Design of Dynamic Multiple Classifier Systems Based on Belief Functions

Deqiang Han
Center for Information Engineering Science Research
Xi'an Jiaotong University
Xi'an, Shaanxi, China 710049
Email: deqhan@gmail.com

X. Rong Li
Department of Electrical Engineering
University of New Orleans
New Orleans, LA 70148 USA
Email: xli@uno.edu

Shaoyi Liang
Inst. of Integrated Automation
Xi'an Jiaotong University
Xi'an, Shaanxi, China 710049
Email: liangshaoyi1987@gmail.com

***Abstract*—The technique of Multiple Classifier Systems (MCSs), which is a kind of decision-level information fusion, has fast become popular among researchers to fuse multiple classification outputs for better classification accuracy. In MCSs, there exist various kinds of uncertainties such as the ambiguity of the output of individual member classifier and the inconsistency among outputs of member classifiers. In this paper, we model the uncertainties in MCSs based on the theory of belief functions. The outputs of member classifiers are modeled using belief functions. A new measure of diversity in member classifiers is established using the distance of evidence, and the fusion rule adopted for MCSs is Demspter's rule of combination. The construction of MCSs based on the proposed diversity measure is a dynamic procedure and can achieve better performance than using existing diversity measures. Experimental results and related analyses show that our proposed measure and approach are rational and effective.**

***Keywords*—multiple classifier system; uncertainty; belief function; diversity; pattern classification.**

I. Introduction

Pattern classification [1] is one of the most important areas in machine learning. When dealing with classification problems in a complicated environment, a single classifier is often not competent. Multiple classifier systems (MCSs) [2], [3], [4], [5] aim at building multiple classifiers and then integrating their outputs for final decision-making. Over the past decade, MCSs have been actively exploited for improving classification accuracy and reliability over individual classifiers. MCSs have been widely used in areas such as handwriting character recognition [6], credit risk analysis [7], biometric identification [8], remote sensing [9], and automatic target recognition [10].

Implementation of a multiple classifier system (MCS), also called a classifiers ensemble, includes the generation of member classifiers and the fusion of the outputs of different member classifiers. There have emerged several approaches [2] to generating various member classifiers, e.g., using different samples, different feature spaces (or subspaces), different types of classifiers, and different parameter settings for classification. They all devote to generating various types of "differences" among member classifiers. Such "differences" are called "diversity" [11] in MCS, which are important for improving classification accuracy using an MCS. It would be meaningless to combine multiple redundant classifiers which have similar or the same misclassification regions. Diversity measures now have become a research focus in the field of MCSs. In 2005, the journal "Information Fusion" published a special issue on "Diversity Measure in Multiple Classifier Systems" [12], paying special attention to the definition of diversity measures and their prediction ability of the combining performance. It should be noted that almost all the existing diversity measures [11] are established based on classification results of the training samples, which are in a statistical sense. Information of a specific query (or test) sample is ignored.

The combination rules used in the MCS are also important. In 1992, Xu et al. [6] provided a detailed research paper on the selection of combination rules in the MCS. Kittler [3] summarized the combination rules used in the MCS, especially those under a Bayesian framework. Many researchers have proposed various combination methods, such as the Bayes method, Behavior Knowledge Space (BKS) [13], logistic regression [14], voting [15], and Dempster-Shafer evidence theory [16] (also called the theory of belief functions). In general, the rule used for MCSs depends on the output type of member classifiers.

The use of multiple classifiers brings more information, and therefore better classification performance can be expected. However, it also brings the problem of conflict or inconsistency because the outputs of member classifiers are often not accordant, especially for the member classifiers with outputs at the measurement level. The theory of belief functions [16] is a powerful tool for uncertainty modeling and reasoning. In this paper, we attempt to model the outputs of member classifiers using belief functions and use Dempster's rule of combination as the fusion rule. As referred above, available diversity measures have limitations. We propose a new diversity measure based on the distance of evidence [17]. Such a measure can make full use of the information of the query sample. In the generation of member classifiers, we adopt the strategy of "overproduction and selection" — many member classifiers are generated, and only those having high diversity and good performance on training sets are selected to construct the MCS. Experimental results and related analyses verify the rationality

This study was supported in part by Grant for State Key Program for Basic Research of China (973) (No. 2013CB329405), National Natural Science Foundation of China (No.61104214, No. 61203222), NASA/LEQSF(2013-15)-Phase3-06 through grant NNX13AD29A, ONR-DEPSCoR through grant N00014-09-1-1169, Foundation for Innovative Research Groups of the National Natural Science Foundation of China (No. 61221063), China Postdoctoral Science Foundation (No. 20100481337, No.201104670), and Fundamental Research Funds for the Central Universities.

and efficiency of our proposed new measure and approach.

II. Basics of Multiple Classifier Systems

MCSs [2], [3], [4], [5] have attracted much interest in the machine learning and pattern classification community thanks to their potential of increasing classification accuracy. As mentioned above, there are several ways to generate member classifiers, e.g., using different feature spaces (or subspaces) and using different types of classifiers. The procedure of implementing an MCS based on different feature spaces (or subspaces) is shown in Fig. 1.

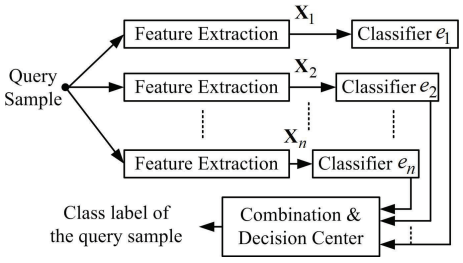

Fig. 1: An implementation of MCSs.

A. Output types of member classifiers

Given a query sample $\boldsymbol{x}_q \in R^d$, there are totally M classes in the class space represented by $\{C_i,\, i = 1, \ldots, M\}$. In Fig. 1, e_k $(k = 1, 2, ..., n)$ denote the member classifiers based on different feature spaces. $\mathbf{X}_k$ $(k = 1, 2, ..., n)$ denote the different feature vectors (corresponding to different feature spaces or subspaces) extracted from $\boldsymbol{x}_q$. Based on the outputs of member classifiers, a combination and decision center can assign a class label to $\boldsymbol{x}_q$. The outputs of a member classifier can be categorized into three types [6]:

1) Abstract Level: the classifier produces a unique class label for $\boldsymbol{x}_q$. Classifier e_k assigns a class label j_k to sample $\boldsymbol{x}_q$, i.e., $e_k(\boldsymbol{x}_q) = j_k$, $k = 1, 2, ..., n$.

2) Rank Level: the classifier ranks all possible labels in a set in a sequence $L_k \subseteq \Lambda$ with the label at top being the first choice.

3) Measurement Level: the classifier attributes to each label a measurement value such as *a posterior* probability or membership function value. For $\boldsymbol{x}_q$, each member classifier e_k brings out an output vector $[\omega_k(C_1), \omega_k(C_2), ..., \omega_k(C_M)]$, where $\omega_k(C_i) \in [0, 1]$ can be considered as the membership function for the given query sample belonging to class C_i.

B. Fusion rules for MCSs

The combination or fusion rules are crucial to the performance of an MCS. Various combination rules [3], [13], [15], [16] can be used in MCSs according to the member classifiers' output types [6]. If the outputs are at the abstract level, we can use the voting rules to fuse member classifiers; if the outputs are at the rank level or measurement level, especially the measurement level, we can use various rules including voting rules, Behavior Knowledge Space (KBS), fuzzy logic and the theory of evidence to fuse according to the outputs' specific representation (e.g., probability, membership function or belief function) at the measurement level. This is because the outputs at the measurement level have relatively rich information.

Combination rules are to combine or fuse different member classifiers. Comparatively, diversity among different member classifiers is a more crucial factor because great diversity is a necessary condition for improving classification performance. Diversity measures are discussed below.

C. Diversity measures for MCSs

Diversity measures quantify the diversity or complementarity among member classifiers. Available diversity measures can be categorized into two major types [11]:

1) Pairwise measures: Pairwise measures are calculated between two member classifiers. Table I shows the joint counts N_{ij}^{ab} of two classifiers e_i and e_j. For example N_{ij}^{01} denotes that e_i obtains an incorrect result and e_j obtains a correct result. Here, subscript ij for N has been omitted for convenience. Some representative pairwise diversity measures, including the Q-statistic (Q), correlation coefficient (R), disagreement measure (D) and double-fault measure (*DF*), are shown in (1)–(4).

TABLE I: The joint counts for outputs of two classifiers

	e_j *correct* (1)	e_j *incorrect* (0)
e_i *correct* (1)	N^{11}	N^{10}
e_i *incorrect* (0)	N^{01}	N^{00}

$$Q_{i,j} = \frac{N^{11}N^{00} - N^{01}N^{10}}{N^{11}N^{00} + N^{01}N^{10}} \tag{1}$$

$$R_{i,j} = \frac{N^{11}N^{00} - N^{01}N^{10}}{\sqrt{(N^{11} + N^{10})(N^{01} + N^{00})(N^{11} + N^{01})(N^{10} + N^{00})}} \tag{2}$$

$$D_{i,j} = \frac{N^{01} + N^{10}}{N^{11} + N^{00} + N^{01} + N^{10}} \tag{3}$$

$$DF_{i,j} = \frac{N^{00}}{N^{11} + N^{00} + N^{01} + N^{10}} \tag{4}$$

For an ensemble of L classifiers, the averaged diversity measure over all classifiers is

$$Diversity_{ave} = \frac{2}{L(L-1)} \sum_{i=1}^{L-1} \sum_{j=i+1}^{L} Diversity_{i,j} \tag{5}$$

where $\text{Diversity}_{i,j}$ can be either $Q_{i,j}$, $R_{i,j}$, $D_{i,j}$ or $DF_{i,j}$.

2) Non-pairwise measures: Non-pairwise measures are calculated directly over all member classifiers. They can be calculated using the proportion of classifiers that misclassify randomly selected samples. A non-pairwise measure is

$$E = \frac{1}{N} \sum_{j=1}^{N} \frac{1}{(L - \lceil L/2 \rceil)} \min\{l(z_j), L - l(z_j)\} \tag{6}$$

where L is the number of classifiers, N is the number of training samples, $\lceil \cdot \rceil$ is the ceiling function and $l(z_j)$ represents the number of classifiers that correctly classify the sample z_j. If for all the samples, all the classifiers are accordant, then E reaches its minimum value of 0. If for each sample z_j, $l(z_j)$ is close to $L - l(z_j)$, i.e., about half classifiers are not accordant to their counterparts, then E is close to its maximum value.

D. Limitations of available diversity measures

As we can see, the available diversity measures are designed using the classification results on training samples, i.e., the consistency or inconsistency of the classification results are used to establish the diversity measures. That is to say, such measures are defined in a statistical sense. Thus, they can not use information of specific query samples.

Furthermore, based on the available diversity measures, it is difficult to quantify "difference in misclassification regions," as illustrated in *Example* 1.

Example 1: Suppose that there are six samples $\boldsymbol{x}_1, ..., \boldsymbol{x}_6$ and two MCSs: MCS_1 and MCS_2. Both MCSs have three member classifiers. The classification results of each classifier are as follows (correct-1/incorrect-0, MCS_i^j denotes the jth classifier in the ith MCS),

$$\begin{array}{ll} MCS_1^1 : [1,0,0,0,0,0] & MCS_2^1 : [1,0,0,0,0,0] \\ MCS_1^2 : [0,1,0,0,0,0] & MCS_2^2 : [0,0,1,0,0,0] \\ MCS_1^3 : [0,0,1,0,0,0] & MCS_2^3 : [0,0,0,0,1,0] \end{array}$$

Quantify the diversity among the two MCSs using the measures Q-statistic (Q), correlation coefficient (R), disagreement measure (D), and double-fault measure (*DF*). When checking any two classifiers in MCS_1 or MCS_2, we obtain

$$N^{11} = 0, N^{10} = 1, N^{01} = 1, N^{00} = 4.$$

Thus, the diversity measures are

$$\begin{array}{l} Q_{MCS_1} = Q_{MCS_2} = -1, R_{MCS_1} = R_{MCS_2} = -0.2, \\ D_{MCS_1} = D_{MCS_2} = 0.33, DF_{MCS_1} = DF_{MCS_2} = 0.67. \end{array}$$

Such results show that the two MCSs have the same diversity according to any of the four measures. However, the member classifiers in the two MCSs have different correct/incorrect classified samples. The diversity in MCS_1 and that in MCS_2 are different in this sense. So, these traditional diversity measures cannot distinguish the two different "diversities". We call this "diversity submergence".

E. Uncertainty in implementation of MCSs

The use of multiple classifiers brings more information, and therefore better classification performance can be expected. However, uncertainty emerges at the same time. Here we are concerned with two types of uncertainty in MCSs.

- Type I: The uncertainty in the output of an individual member classifier.
  There is no uncertainty if a member classifier's output is at the abstract level, i.e., only with a determinate class label. Such an output is relatively arbitrary and might cause a loss of useful information. For the output at the measurement level, the possibility or probability of different class labels is assigned to a given query sample, i.e., information for decision is relatively abundant. However, there exists ambiguity for the class label assignment.
- Type II: The inconsistency among outputs of member classifiers.
  Each member classifier brings its own output. For an MCS, the ensemble of outputs obtained might be accordant or conflicting.

Since there exists uncertainty in implementation of MCSs, a tool of uncertainty modeling and reasoning is needed. The theory of belief function is a good choice, which is briefly recalled below.

III. Basics of Theory of Belief Functions

In the theory of belief functions [16], the elements in the frame of discernment (FOD) denoted by Θ are mutually exclusive and exhaustive. Suppose that 2^Θ denotes the powerset of FOD. Define the function $m : 2^\Theta \to [0, 1]$ as the basic belief assignment (bba) if it satisfies:

$$\sum\nolimits_{A\subseteq\Theta} m(A) = 1,\ m(\emptyset) = 0 \tag{7}$$

A bba is also called a mass function. If $m(A) > 0$, A is called a focal element of $m(\cdot)$.

Belief function (Bel) and plausibility function (Pl) are defined by:

$$Bel(A) = \sum\nolimits_{B\subseteq A} m(B) \tag{8}$$

$$Pl(A) = \sum\nolimits_{A\cap B\neq\emptyset} m(B) \tag{9}$$

Consider two bba's $m_1(\cdot)$ and $m_2(\cdot)$ defined over the FOD Θ. Their corresponding focal elements are $A_1, ..., A_k$ and $B_1, ..., B_l$. If $K = \sum_{A_i\cap B_j=\emptyset} m_1(A_i)m_2(B_j) < 1$, the function $m : 2^\Theta \to [0, 1]$ given by

$$m(A) = \begin{cases} 0,\ A = \emptyset \\ \frac{\sum_{A_i\cap B_j=A} m_1(A_i)m_2(B_j)}{1-\sum_{A_i\cap B_j=\emptyset} m_1(A_i)m_2(B_j)}, A \neq \emptyset \end{cases} \tag{10}$$

is also a bba. The rule defined by (10) is called Dempster's rule of combination for combining distinct bodies of evidence.

After combining the bba's by a given fusion rule we obtain a new bba. To make a decision on an element of the FOD Θ, we use a transformation to approximate the new bba as a probability mass function (pmf). The pignistic probability transformation $BetP(\cdot)$ proposed by Smets [18] is often used, which is illustrated in (11):

$$BetP_m(C_i) = \sum\nolimits_{\{C_i\}\in A\subseteq\Theta} m(A)/\left|A\right| \quad , \quad \forall A \subseteq \Theta \tag{11}$$

where $|A|$ denotes the cardinality of the focal element A.

Other transformations are also possible, such as $DSmP(\cdot)$ which is more complex to implement. Here, we use $BetP(\cdot)$ because of its simplicity. Details of $DSmP(\cdot)$ and other transformations are given in [19].

IV. Modeling Classifiers' Outputs Using Belief Functions

To use the theory of belief functions to deal with the uncertainty in MCSs, we should first model member classifiers' outputs using belief functions, which is in fact the bba generation. In our work, we use two approaches to generate the bba's.

1) Generation of Bayesian bba's: The Bayesian bba refers to a bba with only focal elements of singletons. Here, we use k-nearest neighbor (k-NN) classifier [1] to generate Bayesian bba's. L different feature subspaces of samples are used to generate L classifiers. That is, in each feature subspace i ($i = 1, ..., L$), a k-NN classifier e_i is implemented. For a given query sample $\boldsymbol{x}_q$, using e_i we find its k nearest neighbors: $\{\boldsymbol{x}^1, ..., \boldsymbol{x}^k\}$ and the class distributions of the k samples, i.e., to count the number of samples (denoted by n_C) belonging to each class. Suppose FOD is $\Theta = \{C_1, C_2, ..., C_M\}$. e_i's output in terms of Bayesian bba for the query sample $\boldsymbol{x}_q$ is

$$m_i(C_j) = \frac{n_j}{\sum_{l=1}^{M} n_l} \tag{12}$$

An illustrative example of bba generation is shown in Fig. 2 with $k = 13, n_1 = 7, n_2 = 3, n_3 = 3$. The corresponding Bayesian bba is

$$m(C_1) = 7/13, m(C_2) = 3/13, m(C_3) = 3/13.$$

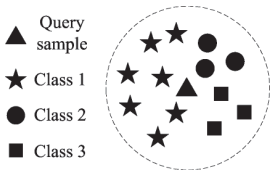

Fig. 2: Bayesian bba generation.

2) Generation of Non-Bayesian bba's: In [20], an evidential clustering approach was proposed. Although it was designed for unsupervised learning, e.g., clustering, we use it here as a reference to generate bba's in supervised learning, e.g., classification.

Suppose that there are M classes. Thus, the FOD is set as $\{C_1, C_2, ..., C_M\}$. There could be $2^M - 1$ focal elements. Suppose there are L feature subspaces. Then we can generate L member classifiers. For a member classifier e_i, we calculate the centroid $\boldsymbol{r}_j = \sum_{i=1}^{n_j} \boldsymbol{x}_i^j / n_j$ over all the training samples belonging to class C_j to represent class C_j. We can also define a compound class. For example, for class C_s and C_t, where $s, t \in \{1, ..., M\}$, we calculate the centroid $\boldsymbol{r}_{s\cup t} = (\boldsymbol{r}_s + \boldsymbol{r}_t)/2$ of all the training samples belonging to C_s or C_t to represent the compound class of $C_s \cup C_t$. Then calculate the distance $d(\boldsymbol{x}_q, \boldsymbol{r}_l)$ between $\boldsymbol{x}_q$ and each centroid $\boldsymbol{r}_l$, respectively. Here l is the index of all the "extended" classes, which include single classes and compound classes. For example, if there are three single classes, then there will be $2^3 - 1 = 7$ extended classes. Thus, $l = 1, ..., 7$. The output of e_i in terms of bba for query sample $\boldsymbol{x}_q$ is generated as

$$m_i^{\boldsymbol{x}_q}(A_j) = \frac{|A_j|^{-\alpha/(\beta-1)} \cdot (d(\boldsymbol{x}_q, \boldsymbol{r}_j))^{-2/(\beta-1)}}{\sum_{A_l \neq \emptyset} |A_l|^{-\alpha/(\beta-1)} \cdot d(\boldsymbol{x}_q, \boldsymbol{r}_l)^{-2/(\beta-1)} + \delta^{-2/(\beta-1)}} \tag{13}$$

where A_j is a focal element, and α and β are parameters of weighting component. See [20] for details. Their default values are $\alpha = 2, \beta = 2$. In our work, we are concerned with only the close-world assumption, i.e., no unknown class, and therefore the parameter δ (distance to empty set) is set to 0.

The non-Bayesian bba generation is also illustrated in Fig. 3 (FOD = $\{C_1, C_2, C_3\}$). For example, for classifier e_i, when $d(\boldsymbol{x}_q, \boldsymbol{r}_1) = 1.8$, $d(\boldsymbol{x}_q, \boldsymbol{r}_2) = 3.5$, $d(\boldsymbol{x}_q, \boldsymbol{r}_3) = 1.7$, $d(\boldsymbol{x}_q, \boldsymbol{r}_4) = 1.7$, $d(\boldsymbol{x}_q, \boldsymbol{r}_5) = 0.5$, $d(\boldsymbol{x}_q, \boldsymbol{r}_6) = 2.0$ and $d(\boldsymbol{x}_q, \boldsymbol{r}_7) = 1.5$, according to (13), e_i's non-Bayesian bba for $\boldsymbol{x}_q$ is

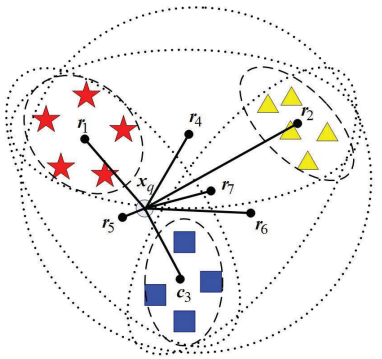

Fig. 3: Non-Bayesian bba generation.

$$m_i^{x_q}(C_1) = 0.1595, m_i^{x_q}(C_2) = 0.0422, m_i^{x_q}(C_3) = 0.1789,$$
$$m_i^{x_q}(C_1 \cup C_2) = 0.0447, m_i^{x_q}(C_1 \cup C_3) = 0.5169,$$
$$m_i^{x_q}(C_1 \cup C_3) = 0.0323, m_i^{x_q}(C_1 \cup C_2 \cup C_3) = 0.0255.$$

V. New Diversity Measure Based on Theory of Belief Functions

Diversity is a crucial factor in designing MCSs. As mentioned above, existing diversity measures have limitations. Since we have modeled the outputs of member classifiers using belief functions, we can use some indices measuring the difference or diversity among belief functions to describe the diversity among member classifiers. Distance of evidence is such a good choice.

A. Distance of evidence

Several distances of evidence have been proposed in the literature [21]. Among all the proposed distances of evidence, we have chosen Jousselme's distance based on the form of Euclidean metric because it is a strict [22] distance metric and takes into account the specificity of focal elements of the bba. Jousselme's distance is defined as

$$d_J(m_1, m_2) = \sqrt{(m_1 - m_2)^T \mathbf{Jac} \ (m_1 - m_2)} \tag{14}$$

Jac is Jaccard's weight matrix whose elements are given by

$$\mathbf{Jac}(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{15}$$

where A and B represent the focal elements of $m_1(\cdot)$ and $m_2(\cdot)$, respectively.

B. Diversity measure based on distance of evidence

Then, we define a diversity measure using the distance of evidence as follows. Given an MCS with L member classifiers. Suppose that there are M possible classes. Given a query sample $\boldsymbol{x}_q$, the classifier e_i's output is modeled using a bba $m_i^{\boldsymbol{x}_q}(\cdot)$, where $i = 1, ..., L$.

Step 1: Calculate the center of all bba's.

The center of all bba's is calculated according to

$$m_c^{\boldsymbol{x}_q}(A_j) = \frac{\sum_{i=1}^{L} m_i^{\boldsymbol{x}_q}(A_j)}{L} \tag{16}$$

where $m_c^{\boldsymbol{x}_q}(\cdot)$ is the center or mean bba and A_j is a focal element ($j = 1, ..., 2^M - 1$).

Step 2: Calculate the average distance between all bba's and the center bba.

For an ensemble of classifiers, if their output bba's are similar to each other, their diversity will be small. Thus, we define the diversity as the average distance between all bba's and the center bba.

$$Div_{bba}^{\boldsymbol{x}_q}(MSC) = \frac{\sum_{i=1}^{L} d_J(m_c^{\boldsymbol{x}_q}(\cdot), m_i^{\boldsymbol{x}_q}(\cdot))}{L} \tag{17}$$

It can be seen that our proposed diversity measure is query-sample-dependent. That is, it is a dynamic index dependent on query samples, while traditional diversity measures are established in a statistical sense, as referred above. Our proposed diversity measure can alleviate some limitations of available diversity measures. See Example 2 below.

Example 2: Suppose the FOD is $\{C_1, C_2, C_3\}$. Given a sample $\boldsymbol{x}$ belonging to C_1. By using the member classifiers in two MCSs (MCS_1, MCS_2), the classification results are shown in Table II. It should be noted that each bba in Table II is a vector of $[m(C_1), m(C_2), m(C_3)]$.

TABLE II: Classification results for a query sample

Classifiers	Output bba	Classification result	correct(1) /incorrect(0)
$MCS_1 - e_1$	[0.8, 0.1, 0.1]	C_1	1
$MCS_1 - e_2$	[0.1, 0.8, 0.1]	C_2	0
$MCS_2 - e_1$	[0.5, 0.4, 0.1]	C_1	1
$MCS_2 - e_2$	[0.4, 0.5, 0.1]	C_2	0

For the sample $\boldsymbol{x}$, the member classifiers in MCS_1 disagree with each other and so does MCS_2. If we want to calculate the traditional diversity of MCS_1 and MCS_2, using of the sample $\boldsymbol{x}$ only increases the number of disagreed classification (N^{10}) by one for both MCSs considered. This is because for both MCSs, $e_1(\boldsymbol{x})$ is correct and $e_2(\boldsymbol{x})$ is incorrect. However, based on the outputs in terms of bba's, we can obviously find that the inconsistency in MCS_1 is more significant than in MCS_2. With the traditional diversity measures, such a difference cannot be revealed.

By using our proposed diversity measure (17), we obtain $Div_{bba}^{\boldsymbol{x}_q}(MCS_1) = 0.7$ and $Div_{bba}^{\boldsymbol{x}_q}(MCS_2) = 0.1$. Such results can well describe the real situations for the two MCSs.

Diversity measures can be used to evaluate the diversity among member classifiers. However, the more important issue is to use the diversity measure to design MCSs, i.e., generate member classifiers. In this paper, we propose a dynamic MCS design using the proposed diversity above.

VI. Dynamic Design of MCSs Based on Theory of Belief Functions

For MCSs, more classifiers do not always bring better classification accuracy. In our work, we adopt the strategy of "overproduction-selection" to implement MCSs. That is, we over produce individual classifiers at first and then select some of them according to a criterion. The whole procedure is illustrated in Fig. 4.

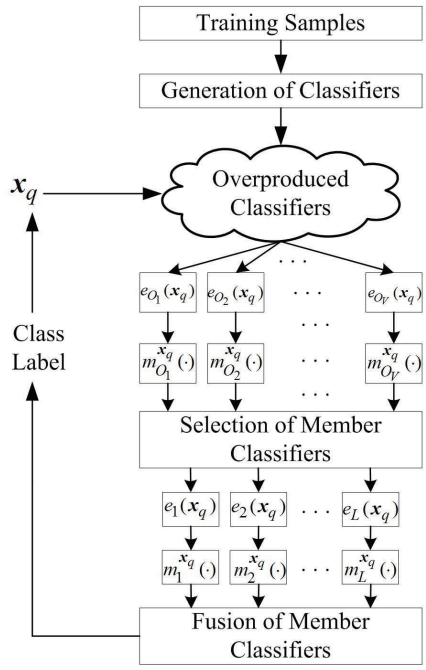

Fig. 4: Procedure of MCSs implementation.

In Fig. 4, $e_{O_i}(\cdot)(i = 1, ..., V)$ represent the overproduced individual classifiers and $e_j(\cdot)$ $(j = 1, ..., L)$ represent the member classifiers selected out of the overproduced ensemble.

A. Overproduction

In our work, the member classifiers are generated using different feature subspaces of the given training samples. Other methods such as using different samples and using different types of classifiers can also be used.

B. Selection

Now we discuss how to select member classifiers. Note that although diversity is crucial, it is only necessary for MCSs' improvement of classification performance. That is, only diversity can not assure better performance. If the classifiers with a large diversity and simultaneously having high classification accuracy on training samples are used to construct MCSs, better classification accuracy can be expected. Therefore, we convert the implementation of MCSs into an optimization problem whose objective function is based on the joint use of the newly proposed diversity measure and the average classification accuracy:

$$fitness^{\boldsymbol{x}_q}(MCS) = w_D \cdot Div_{bba}^{\boldsymbol{x}_q}(MCS) + w_A \cdot Acc_{ave}(MCS) \tag{18}$$

where $Acc_{ave}(MCS)$ is the average classification accuracy of the given MCS. w_D, w_A represent the weights of diversity and accuracy, respectively. It should be noted that the ranges of the distance-based diversity and of the accuracy are both $[0, 1]$. The weighting parameters selection is necessary. It depends on the

users' preference. We suggest to use $w_D = w_A = 1$, which indicates an equal-treat attitude.

We use the Genetic Algorithm (GA) [23] to find the best MCS, where the fitness function is (18), which is maximized as

$$MCS^{\boldsymbol{x}_q}_{selected} = \arg\max_{MCS}\{fitness^{\boldsymbol{x}_q}(MCS)\} \quad (19)$$

that is, to find an MCS with high diversity and simultaneously with good accuracy on training samples. We can see that for different query samples, different member classifiers are generated and selected. Therefore, in our work, the design of MCSs is query-sample-dependent. Suppose that V classifiers $e_{O_i}(\cdot)(i = 1, ..., V)$ are overproduced. We define a V-dimensional vector $\boldsymbol{S} = [s(1), ..., s(V)]$ to represent the MCS, i.e., the unknown variable of the optimization problem. The value of $s(i)$ can be 0 or 1. $s(i) = 1$ indicates that the classifier $e_{O_i}(\cdot)$ is selected and $s(i) = 0$ otherwise. For example if $\boldsymbol{S} = [1, 0, 0, 1]$, it means that four classifers are overproduced and two classifiers $e_{O_1}(\cdot)$ and $e_{O_4}(\cdot)$ are selected for the MCS.

Here we give a simple illustrative example for our dynamic design of MCSs.

Example 3: Suppose that FOD =$\{C_1, C_2, C_3\}$. That is, there are three classes. Suppose that three classifiers (e_1, e_2, e_3) are generated along with their corresponding outputs for query sample x_q and their classification accuracies on training samples are illustrated in Table III. (Here, each bba in Table III is a vector of $[m(C_1), m(C_2), m(C_3)]$.) Set $w_A = w_D = 1$. Then

TABLE III: Output bba's and Classification accuracies

Classifiers	Output	Accu.
e_1	[0.40, 0.50, 0.10]	90%
e_2	[0.60, 0.30, 0.10]	80%
e_3	[0.35, 0.30, 0.35]	85%

we calculate the fitness function values for different groups of classifiers as

$$\begin{aligned} &fitness^{\boldsymbol{x}_q}(\{e\_1, e\_2, e\_3\}) = 0.9807;\\ &fitness^{\boldsymbol{x}_q}(\{e\_1, e\_2\}) = 1.0500;\\ &fitness^{\boldsymbol{x}_q}(\{e\_1, e\_3\}) = 1.1041;\\ &fitness^{\boldsymbol{x}_q}(\{e\_2, e\_3\}) = 1.0750. \end{aligned}$$

According to the maximization criterion, the member classifiers selected for MCS are e_1 and e_3. Then by using Dempster's rule of combination, we can obtain the combined bba $m_{comb}(\cdot) = m_1(\cdot) \oplus m_3(\cdot)$. Since in this example, m_{comb} is a Bayesian bba, $BetP(C_i) = m_{comb}(C_i)$. We can obtain

$$BetP(C_1) = 0.4308, BetP(C_2) = 0.4615, BetP(C_3) = 0.1077$$

As we can see, $BetP(C_2)$ has the maximum value. Thus, the query sample $\boldsymbol{x}_q$ is labeled as class C_2.

C. Simplifications of the selection procedure

Since the vector to represent an MCS is V-dimensional, there are $2^V - 1$ possible MCSs for selection. If the number of overproduced classifiers is large, the search space becomes large, which is harmful for rapid finding of the best MCS. When $V = 20$, the number of solutions is $C^1_{20} + C^2_{20} + C^3_{20} + ... + C^{20}_{20} = 1048575$, where C^n_V is the combination number for selecting n classifiers out of V. If we can know in advance the number of member classifiers (L) in the MCS, the size of the search space significantly decreases from $C^1_{20} + C^2_{20} + ... + C^{20}_{20}$ to only C^L_{20}. Then, how do we know a priori the number of member classifiers (L)?

1) Simplification I: We attempt to estimate L using clustering analysis to estimate the number of selected member classifiers. When V classifiers are produced, there are V bba's. We can use clustering analysis on the obtained bba's, i.e., we treat a bba as a piece of "sample data" in the "classifier space". By using the clustering method without a preset cluster number such as ISODATA [1] and by the criterion based on the distance of evidence, the bba's (representing classifiers) can be automatically grouped. Actually, clustering bba's is clustering classifiers overproduced. After the value of L is obtained, the size of the search space can be reduced to C^L_V. For example, when we set $V = 20$, if the obtained L is 6, the size of the search space is reduced to $C^6_{20} = 38760$, which is only $38760/\sum_{i=1,...,20}(C^i_{20}) = 3.7\%$ of the size of the original search space.

2) Simplification II: We also estimate the value of L using clustering analysis. However, we use a simple rank instead of optimization such as GA. We select the classifier with the highest average classification accuracy on training samples from each bba's cluster (i.e., the classifier clusters) obtained. For example, set V = 20. If 20 bba's are clustered into four clusters with 5, 6, 4, and 5 bba's inside, respectively, what we should do is to select one member classifier out of 5, 1 out of 6, 1 out of 4 and 1 out of 5 according to the member classifier's average classification accuracy on training samples. Here four different bba's are selected from four different clusters. Thus, the diversity among them can be assured to some extent. Furthermore, the selection in each cluster is based on the classification accuracy. Thus, both the diversity and performance are indicated. The number of solutions will be significantly reduced to $5 + 6 + 4 + 5 = 20$. Our GA-based approach in (19) can be seen as a joint optimization problem (the objective function is a weighted sum of two goals), while simplification II can be considered as achieving the two goal one by one. Thus, simplification II might lose performance, although it significantly reduce the computational costs.

It should be noted that the performance of both simplification I and simplification II depend on the clustering results using ISODATA, which is affected by ISODATA's parameter selection.

D. Combination of member classifiers

For the L member classifiers selected for an MCS, we combine their corresponding L bba's using Dempster's rule (10) to obtain the combined bba $m^{\boldsymbol{x}_q}_c(\cdot) = m^{\boldsymbol{x}_q}_1(\cdot) \oplus m^{\boldsymbol{x}_q}_2(\cdot) \oplus ... \oplus m^{\boldsymbol{x}_q}_L(\cdot)$. Then by using the pignistic probability transformation (11), we can obtain the probability for different classes. Based on it the final classification decision for $\boldsymbol{x}_q$ can be made. Other combination rules [19] and probability transformations [19] can also be used here.

VII. Experiments

In the experiments, we use eight datasets from the UCI [24] for pattern classification, as listed in Table IV. Our experimental settings are as follows.

TABLE IV: UCI datasets used in the experiments

Dataset	Number of classes	Feature dimension	Number of samples
Pima	2	8	768
Wine	3	13	178
Ionosphere	2	34	351
Haberman	2	3	306
Glass	6	9	214
Iris	3	4	150
Bupa	2	6	345
WDBC	2	30	569

TABLE V: Comparison of different approaches in classification accuracy

Datasets	Fusion rule	New_NB[a]	New_B[b]	DF	Q	R	D
Pima	Majority voting	72.14 %	69.27 %	71.02 %	70.03 %	71.30 %	69.71 %
	Dempster's rule	74.45 %	72.97 %	73.83 %	72.71 %	73.33 %	73.65 %
Wine	Majority voting	97.14 %	95.96 %	95.33 %	96.24 %	96.30 %	95.61 %
	Dempster's rule	97.18 %	96.12 %	96.95 %	96.46 %	96.73 %	95.97 %
Ionosphere	Majority voting	83.64 %	83.59 %	83.23 %	83.43 %	83.07 %	83.30 %
	Dempster's rule	84.31 %	83.64 %	83.44 %	83.85 %	83.11 %	83.72 %
haberman	Majority voting	74.18 %	74.25 %	73.66 %	73.79 %	74.05 %	73.86 %
	Dempster's rule	74.18 %	74.25 %	73.73 %	73.79 %	74.05 %	73.86 %
glass	Majority Voting	64.26 %	65.50 %	64.59 %	65.39 %	60.81 %	64.51 %
	Dempster's rule	64.77 %	66.37 %	65.88 %	65.17 %	64.42 %	64.67 %
iris	Majority voting	95.33 %	95.56 %	96.44 %	92.22 %	95.78 %	95.33 %
	Dempster's rule	96.67 %	96.22 %	96.67 %	96.67 %	96.67 %	96.67 %
bupa	Majority voting	64.64 %	63.56 %	61.06 %	62.03 %	61.55 %	61.93 %
	Dempster's rule	71.88 %	68.12 %	71.21 %	70.92 %	69.86 %	71.88 %
WDBC	Majority voting	94.63 %	90.53 %	94.34 %	91.56 %	94.34 %	94.83 %
	Dempster's rule	96.39 %	95.36 %	96.49 %	95.90 %	96.19 %	96.39 %

[a] using the new diversity measure and the non-Bayesian bba's [b] using the new diversity measure and Bayesian bba's

1) Generation of training and testing samples: Each dataset's classification procedure is executed for 30 times. At each time, samples for training and testing are re-selected randomly. (In each experiment, each dataset is randomly grouped into two approximately equal parts, one for training and the other for testing.)

2) Overproduction of classifiers: We use the feature subspaces to overproduce classifiers. Each generated classifier corresponds to a 2-dimensional feature subspace. Suppose that the original feature space is F-dimensional. Then, C_F^2 classifiers are overproduced. For example, suppose $F = 8$. Then, $C_8^2 = 28$ classifiers are overproduced, out of which member classifiers are selected to construct the MCS. It should be noted that both Bayesian and non-Bayesian bba's are used in the experiments. In generating Bayesian bba's, the parameter k in k-NN is empirically set to 1/10 of the number of training samples,

3) Different diversity measures: We use some existing diversity measures and our proposed measures to design the MCSs. In designing MCSs, we use the same form of the objective function in (18) using different diversity measures including Q-statistic (Q), correlation coefficient (R), disagreement (D), and double fault (DF). In our MCS design based on the existing diversity measures, the classifier types used is k-NN (with k empirically set to 1/10 of the number of training samples).

4) Selection of classifiers: Among the original GA and the two simplifications, simplification I provides a tradeoff between performance and cost, and so it is used here.

5) Fusion rules: In this work, we have compared two fusion rules including the majority voting and Dempster's rule of combination. If we want to use the majority voting to fuse the member classifiers, each classifier should have its classification decision. Since the output of classifiers are bba's, we should transform each output bba's into the pignistic probability according to (11) and then make the classification decisions. (The class with the maximum pignistic probability value, which is no less than 0.5, is the decision result.)

The average classification accuracy of different approaches are shown in Table V. New_NB and New_B denote the approach based on the new diversity measure with the non-Bayesian bba's and Bayesian bba's, respectively.

In Table V, for each dataset and each fusion rule, with respect to classification accuracy, the first place is in red and the 2nd place is in blue. As we can see, for many datasets, our newly proposed approaches take the 1st or 2nd place of all the approaches compared. Using non-Bayesian bba's more often achieves better performance than using Bayesian bba's. This is caused by the rich information remained in non-Bayesian bba's. We can also see that using Dempster's rule is better than using the majority voting. This is because the majority voting uses only the information of class labels, while using bba's for fusion uses more useful information.

We compare the original GA-based approach, simplification I, and simplification II on the Iris dataset. The results are in Table VI.

TABLE VI: Comparison of original and simplification methods on iris dataset

Approaches	Average accuarcy	Time cost per sample (sec)
Original GA	96.67 %	1.4943
Simplification I	96.47 %	0.9194
Simplification II	92.67 %	0.0042

As we can see, simplification I provides a good trade off between the performance and the cost.

VIII. Conclusion

In this paper, a dynamic belief-functions based approach to designing multiple classifier systems is proposed, and a diversity measure is proposed using the distance of evidence. Our proposed MCS approach and diversity measure are query-sample dependent, which can better use of the information of each query sample. The designing of MCS is converted to an optimization problem in our work. Experimental results verify that our proposed measure, approach, and related simplifications are rational and effective compared with existing ones.

The theory of belief functions is good for uncertainty modeling and uncertainty reasoning. However, one of its main limitations is its computational complexity when the cardinality of FOD is large. It should be noted that since our proposed measure and approach are based on the theory of belief functions, when the number of possible classes gets large, the computational complexity also increases. For Bayesian bba's, the problem is not so significant. For non-Bayesian bba's, however, computational costs do increase significantly. Therefore, our future work will attempt to design some new fast algorithms for Dempster's rule of combination and propose bba approximations to reduce computational costs. Dempster's rule produces results that maybe judged as unsatisfactory or counter-intuitive [25] when the bodies of evidence to be combined are highly conflicting. Therefore, in our future work, other evidence combination rules [19] will also be used and compared.

Furthermore, in our proposed simplifications, the results depend on the clustering results, which is affected by parameter selection in clustering. More stable and robust clustering algorithms are required for improving our approach, which are important research topics in our future work.

References

[1] O. R. Duda, E.P. Hart, and D.G. Stork, *Pattern Classification*, 2nd ed. New York: Wiley Inter-Science Publication, 2001.

[2] R. Ranawana and V. Palade, "Multi-Classifier Systems: Review and a roadmap for developers", International Journal of Hybrid Intelligent Systems, vol. 3, no. 1, p. 35–61, 2006.

[3] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 3, p. 226–239, 1998.

[4] L. I. Kuncheva, "Switching between selection and fusion in combining classifiers: An experiment", IEEE Transaction on Systems, Man and Cybernetics - Part B, vol. 32, no. 2, p. 146–156, 2002.

[5] E. Bahri, N. Harbi, and H. N. HuuA, "Multiple classifier system using an adaptive strategy for intrusion detection", International Conference on Intelligent Computational Systems (ICICS'2012) Jan. 7-8, 2012 Dubai, p. 124–128.

[6] L. Xu, A. Krzyzak, and C.Y. Suen. "Methods of combining multiple classifiers and their applications to handwriting recognition", IEEE Transactions on Systems, Man, and Cybernetics, vol. 22, no. 3, p. 418–435, 1992.

[7] B. Twala, "Multiple classifier application to credit risk assessment", Expert Systems with Applications, vol. 37, no. 4, p. 3326–3336, 2010.

[8] S. Soviany, C. Soviany, and M. Jurian, "A multimodal approach for biometric authentication with multiple classifiers", in Proc. of International Conference on Communication, Information and Network Securiy, November 28-30, 2011, Venice, Italy, p. 2235–2240.

[9] P. Du, J. Xia, W. Zhang, et al., "Multiple classifier system for remote sensing image classification: a review", Sensors, vol. 12, no. 4, p. 4764–4792, 2012.

[10] W. Asdornwised and S. Jitapunkul, "Automatic target recognition using multiple description coding models for multiple classifier systems", in Proc. of the 4th international conference on Multiple classifier systems (MCS 2003), June 2003, Guildford, UK, p. 336–345.

[11] L. I. Kuncheva and C. J. Whitaker, "Measures of diversity in classifier ensembles", Machine Learning, vol. 51, no.2, p. 181–207, 2003.

[12] B. V. Dasarathy (Editor), *A special issue on diversity in multiple classifier systems*, Information Fusion, vol. 6, no. 1, 2005.

[13] Y. S. Huang and C. Y. Suen. "The behavior-knowledge space method for combination of multiple classifers", in: Proc. of IEEE Conference on Computer Vision and Pattern Recognition, New York, USA, 1993, p. 347-352.

[14] P. Verlinde and G. Ghollet. "Comparing decision fusion paradigms using k-NN based classifiers, decision trees and logistic regression in a multi-modal identity verification application", In: Proc. of the 2nd International Conference on Audio and Video Based Biometric Person Authentication, 1999, Washington D.C, p. 188–193.

[15] A. Narasimhamurthy, "Theoretical bounds of majority voting performance for a binary classification problem", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27 no. 12, p. 1988–1995, 2005.

[16] G. Shafer, *A Mathematical Theory of Evidence*, Princeton, NJ: Princeton University Press, 1976.

[17] A.-L. Jousselme, D. Grenier, and E. Bosse, "A new distance between two bodies of evidence", Information Fusion, vol. 2, no. 2, p. 91–101, 2001.

[18] P. Smets, The transferable belief model, Artificial Intelligence, vol. 66, no. 2, p. 191–234, 1994.

[19] F. Smarandache and J. Dezert (Editors), *Applications and Advances of DSmT for Information Fusion (Vol III)*, American Research Press, Rehoboth, NM, USA, 2009.

[20] M.CH. Masson and T. Denoeux, "ECM: An evidential version of the fuzzy c-means algorithm", Pattern Recognition, vol. 41, no. 4, p. 1384–1397, 2008.

[21] A.-L. Jousselme and P. Maupin, "Distances in evidence theory: Comprehensive survey and generalizations", International Journal of Approximate Reasoning, vol. 53, no. 2, p. 118–145, 2012.

[22] M. Bouchard, A.-L. Jousselme, P.-E. Doré, "A proof for the positive definiteness of the Jaccard index matrix", International Journal of Approximate Reasoning, vol. 54, no. 5, p. 615–626, 2013.

[23] D. T. Pham, *Intelligent Optimisation Techniques(1st Edition)*, Springer-Verlag, London, 2000.

[24] C. L. Blake and C. L. Merz. UCI repository of machine learning databases. 1998. See http://www.ics.uci.edu/∼mlearn/MLRepository.html

[25] L. A. Zadeh. "A simple view of the Dempster-Shafer theory of evidence and its implication for the rule of combination," AI magazine, 1986, vol. 2, no. 7, p. 85–90.