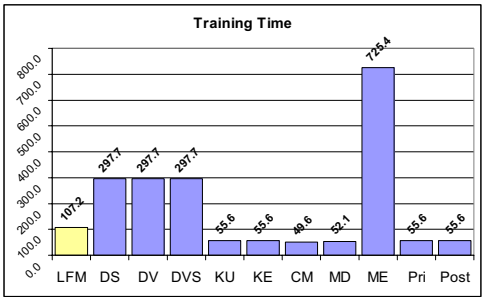

Figure 4. LFM VS Other Fusion Methods: Average Training Time of MCSs with 30 base classifiers on Ten Datasets over Thirty Independent Runs

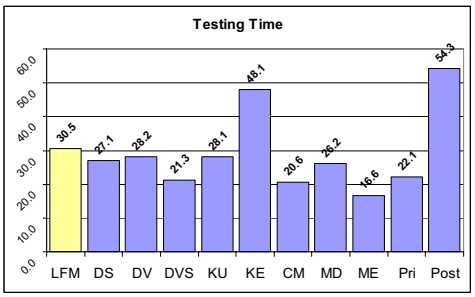

Figure 5. LFM VS Other Fusion Methods: Average Testing Time of MCSs with 30 base classifiers on Ten Datasets over Thirty Independent Runs

MD has the shortest training time among those fusion methods since it only needs to calculate the mean and variance of the training set. KU, KE, Pri and Post are slightly slower than MD. They just prepare the error of training set. The calculation of additional term (sensitivity term) causes LFM needs longer training time which is roughly double the time of methods using the training error only. ME, DS, DV and DVS have a longest training time because additional neural networks are needed to train. The differences of testing times among fusion methods are not as much as the training time. Post has the longest testing time because the computational time of posterior probability is high. Since only the base classifiers recognized the all K-nearest samples correctly are used in KE, sometimes all base classifiers cannot meet the requirement. The time is needed to re-select the classifier by reducing the value of K. ME has the lowest testing time since it only need to

calculate the Mahalanobis distance between the testing samples and dataset for each classifier. The testing time of the rest methods are similar. It is worth to note that although training and error and sensitivity are measured in LFM, not much testing time is required.

V. CONCLUSION

In this paper, a novel dynamic fusion method, L-GEM Fusion Method (LFM), is proposed. LFM estimates the local performance of base classifiers by using L-GEM. The current dynamic fusion methods estimate the local competence of base classifiers only based on the training error. The limitation of these approaches is the information of stability of classifier does not be considered. LFM assigned the weight to each base classifier according to the localized generalization error bound of the local area surrounded the testing sample. The information of training error and sensitivity of the classifier have been measured in the localized generalization error bound. LFM can be applied to any MCSs combined with continues-output base classifiers and does not limited to the MCS construction methods.

LFM has been compared with other ten dynamic fusion methods. The experimental results show that for a given a set of trained base classifiers, the testing accuracies of MCSs with LFM is higher than other fusion dynamic methods. One of the explanations may be the contribution of considering the stability of classifiers. The localized generalization error bound can estimate the local competence of base classifier more accurate than other methods. Although the estimation is more complexity in LFM, the testing time is similar to the other methods which consider the training error only. This is because most calculation can be completed in training phase. The information can be retrieved when classifying the testing samples.

ACKNOWLEDGEMENT

This work is supported by a 985 project, South China University of Technology.

REFERENCES

[1] Dietterich, T.G.: Machine Learning Research: Four Currect Directions, AI Magazine, Vol. 18 No 4 (1997) 97-136

[2] Merz, C.J.: Combining Classifiers Using Correspondence Analysis. In: Advances in Neural Information Processing Systems 10, M.I.Jordan, M.J.Kearns, S.A.Solla, eds., MIT Press, 1998

[3] Albert H. R. Ko, Robert Sabourin and Alceu Souza Britto, Jr. "From dynamic classifier selection to dynamic ensemble selection", Pattern Recognition, Volume 41, Issue 5 (May 2008), pp. 1735-1748

[4] K. Woods et al., "Combination of multiple classifiers using local accuracy estimates", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.19, No.4, April 1997, pp. 405-410

[5] R. Battiti and A. M. Colla Democracy in neural nets: Voting schemes for classification, Neural Networks, 7, 1994, 691 – 707

[6] Fumera, G.; Roli, F.; Serrau, A., A Theoretical Analysis of Bagging as a Linear Combination of Classifiers, Pattern Analysis and Machine Intelligence, IEEE Transactions on, Volume 30, Issue 7, July 2008 Page(s):1293 - 1299

[7] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, Geoffry E Hinton, Adaptive Mixture of Local Experts, Neural Computation, Vol. 3 (1991), pp. 79-87.

[8] Seppo Puuronen, Vagan Y. Terziyan, Alexey Tsymbal: A Dynamic Integration Algorithm for an Ensemble of Classifiers. Proceedings of the 11th International Symposium on Foundations of Intelligent Systems 1999: 592-600

[9] T.G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. Machine Learning, 40(2):139-157, 2000

[10] Alexey Tsymbal, Seppo Puuronen, Vagan Terziyan, A Technique for Advanced Dynamic Integration of Multiple Classifiers, Finnish Conference on Artificial Intelligence, 7-9 September 1998, pp. 71-79.

[11] D.S. Yeung, W.W.Y. Ng, D. Wang, E.C.C. Tsang, X.Z. Wang, Localized Generalization Error Model and Its Application to Architecture Selection for Radial Basis Function Neural Network, IEEE Transactions on Neural Networks 18 (5) (2007) 1294-1305.

[12] Alexey Tsymbal, Mykola Pechenizkiy, Padraig Cunningham: Dynamic Integration with Random Forests. 17th European Conference on Machine Learning 2006: 801-808

[13] Alexey Tsymbal, Mykola Pechenizkiy, Seppo Puuronen, David W. Patterson: Dynamic Integration of Classifiers in the Space of Principal Components. Advances in Databases and Information Systems 2003: 278-292

[14] Alexey Tsymbal, Seppo Puuronen: Dynamic Integration of Decision Committees. 7th International Conference on High Performance Computing 2000: 537-546

[15] Polikar, R. Krause, S. Burd, L. "Ensemble of classifiers based incremental learning with dynamic voting weight update", 2003. Proceedings of the International Joint Conference on Neural Networks, 20-24 July 2003, pp 2770- 2775 vol.4

[16] Ajith Abraham and Crina Grosan, Pharmaceutical Drug Design Using Dynamic Connectionist Ensemble Networks, Communications and Discoveries from Multidisciplinary Data, Studies in Computational Intelligence, Springer Verlag, Germany, Vol. 123, Iwata S. et al (Eds.), ISBN: 978-3-540-78732-7, pp. 221-231, 2008.

[17] Giacinto G. and Roli F.: Methods for Dynamic Classifier Selection. 10th International Conference on Image Analysis and Processing, Venice, Italy (1999), 659-664

[18] W.W.Y. Ng, D.S. Yeung, M. Firth, E.C.C. Tsang, X.Z. Wang, Feature selection using localized generalization error for supervised classification problems using RBFNN, Pattern Recognition 41 (12) (2008) 3706 - 3719

[19] http://archive.ics.uci.edu/ml/

[20] http://ida.first.fraunhofer.de/homepages/ida/

[21] L. Kiernan, J.D. Mason, K. Warwick, Robust initialisation of gaussian radial basis function networks using partitioned k-means clustering, Electronics Letters 32 (7) (1996) 671–673.

[22] M.T. Musavi, W. Ahmed, K.H. Chan, K.B. Faris, D.M. Hummels, On the training of radial basis function classifiers, Neural Network 5 (1992) 595–603.

[23] M.W. Mak, K.W. Cho, Genetic evolution of radial basis function centers for pattern classification, in: Proceedings of the IEEE International Joint Conference on Neural Networks, 1998, vol. 1, pp. 669 – 673.

[24] L. Breiman, Bagging predictor, Machine Learining 26 (26)123-140