Available online at www.sciencedirect.com
ScienceDirect
Pattern Recognition 40 (2007) 2198–2210
PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/pr

Pairwise fusion matrix for combining classifiers

Albert H.R. Ko[a,*], Robert Sabourin[a], Alceu de Souza Britto Jr.[b], Luiz Oliveira[b]

[a]*LIVIA, École de Technologie Supérieure, University of Quebec, 1100 Notre-Dame West Street, Montreal, Que., Canada H3C 1K3*
[b]*PPGIA, Pontifical Catholic University of Parana, Rua Imaculada Conceicao, 1155, PR 80215-901, Curitiba, Brazil*

Received 6 July 2006; received in revised form 13 October 2006; accepted 26 January 2007

Abstract

Various fusion functions for classifier combination have been designed to optimize the results of ensembles of classifiers (EoC). We propose a pairwise fusion matrix (PFM) transformation, which produces reliable probabilities for the use of classifier combination and can be amalgamated with most existent fusion functions for combining classifiers. The PFM requires only crisp class label outputs from classifiers, and is suitable for high-class problems or problems with few training samples. Experimental results suggest that the performance of a PFM can be a notch above that of the simple majority voting rule (MAJ), and a PFM can work on problems where a behavior–knowledge space (BKS) might not be applicable.

Keywords: Fusion function; Combining classifiers; Confusion matrix; Pattern recognition; Majority voting; Ensemble of learning machines

1. Introduction

Different classifiers usually make different errors on different samples, which means that we can arrive at an ensemble that makes more accurate decisions by combining classifiers [1–9]. For this purpose, diverse classifiers are grouped together into what is known as an ensemble of classifiers (EoC). There are two problems in optimizing the performance of an EoC: first, how classifiers are selected, given a pool of different classifiers, to construct the best ensemble; and second, given all the selected classifiers, choosing the best rule to combine their outputs. These problems are fundamentally different, and should be solved separately to reduce the complexity involved in optimizing EoCs; the former focuses on ensemble selection [3,6,10–14] and the latter on ensemble combination, i.e. the choice of fusion functions [2,5,9,14,15]. Various fusion functions for classifier combination have been designed to facilitate a consensus decision from the outputs of each individual classifier. Through experimentation, some fusion functions have been shown to perform better than the single best classifier. But, we have no adequate understanding of the reasons why some classifier combination schemes are better than others [2,7,14,16,17].

An important consideration in classifier combination is that much better results can be achieved if diverse classifiers, rather than similar classifiers, are combined. There are several methods for creating diverse classifiers, among them are Random Subspaces [18], Bagging and Boosting [19–21]. The Random Subspaces method creates various classifiers by using different subsets of features to train them. Bagging generates diverse classifiers by randomly selecting subsets of samples to train classifiers. Boosting also uses parts of samples to train classifiers, but not randomly; in this case, difficult samples have a greater probability of being selected and easier samples have less chance of being used for training. To summarize, diverse classifiers are needed to optimize the performance of an EoC, as well as an adequate fusion function for classifier combination. A number of different combination schemes have been suggested [2,5–7,9,11,14,15,22,23]. In general, two kinds of fusion functions are available: (a) fusion functions of label outputs, such as majority voting, behavior–knowledge space (BKS), naive Bayes (NB) methods, etc. and (b) fusion functions of continuous-value outputs, which require the class probabilities

* Corresponding author.
E-mail addresses: albert@livia.etsmtl.ca (A.H.R. Ko), robert.sabourin@etsmtl.ca (R. Sabourin), alceu@ppgia.pucpr.br (A. Britto), soares@ppgia.pucpr.br (L. Oliveira).

0031-3203/$30.00
doi:10.1016/j.patcog.2007.01.031

outputs from classifiers. Different from the continuous-valued fusion functions, the label outputs fusion functions could not apply *a posteriori* probabilities of classes provided by each individual classifier. In the case where only class labels are offered as outputs by each individual classifier, then the simple majority vote rule (MAJ) is suggested.

To improve the performance of the fusion functions of label outputs, the BKS [11] has been proposed as an interesting fusion function that takes into account the interaction of classifiers. The method does not require any *a posteriori* probabilities of classes provided by each individual classifier. By contrast, it estimates the probability of each possible class label by constructing a table with $L+1$ dimensions for an ensemble of L classifiers, each dimension corresponds to the output of each classifier, and the additional dimension is for the true labels of concerned samples. By this means, with only the class label outputs of each classifier the BKS can estimate the likelihood of a given sample belonging to a class. The problem of the BKS is that it can apply only on low-dimensional problems. Moreover, in order to have an accurate probability estimation, it requires a large number of samples for the training.

On the other hand, the continuous-valued fusion functions require *a posteriori* probabilities of classes provided by each individual classifier and thus can use simple probability combination functions, such as sum, product, maximum and minimum. Moreover, they can also be more sophisticated classifier combination schemes than label outputs fusion functions, such as decision templates (DTs), Dempster–Shafer combination (DSC), fuzzy integral, or multilayer perceptrons (MLP) [6,11,22,23]. While it is true that these functions deal with the problem of combining classifiers as a problem of pattern recognition and take into account the interactions from classifiers, most of them do need further training. As insufficient training data usually lead to imperfect training, these sophisticated fusion functions might perform worse than the simple fusion functions [24]. It has, in fact, been suggested that, given insufficient training samples, simple fusion functions may outperform some trained fusion functions [24].

Herein lies the dilemma of EoCs. Given a limited number of samples, we need to take into account the interaction among classifiers. When the number of samples is too small, most trained fusion functions will not work well. For classifiers with crisp label outputs, this is especially a serious problem, because the number of fusion functions for label outputs is limited, and the BKS is suited neither to high-dimensional class problem nor to ensembles with a large number of classifiers. Therefore, we note three constraints for classifier combination: (a) classifiers without *a posteriori* probabilities of classes as outputs cannot use continuous-valued fusion functions; (b) trainable fusion functions need a number of samples for training, otherwise they will not perform well; (c) in most cases the independence of each classifier is the basic assumption. This assumption is, however, usually not true. Here are the key questions that need to be addressed:

(1) Can label outputs classifiers apply continuous-valued fusion functions?

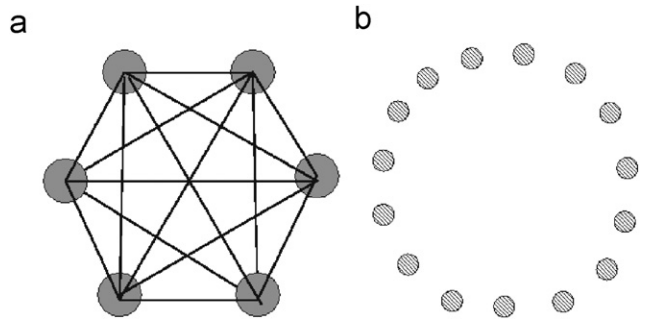

Fig. 1. An example of pairwise confusion matrices transformation in a six-classifier ensemble. (a) The original ensemble with six classifiers and (b) the transformation yields to 15 classifier pairs, each classifier pair is equal to the link between two classifiers in (a).

(2) Can a trainable fusion function perform well without a large training data set?
(3) Can we take the interaction among classifiers into account in combining classifiers?

Given the challenge of combining classifiers, we suggest that the methods for combining classifiers can be improved by a simple transformation of an EoC into an ensemble of classifier pairs. We propose a pairwise fusion matrix (PFM) for classifier combination. A PFM is actually a three-dimensional confusion matrix consisting of the label outputs of any two classifiers and the real labels of samples. It is a method for transforming EoCs (Fig. 1) by which an ensemble of L classifiers is transformed into another ensemble of $L \times (L-1)/2$ classifier pairs.

With the prospect of using classifier pairs, it becomes possible to transform the crisp class label outputs into class probability outputs and thus allow the use of other fusion functions of continuous-valued outputs. At the same time we do take into account the interaction between classifiers in a pairwise manner. Moreover, the construction of PFM does not require as many samples needed for ensemble training as the BKS.

It is important to note that the classifier combination problem is very complex, and there are still a great many issues associated with it that we do not fully understand. It is difficult to say whether or not a method is better if we have an insufficient theoretical framework with which to assess it. The analysis and the method in this paper constitute only a small step towards a considerably improved understanding of classifier combination.

The paper is organized as follows. In Section 2, we introduce label outputs fusion functions for classifier combination. The proposed pairwise confusion matrices are presented in Section 3, and we discuss its relationship with BKS in Section 4. Experimental results are compared in Section 5. Discussion and our conclusion are presented in the remaining sections.

2. Fusion functions for label outputs classifier combination

Several fusion functions of label outputs for combining classifiers have been proposed [2,7,16,17]. These directly compare the outputs from all individual classifiers in an ensemble. Some related theoretical studies are presented in Refs. [2,7,17]. As

stated in Refs. [17,25], most of these fusion functions rely on the very restrictive assumption of the independence of estimates. To address this shortcoming, other, more sophisticated strategies have been proposed which use more available information in combining classifiers [6,11,22,23]. We detail some popular fusion functions of label outputs in the section below.

2.1. Simple majority voting rule (MAJ)

This rule does not require the *a posteriori* outputs for each class, and each classifier gives only one crisp class output as a vote for that class. Then, the ensemble output is assigned to the class with the maximum number of votes among all classes. For any sample $x \in X$, for a group of L classifiers in a T-class problem, we denote the decision of label outputs from classifier $f(i)$ is $c(i), 1 \leqslant c(i) \leqslant T$, and we write $d_{i,t} = 1$ for $c(i) = t, 1 \leqslant t \leqslant T$ and zero otherwise. Consequently, we calculate the discriminant function for class $l, 1 \leqslant l \leqslant T$, as

$$g(l|x) = \sum_{i=1}^{L} d_{i,l}, \tag{1}$$

and the class is selected as the one with the maximum value of $g(l|x)$:

$$k = \arg\max_{l=1}^{T} g(l|x). \tag{2}$$

2.2. Weighted majority voting rule (W-MAJ)

Similar to MAJ, the weighted majority voting rule (W-MAJ) applies a voting scheme to decide the output class. However, in this case each classifier is weighted by a different coefficient:

$$g(l|x) = \sum_{i=1}^{L} b_i d_{i,l}, \tag{3}$$

where b_i is the coefficient for the classifier $f(i)$, with the sum equal to 1:

$$\sum_{i=1}^{L} b_i = 1. \tag{4}$$

It has been suggested that if each classifier is independent of one another, then the coefficient b_i can be set as [22]:

$$b_i \propto \log \frac{p_i}{1 - p_i}, \tag{5}$$

where p_i is the classification accuracy of classifier $f(i)$ on a training data set.

2.3. Naive bayes (NB)

Among these methods, the simplest is based on the assumption that all classifiers are mutually independent. Under this precondition, for a group of L classifiers in a T-class problem, we can calculate the probability $P(l|c(i), x)$ of the class label being $l, 1 \leqslant l \leqslant T$, if classifier $f(i)$ gives the class label output $c(i)$ on a sample x. Then we can use these estimated probabilities for classifying samples in the test set X:

$$\tilde{P}(l|x) \propto \prod_{i=1}^{L} P(l|c(i), x), \tag{6}$$

$$k = \arg\max_{l=1}^{T} \tilde{P}(l|x). \tag{7}$$

This is the so-called naive Bayes (NB) combination [6,7]. However, it is very unlikely that all classifiers in an ensemble will be mutually independent.

2.4. Behavior–knowledge space (BKS) and Wernecke's method (WER)

Some authors propose constructing a complex BKS table [11] in order to have full access to the information on classifier behavior. Given N samples and L classifiers in a T-class problem, the ideal goal is to obtain the probability $P(l|c(1), \ldots, c(i), \ldots, c(L), x)$ for the whole data X, where l is a possible class label for a sample $1 \leqslant l \leqslant T$, and $c(i)$ is the decision of classifier $f(i)$ over the sample, with L classifiers $1 \leqslant i \leqslant L$. Each probability can be located in a cell of a look-up table (BKS table), and then be used by multinomial combination, such as direct comparison of these probabilities in the BKS table, known as the BKS [11], or considering a 95% confidence interval of the probabilities in the BKS table, known as Wernecke's method (WER) [23]. For BKS, the class is assigned by simply comparing the values in each cell in BKS table:

$$k = \arg\max_{l=1}^{T} P(l|c(1), \ldots, c(i), \ldots, c(L), x). \tag{8}$$

In reality, however, this probability could be impossible to obtain. With L classifiers in a T-class problem, there are $T \times T^L$ different situations for this group of classifiers, and it is not difficult to see that the number of samples N is unlikely to be sufficient for T^{L+1} different situations, i.e. in general, $N \ll T^{L+1}$. As a result, obtaining any idea of this probability is also unlikely, and thus it is usually impossible to proceed with BKS or WER, except on low-class dimensions with a very small number of classifiers in an ensemble and a very large number of samples. Given the strict limit on the size of the training data set, some authors suggest that BKS tends to overfit [22], as well as being too self-assured [24].

Above all, it is remarkable that most trained fusion functions tend to explore more information from the training set. For this reason, most classifier combination strategies need to take the interaction between classifiers and between classes into consideration. If these elements are ignored, as with NB, then the performance cannot be satisfactory. If these elements are fully explored, as with BKS or WER, given the complicated behavior of classifiers in an ensemble, especially in a high-class dimension and with a large number of classifiers, the number of samples can scarcely be sufficient, and the probabilities obtained will usually be unreliable.

Herein lies the problem with training ensembles for combining classifiers. The fact that an ensemble acts in an extremely large space means that we need to use a method which is both effective and accurate. To partly resolve the problem, we propose a trained fusion function for better classifier combination in large-class dimension.

3. The concept of pairwise fusion matrices

3.1. Pairwise fusion matrix (PFM) transformation

The dilemma of EoCs is that, given a limited number of samples, we need to take into account the interaction among classifiers. Pairwise fusion matrix (PFM) transformation makes use of pairwise estimation to solve this problem. If we only take classifier pairs into account, we need only to calculate the probability $P(l|c(i), c(j), x)$, where $c(i)$ and $c(j)$ are the decisions of classifier $f(i)$ and classifier $f(j)$ over a sample x, respectively. For $P(l|c(i), c(j), x)$, there are only $T \times T^2 = T^3$ different situations, and if the number of samples N is large enough, i.e. $N \gg T^3$, we can obtain a reliable estimation of this probability. This probability can be approximated by calculating PFM:

$$P(l|c(i), c(j), x) = n(x \in l, c(i), c(j))/n(c(i), c(j)), \quad (9)$$

where $n(c(i), c(j))$ is the total number of samples on which classifier $f(i)$ gives crisp output $c(i)$ and classifier $f(j)$ gives crisp output $c(j)$, while $n(x \in l, c(i), c(j))$ is the number of samples the real class label of which is l, $1 \leqslant l \leqslant T$. The probability $P(l|c(i), c(j), x)$ is, in fact, the concept of a three-dimensional confusion matrix, where the decision of classifier $c(i)$, the decision of classifier $c(j)$ and the real class label of such samples represent each dimension.

The following is one example of a three-classifier PFM, which demonstrates the situation where the classifiers give different decisions. Suppose for a pattern x in a 10-class problem, the decision of the first classifier is 3, that of a second classifier is 8 and that of a third classifier is 5, i.e. $c(1) = 3$, $c(2) = 8$ and $c(3) = 5$. Obviously, for any class label l, PFM will give three probabilities based on different classifier pairs, $P(l|c(1) = 3, c(2) = 8, x)$, $P(l|c(1) = 3, c(3) = 5, x)$, and $P(l|c(2) = 8, c(3) = 5, x)$.

For any sample x with a class label k, PFM provides a pairwise matrix of classifier $f(i)$ and classifier $f(j)$ with the probability of how likely it will be classified as class $c(i)$ by $f(i)$ and as class $c(j)$ by $f(j)$. For any sample x classified as class l by classifier $f(i)$, PFM provides a partial confusion matrix between classifier $f(j)$ and the real class labels of samples. All the confusion matrices of classifier $f(j)$ can be derived quickly from any pairwise confusion matrices concerning $f(j)$:

$$P(l|c(j), x) = \sum_{i=1}^{T} P(l|c(i), c(j), x), \quad (10)$$

where $c(i)$ constitutes the class label outputs of classifier $f(i)$. In other words, it is a cube of T^3 cells with N samples filled in; since L classifiers mean $L \times (L-1)/2$ classifier pairs, we can obtain $L \times (L-1)/2$ pairwise confusion matrices (PFM).

Even though PFM is basically based on the label outputs of classifiers, it can also be constructed based on continuous-valued outputs of classifiers, in case it is applicable. If classifiers give the continuous class probability of each sample, PFMs can explore this property by calculating the probability-based PFM (PPFM):

$$P(l|c(i), c(j), x) = \frac{1}{N} \sum_{x=1}^{N} P(l|c(i), x) \cdot P(l|c(j), x), \quad (11)$$

where $P(l|c(i), x)$ is the probability of a class $c(i)$ being assigned by classifier $f(i)$ to sample x, the real class label of which is l, and $P(l|x, c(j))$ is the probability of a class $c(j)$ assigned by classifier $f(j)$ to sample x whose real class label is l.

The probabilities from these pairwise confusion matrices offer several advantages over the traditional ensemble combination strategies: (a) they do not require the class probability outputs of each sample but only the class label outputs of each sample from individual classifiers; (b) they transform the simple class label outputs into the class probability outputs; and (c) they take into account of the interaction between classifiers.

Note that the use of pairwise confusion matrices is a transformation that is to be combined with other fusion functions for the classifier combination. But, PFM allows the use of other fusion functions of continuous-value outputs and does not suppose the independence of each classifier. We show several examples of applied PFM on some fusion functions in the next section.

3.2. Apply PFM on fusion functions of continuous-value outputs

Based on these pairwise class probabilities, we can apply other different classifier combination rules. We give an example of the application of PFMs in general fusion functions of continuous-value outputs:

(1) PFM–maximum rule (PFM–MAX)

$$k = \arg\max_{l=1}^{T} \max_{i,j=1,i\neq j}^{L/2} P(l|c(i), c(j), x). \quad (12)$$

(2) PFM–minimum rule (PFM–MIN)

$$k = \arg\max_{l=1}^{T} \min_{i,j=1,i\neq j}^{L/2} P(l|c(i), c(j), x). \quad (13)$$

(3) PFM–sum rule (PFM–SUM)

$$k = \arg\max_{l=1}^{T} \frac{2}{L \times (L-1)} \sum_{i,j=1,i\neq j}^{L/2} P(l|c(i), c(j), x). \quad (14)$$

(4) PFM–product rule (PFM–PRO)

$$k = \arg\max_{l=1}^{T} \prod_{i,j=1,i\neq j}^{L/2} P(l|c(i), c(j), x). \quad (15)$$

Other fusion functions, such as DT or NB, will require further training, but are applicable as well. Furthermore, since the nature of pairwise confusion matrices is based on a pairwise approach, it is very likely that the probabilities displayed in the cells of pairwise confusion matrices can be weighted by the classification rates of classifiers and the pairwise diversity between classifiers. We discuss this idea in the next section.

3.3. Apply PFM on fusion functions of label outputs

Although one of the advantages of PFM lies in the use of fusion functions of continuous-value outputs, PFM can apply on fusion functions of label outputs as well. Given that MAJ can outperform some fusion functions of continuous-value outputs [24], we are interested to know if the PFM can bring about any improvement on MAJ.

We define this combination scheme as PFM–majority voting rule (PFM–MAJ). This rule is similar to the simple MAJ rule, but uses the pairwise probability $P(l|c(i), c(j), x)$ from the classifier pair $f(i)$ and $f(j)$ instead of the simple probability $P_i(l|x)$ from a single classifier $f(i)$ considering class l. For any sample $x \in X$, for a group of $L \times (L-1)/2$ classifier pairs in a T-class problem, we denote the decision of label outputs from classifiers $f(i)$ and $f(j)$ is $c(i)$ and $c(j)$, respectively:

$$\tilde{l} = \arg\max_{l=1}^{T} P(l|c(i), c(j), x). \quad (16)$$

We then denote $d_{i,j|t} = 1$ for $\tilde{l} = t$, $1 \leqslant t \leqslant T$ and zero otherwise. Consequently, we calculate the discriminant function for class l, $1 \leqslant l \leqslant T$ as

$$g(\hat{l}|x) = \sum_{i,j=1;i\neq j}^{L} d_{i,j|\tilde{l}}, \quad (17)$$

and the class is selected as the one with the maximum value of $g(\hat{l}|x)$:

$$k = \arg\max_{\hat{l}=1}^{T} g(\hat{l}|x). \quad (18)$$

Suppose for a pattern x in a 10-class problem classified by three classifiers with the decisions $c(1)=3$, $c(2)=8$ and $c(3)=5$. For any class label l, PFM gives the probabilities based on classifier pairs $P(l|c(1)=3, c(2)=8, x)$, $P(l|c(1)=3, c(3)=5, x)$, and $P(l|c(2)=8, c(3)=5, x)$. Suppose for all class label $1 \leqslant l \leqslant 10$, $P(3|c(1)=3, c(2)=8, x)$, $P(3|c(1)=3, c(3)=5, x)$ and $P(8|c(2)=8, c(3)=5, x)$ have the greatest probabilities based on its own classifier pairs. The class 3 has the support of the classifier pair $c(1)=3$, $c(2)=8$ and the classifier pair $c(1)=3$, $c(3)=5$, and the class 8 has the support of the classifier pair $c(2)=8$, $c(3)=5$, i.e. $d_{1,2|3}=1$, $d_{1,3|3}=1$ and $d_{2,3|8}=1$. As a result, the class 3 has more votes than the class 8 and any other class labels, since $g(3|x)=2$ and $g(8|x)=1$, the class 3 will be the decision of the EoC.

3.4. Other alternatives for PFM

We have shown that PFM can apply on both label outputs and continuous-value fusion functions. We also know that PFM can be constructed based on label outputs (PFM) or probability outputs (PPFM). PFM is, in fact, a flexible transformation that can allow us to apply various classifier combination schemes. Moreover, thanks to its pairwise nature, PFM can be further weighted by other factors. We give some examples of its alternatives:

(1) *PFM weighted by individual classifier recognition rate* (*PFM–IRR*):
Given the probability $P(l|c(i), c(j), x)$ from pairwise confusion matrices on an evaluated class k, where $c(i)$ and $c(j)$ are the decisions of classifier $f(i)$ and classifier $f(j)$, with $1 \leqslant i, j \leqslant L$, $i \neq j$ and $1 \leqslant l \leqslant T$, we can use the individual classifier recognition rate (IRR) $R(f(i))$ and $R(f(i))$ of classifier $f(i)$ and classifier $f(j)$, respectively, to weight the probability obtained (PFM–IRR):

$$\dot{P}(l|c(i), c(j), x) = P(l|c(i), c(j), x) * R(f(i)) * R(f(j)). \quad (19)$$

(2) *PFM weighted by diversity of classifier pair* (*PFM–DIV*):
If the pairwise diversity $div(f(i), f(j))$ between classifier $f(i)$ and classifier $f(j)$ is offered, we can use this property too. Note that there are two types of diversity measures. Diversity might measure the ambiguity between classifiers $f(i)$, $f(j)$, denoted $div_{amb}(f(i), f(j))$, or the similarity between classifiers $f(i)$, $f(j)$, denoted $div_{sim}(f(i), f(j))$. According to the different properties of diversity measures, we make use of them in different ways (PFM–DIV):

$$\ddot{P}(l|c(i), c(j), x) = P(l|c(i), c(j), x) * R(f(i)) * R(f(j)) * div_{amb} \times (f(i), f(j)), \quad (20)$$

$$\ddot{P}(l|c(i), c(j), x) = P(l|c(i), c(j), x) * R(f(i)) * R(f(j)) * (1 - div_{sim} \times (f(i), f(j))). \quad (21)$$

(3) *PFM weighted by class probabilities* (*PFM–P*):
In a case where an *a posteriori* probability of each class is given by classifiers, a PFM can be weighted by this confidence value as well (PFM–P):

$$\breve{P}(l|c(i), c(j), x) = P(l|c(i), c(j), x) * P(c(i)|x) * P(c(j)|x), \quad (22)$$

where $P(c(i)|x)$ is the *a posteriori* probability of class $c(i)$ that classifier $f(i)$ assigns to a sample x.

In order to prove that PFMs are applicable, we need to carry out the experiments on classifier combination. But before that,

we shall discuss the similarity and the difference of PFM and BKS, which is one of the most popular fusion functions of label outputs. Since PFM transforms a group of classifiers into another group of classifier pairs, we need to apply a certain fusion function on PFM so that we can compare it and understand its relationship with BKS. Given that MAJ is one of the most used fusion functions of label outputs, we decide to focus on PFM–MAJ on our discussion.

4. The relationship between BKS and PFM–MAJ

To better understand the relationship between the BKS and the PFM, we start with a simplified 2-class problem. Supposing three classifiers f_i, f_j, f_k are constructed for BKS, the class l_{max} is selected among all classes l, $1 \leqslant l \leqslant L$ as the ensemble output on a sample x if:

$$l_{max} = \arg\max_{l} n(l|c_i, c_j, c_k), \tag{23}$$

where $n(l|c_i, c_j, c_k)$ is the number of samples found in the BKS table. It refers to the number of samples with the real class l being classified as class c_i, c_j, c_k by three classifiers f_i, f_j, f_k, respectively.

For the PFM–MAJ, the decision is made by the outputs of three classifier pairs, $l_{max}(c_i, c_j)$, $l_{max}(c_i, c_k)$ and $l_{max}(c_j, c_k)$.

$$l_{max}(c_i, c_j) = \arg\max_{l} n(l|c_i, c_j). \tag{24}$$

Now, we notice the relationship between BKS and PFM–MAJ, for there is a direct relationship between $n(l|c_i, c_j, c_k)$ and $n(l|c_i, c_j)$:

$$n(l|c_i, c_j) = n(l|c_i, c_j, c_k) + n(l|c_i, c_j, \bar{c_k}), \tag{25}$$

where $\bar{c_k}$ is any class outputs different from c_k from the classifier f_k. As a result, $l_{max}(c_i, c_j)$ can be written as

$$l_{max}(c_i, c_j) = \arg\max_{l}(n(l|c_i, c_j, c_k) + n(l|c_i, c_j, \bar{c_k})). \tag{26}$$

For any class outputs $\bar{l_{max}} \neq l_{max}$, this indicates that

$$\begin{aligned} &n(l_{max}|c_i, c_j, c_k) + n(l_{max}|c_i, c_j, \bar{c_k}) \\ &\quad > n(\bar{l_{max}}|c_i, c_j, c_k) + n(\bar{l_{max}}|c_i, c_j, \bar{c_k}). \end{aligned} \tag{27}$$

The sufficient condition that guarantees $l_{max}(c_i, c_j) = l_{max}$ is thus that

$$\begin{aligned} &n(l_{max}|c_i, c_j, c_k) - n(\bar{l_{max}}|c_i, c_j, c_k) \\ &\quad > n(\bar{l_{max}}|c_i, c_j, \bar{c_k}) - n(l_{max}|c_i, c_j, \bar{c_k}). \end{aligned} \tag{28}$$

Note that from the BKS, we already know that

$$n(l_{max}|c_i, c_j, c_k) > n(\bar{l_{max}}|c_i, c_j, c_k), \tag{29}$$

so that the first term of the above equation is greater than 0:

$$n(l_{max}|c_i, c_j, c_k) - n(\bar{l_{max}}|c_i, c_j, c_k) > 0. \tag{30}$$

This indicates that PFM–MAJ is different from BKS, although they have a strong relationship. In some certain cases, they might produce the same results. In other cases, they will lead to different decisions. But, we do not know whether PFM–MAJ can perform better than BKS. For other PFM-related fusion functions such as PFM–SUM, PFM–PRO, PFM–MAX and PFM–MIN, we have even less understanding about the relationship with BKS. We could, however, compare their performances and have a general idea on whether it is adequate to apply PFM. For this reason, we carry out experiments on UCI machine learning repository in the next section.

5. Experimental comparison of classifier combination rules of crisp label outputs

Contrary to the fusion methods of continuous-valued outputs, until now there are only few fusion methods of crisp label outputs. The PFM is a practical concept and might be a good solution for the crisp label output combination. It has three fundamental aspects different from other fusion functions: First, it requires only crisp label outputs and not the continuous-valued outputs. Second, it is actually a transformation from the crisp label outputs of classifiers to the continuous-valued outputs of classifier pairs. Third, in general, PFM is itself not a fusion function, it should be applied on other existing fusion functions like SUM, majority voting, etc.

This paper focuses thus on the comparison of PFM and other fusion methods of crisp label outputs, such as the (NB) combination, the BKS, the Majority Vote (MAJ) and the Weighted Majority Vote (W-MAJ). The PFM is combined with some simple fusion functions such as SUM, MAJ, MAX, MIN and MAJ. Note that for every fusion function, we can always carry out the PFM. Although it is possible for us to combine PFM with other more sophisticated fusion functions, this will require more training. In this paper we only evaluate the PFM combined with the simple fusion functions.

For the experiments, we think it is important to evaluate the PFM on different ensemble creation methods, namely Random Subspaces, Bagging and Boosting, and these experiments were carried out on the problems extracted from the UCI machine learning repository. We also regard it important to evaluate the PFM on a large database with a large ensemble size, so we carried out an experiment on a 10-class handwritten numeral problem extracted from *NIST SD*19 with 100 classifiers. The experimental protocols and the results are shown in the following sections.

5.1. Experiments on UCI machine learning repository

To ensure that the PFM is useful for combining classifiers, we tested it on problems extracted from a UCI machine learning repository. There are several requirements for the selection of pattern recognition problems. First, to avoid identical samples being trained in Random Subspace, only databases without symbolic features are used. Second, to simplify the problem, we do not use databases with missing features. In accordance with the requirements listed above, we carried out our experiments on 13 databases selected from the UCI data repository (see Table 1). Among available samples, in general, 50% are used as a training data set, and 50% are used as a test data

Table 1
UCI data for ensembles of classifiers

Database	Classes	Tr	Ts	Features	RS-Card.	Bagging (%)	Boosting (%)
Ionosphere	2	175	175	34	20	66	66
Liver disorders	2	172	172	6	4	66	66
Pima diabetes	2	384	384	8	4	66	66
Wisconsin breast cancer	2	284	284	30	5	66	66
Iris	3	75	75	4	2	66	66
Wine	3	88	88	13	6	66	66
New thyroid	3	107	108	5	3	66	66
Vehicle	4	423	423	18	16	66	66
Satellite	6	4435	2000	36	6	66	66
Glass	7	107	107	10	8	66	66
Image segmentation	7	210	2100	19	4	66	66
Vowel	11	495	495	10	8	66	66
Letter recognition	26	10 000	10 000	16	12	66	66

Tr: training samples; Ts: test samples; RS-Card: random subspace cardinality; Bagging: proportion of samples used for bagging; Boost: proportion of samples used for boost.

Table 2
Comparison of recognition rates of different fusion functions with *Random Subspace* on UCI machine learning problems

Fusion functions	MAJ (%)	NB (%)	BKS (%)	PFM–MAJ (%)	PFM–SUM (%)	W-MAJ (%)
Ionosphere	81.39 (0.09)	81.47 (0.06)	90.75 (–)	83.10 (0.06)	81.09 (0.07)	80.46 (0.06)
Liver disorders	63.90 (0.11)	56.53 (0.24)	81.01 (0.04)	65.28 (0.08)	64.96 (0.08)	64.10 (0.06)
Pima diabetes	78.94 (0.16)	60.23 (0.60)	83.68 (0.03)	80.34 (0.06)	78.30 (0.05)	79.40 (0.03)
Wisconsin Breast cancer	93.54 (0.05)	93.68 (0.48)	92.14 (0.04)	94.17 (0.03)	93.54 (0.03)	93.78 (0.01)
Iris	90.06 (0.18)	91.53 (0.08)	88.81 (0.12)	93.21 (0.11)	91.84 (0.17)	91.52 (0.27)
Wine	84.42 (0.15)	89.96 (0.23)	94.76 (0.13)	90.30 (0.24)	88.82 (0.18)	85.92 (0.31)
New thyroid	95.27 (0.02)	88.04 (0.10)	91.80 (0.04)	94.95 (0.01)	93.91 (0.03)	95.43 (0.03)
Vehicle	68.08 (0.01)	63.66 (0.03)	63.87 (0.02)	67.01 (0.01)	68.20 (0.01)	68.18 (0.01)
Satellite	93.64 (–)	94.03 (–)	–	94.37 (–)	93.72 (–)	93.64 (–)
Glass	94.27 (0.50)	76.85 (0.43)	–	95.57 (0.24)	94.88 (0.26)	92.99 (1.09)
Image segmetation	75.91 (0.51)	64.78 (2.88)	–	85.31 (0.19)	82.98 (0.17)	73.92 (1.42)
Vowel	95.08 (0.01)	92.35 (0.02)	–	94.85 (0.01)	95.40 (–)	95.11 (0.01)
Letter	84.24 (0.04)	90.72 (0.04)	–	91.08 (0.09)	85.56 (0.09)	84.78 (0.03)
Fusion functions →	PFM–MIN (%)	PFM–MAX (%)	PFM–PROD (%)	PFM–IRR–MAJ (%)	PFM–DIV–MAJ (%)	
Ionosphere	79.66 (0.11)	67.59 (0.05)	79.76 (0.11)	82.89 (0.02)	82.86 (0.02)	
Liver disorder	64.41 (0.06)	56.14 (0.07)	65.13 (0.05)	65.33 (0.04)	65.26 (0.05)	
Pima diabetes	79.11 (0.02)	74.31 (0.01)	80.51 (0.04)	80.40 (0.04)	80.33 (0.03)	
Wisconsin Breast cancer	92.90 (0.03)	87.32 (0.07)	93.89 (0.01)	94.20 (0.01)	93.70 (0.02)	
Iris	89.04 (0.12)	86.39 (0.06)	88.96 (0.13)	93.36 (0.11)	92.88 (0.04)	
Wine	94.47 (0.11)	81.47 (0.08)	93.05 (0.13)	90.73 (0.23)	92.69 (0.08)	
New thyroid	84.87 (0.14)	90.29 (0.04)	85.09 (0.14)	95.13 (0.02)	94.61 (0.01)	
Vehicle	62.50 (0.03)	68.27 (0.01)	62.30 (0.03)	67.04 (0.01)	66.77 (0.01)	
Satellite	95.15 (–)	91.56 (0.01)	94.87 (–)	94.40 (–)	94.43 (–)	
Glass	84.98 (0.47)	86.71 (0.15)	85.07 (0.47)	96.28 (0.14)	90.01 (0.83)	
Image segmentation	91.43 (0.12)	53.80 (1.68)	90.85 (0.12)	86.32 (0.16)	87.67 (0.11)	
Vowel	90.34 (0.05)	91.83 (0.02)	90.48 (0.05)	94.90 (0.01)	93.89 (0.02)	
Letter	96.41 (0.02)	79.87 (0.04)	96.22 (0.02)	91.15 (0.02)	91.96 (0.01)	

All numbers are in percents (%), the variances are indicated in parenthesis. Note that three classification algorithms were used and only average values are shown here.

set, except for the Image Segmentation data set, whose training data set and test data set have been defined on UCI data repository. Of the training data set, 70% are used for classifier training and 30% are used for validation.

Three ensemble creation methods have been used in our study: Random Subspaces [10], Bagging and Boosting [26–28]. The Random Subspaces method creates various classifiers by using different subsets of features to train them. Bagging generates diverse classifiers by randomly selecting subsets of samples to train classifiers. Similar to Bagging, Boosting uses parts of samples to train classifiers as well, but not randomly; difficult samples have a greater probability of being selected, and easier samples have less chance of being used for training. Ensemble training (including BKS, NB and PFM) used the entire

Table 3
Comparison of recognition rates of different fusion functions with Bagging on UCI machine learning problems

Fusion Functions →	MAJ (%)	NB (%)	BKS (%)	PFM–MAJ (%)	PFM–SUM (%)	W-MAJ (%)
Ionosphere	78.40 (0.04)	77.07 (0.98)	91.04 (–)	79.81 (0.02)	79.49 (0.02)	79.20 (0.05)
Liver disorders	61.22 (0.08)	55.86 (0.02)	80.00 (0.03)	62.38 (0.08)	62.17 (0.07)	61.50 (0.06)
Pima diabetes	72.88 (0.01)	59.49 (0.01)	80.24 (0.02)	72.96 (0.01)	72.82 (0.01)	72.91 (0.01)
Wisconsin Breast cancer	94.27 (–)	94.36 (0.01)	94.32 (–)	94.53 (–)	94.27 (–)	94.34 (–)
Iris	91.32 (0.02)	92.51 (0.02)	88.81 (0.03)	92.09 (0.02)	91.77 (0.02)	91.66 (0.02)
Wine	78.71 (0.06)	79.41 (0.04)	78.50 (0.06)	80.05 (0.05)	79.08 (0.06)	78.86 (0.11)
New thyroid	92.14 (0.01)	89.48 (1.99)	91.73 (0.02)	92.33 (0.02)	90.98 (0.02)	92.39 (0.01)
Vehicle	67.29 (0.01)	65.74 (0.01)	64.82 (0.03)	67.01 (0.01)	67.23 (0.01)	67.26 (0.01)
Satellite	93.16 (–)	93.62 (–)	–	93.90 (–)	93.24 (–)	93.14 (–)
Glass	96.50 (–)	88.15 (–)	–	96.50 (–)	96.45 (–)	96.52 (0.01)
Image segmentation	86.22 (0.03)	87.78 (–)	–	89.02 (–)	86.68 (–)	88.77 (–)
Vowel	95.69 (0.02)	94.52 (0.01)	–	96.55 (0.02)	96.20 (0.02)	95.91 (0.01)
Letter	91.19 (–)	90.85(–)	–	92.79 (–)	94.30 (–)	90.87 (–)

Fusion functions →	PFM–MIN (%)	PFM–MAX (%)	PFM–PROD(%)	PFM–IRR–MAJ (%)	PFM–DIV–MAJ (%)
Ionosphere	79.55 (0.02)	66.41 (0.92)	79.63 (0.02)	79.97 (0.02)	79.79 (0.01)
Liver disorder	60.76 (0.09)	56.44 (0.05)	63.59 (0.07)	62.58 (0.08)	63.15 (0.09)
Pima diabetes	71.81 (0.01)	71.03 (0.01)	73.01 (0.01)	73.00 (0.01)	72.8867
Wisconsin Breast cancer	94.23 (0.01)	93.48 (–)	94.59 (–)	94.58 (–)	94.42 (–)
Iris	89.60 (0.03)	87.87 (0.03)	89.60 (0.03)	92.10 (0.02)	92.18 (0.02)
Wine	76.48 (0.10)	64.58 (0.20)	76.41 (0.11)	80.01 (0.06)	79.92 (0.05)
New thyroid	90.84 (0.03)	89.25 (0.01)	90.88 (0.03)	92.46 (0.02)	92.73 (0.02)
Vehicle	63.60 (0.02)	66.61 (0.01)	64.11 (0.02)	66.96 (0.01)	67.04 (0.01)
Satellite	94.80 (–)	90.03 (0.01)	94.54 (–)	93.94 (–)	93.92 (–)
Glass	94.60 (0.01)	95.34 (–)	94.66 (0.01)	96.54 (–)	96.28 (0.01)
Image segmentation	85.14 (0.02)	85.88 (0.01)	85.14 (0.02)	89.10 (–)	89.04 (–)
Vowel	91.84 (0.03)	86.80 (0.03)	91.89 (0.03)	96.61 (0.01)	96.38 (0.02)
Letter	87.54 (0.02)	93.48 (–)	87.61 (0.02)	92.89 (–)	92.49 (–)

All numbers are in percents (%), the variances are indicated in parenthesis. Note that three classification algorithms were used and only average values are shown here.

available training data set. The cardinality of Random Subspace is set under the condition that all classifiers have recognition rates of more than 50%.

The three different classification algorithms used in our experiments are K-nearest neighbors classifiers (KNN), Parzen windows classifiers (PWC) and quadratic discriminant classifiers (QDC) [29]. For each of the 13 databases and for each of the three classification algorithms, 10 classifiers were generated as the pool of classifiers. Among these, each classifier has a 50% chance of being selected from this pool to construct ensembles, ensembles were thus constructed by different numbers of classifiers, and at least three classifiers are required for an ensemble. As a result, all ensembles were constructed from 3 to 8 classifiers. Thirty ensembles had been generated for each database, for each ensemble generation method and for each classification algorithm. Note that each ensemble can have different number of classifiers. In total, we evaluated $30 \times 13 \times 3 \times 3 = 3510$ ensembles. We then combined these ensembles with 10 different fusion functions.

First, we see that the use of the PFM does make other continuous-valued fusion functions applicable, and PFM gives comparable results with other traditional label outputs fusion functions. Second, we also note that the best fusion function depends on the different problems, and the BKS is not always better than PFM applied fusion functions [14]. Third, among all the PFM applied fusion functions, we cannot figure out the best fusion function for PFM, but all PFM–MAJ, PFM–IRR–MAJ and PFM–DIV–MAJ have stable performances (Tables 2–4).

In previous studies, the BKS has been shown to be comparatively accurate when an ensemble of three classifiers is involved [19], but the BKS could be outperformed by most of the other fusion functions when more classifiers are involved [22]. In our study, the BKS apparently performs very well in 2- and 3-class problems (Tables 2–4). But when the class dimension is larger than 6, due to huge data size and limited computer memory we could not construct the BKS table.

Finally, if we compare the performance of the PFM–MAJ with that of the MAJ, which is concerned with one of the best fusion functions for classifiers with only crisp class label outputs [14], we find that in general the PFM–MAJ gives better performances than the simple MAJ rule, and in some cases comparable with that achieved by the BKS (Tables 2–4). The advantage of the PFM–MAJ over the simple MAJ might be due to the exploration of the interaction of classifiers from the PFM. The results are thus encouraging.

Nevertheless, the ensembles tested were constructed by randomly selected classifiers without any ensemble selection procedure. To better understand the effect of fusion functions on real problems, we must test this rule on a high-class problem with a large data set, and we need to go through the ensemble

Table 4
Comparison of recognition rates of different fusion functions with *Boosting* on UCI machine learning problems

Fusion functions →	MAJ (%)	NB (%)	BKS (%)	PFM–MAJ (%)	PFM–SUM (%)	W-MAJ(%)
Ionosphere	62.40 (0.74)	74.85 (0.77)	77.53 (2.02)	80.19 (0.01)	79.42 (0.12)	63.32 (2.65)
Liver disorders	61.43 (0.21)	57.22 (0.35)	80.76 (0.05)	64.09 (0.18)	64.07 (0.14)	63.46 (0.22)
Pima diabetes	70.09 (0.34)	68.59 (0.32)	79.28 (0.09)	71.37 (0.04)	70.26 (0.01)	70.17 (0.47)
Breast cancer	94.91 (–)	94.77 (–)	94.59 (–)	94.86 (–)	94.88 (–)	94.92 (–)
Iris	93.91 (0.01)	94.93 (0.01)	94.19 (–)	94.12 (0.01)	93.96 (0.01)	94.12 (0.03)
Wine	81.28 (0.02)	79.76 (0.05)	80.61 (0.04)	81.79 (0.02)	81.45 (0.02)	81.40 (0.02)
New thyroid	92.51 (–)	92.28 (–)	92.88 (–)	92.71 (–)	92.71 (–)	92.45 (–)
Vehicle	67.29 (–)	65.74 (0.01)	64.82 (0.02)	67.01 (0.01)	67.23 (–)	68.21 (–)
Satellite	96.39 (–)	96.57 (–)	–	96.66 (–)	96.43 (–)	96.40 (–)
Glass	95.96 (–)	88.18 (–)	–	95.95 (–)	95.95 (–)	95.96 (–)
Image	86.33 (–)	88.62 (–)	–	89.17 (–)	88.76 (–)	86.34 (–)
Vowel	97.90 (–)	97.00 (–)	–	97.87 (–)	97.96 (–)	97.91 (–)
Letter	92.23 (–)	93.96 (–)	–	94.70 (–)	93.31 (–)	92.05 (–)

Fusion functions →	PFM–MIN	PFM–MAX	PFM–PROD	PFM–IRR–MAJ	PFM–DIV–MAJ
Ionosphere	78.15 (0.04)	69.08 (0.27)	78.27 (0.04)	78.60 (0.04)	77.12 (2.07)
Liver disorder	62.89 (0.16)	55.22 (0.05)	63.89 (0.16)	64.26 (0.18)	64.28 (0.21)
Pima diabetes	71.88 (0.04)	69.35 (0.01)	71.78 (0.03)	71.56 (0.04)	71.49 (0.04)
Breast cancer	94.26 (–)	94.28 (–)	94.42 (–)	94.86 (–)	94.82 (–)
Iris	94.19 (–)	93.64 (0.01)	93.64 (–)	94.12 (0.01)	94.55 (0.01)
Wine	80.26 (–)	78.86 (–)	81.06 (–)	81.78 (–)	81.34 (–)
New thyroid	92.00 (–)	92.32 (0.01)	92.00 (–)	92.71 (–)	92.71 (–)
Vehicle	65.26 (0.02)	67.71 (–)	65.33 (0.02)	68.10 (0.01)	68.18 (–)
Satellite	96.85 (–)	95.41 (–)	96.83 (–)	96.67 (–)	96.72 (–)
Glass	95.95 (–)	96.00 (–)	95.95 (–)	95.95 (–)	95.95 (–)
Image	87.99 (–)	88.85 (–)	87.87 (–)	89.21 (–)	89.08 (–)
Vowel	96.35 (0.01)	96.71 (0.01)	96.34 (0.01)	97.90 (–)	97.78 (–)
Letter	94.29 (–)	92.00 (–)	94.25 (–)	94.72 (–)	94.83 (–)

All numbers are in percents (%), the variances are indicated in parenthesis. Note that three classification algorithms were used and only average values are shown here.

selection procedure. We then thus detail the further experiments in the next section.

5.2. Large size and high-dimensional ensembles: random subspace with KNN classifiers

Although experiments on the UCI machine learning repository suggest that the PFM is useful and stable for classifier combination, the results are still not reliable, for most problems on UCI machine learning repository have low-class dimensions, have few samples and have few features. Because of low-class dimensions, the problems are too simplified and not always fit to the real world problems; because of few samples, the Bagging and Boosting Ensemble Creation Methods cannot create diverse ensembles, and because of few features, the Random Subspace Ensemble Creation Method is strongly limited in its feature subspaces. It is doubtful that the experiments on the UCI machine learning repository can represent the qualities of the fusion functions in high-class problems with large data set.

To compensate this drawback of UCI data sets, we carry out further experiments on a well-known database, a handwritten numeral recognition problem known as *NIST SD*19. It is a 10-class problem and the problem includes more than 150 000 samples for the training and the validation, 60 089 samples for the test and a large number of features can be extracted from it. In our case more than 100 features were extracted from the patterns. We detail the experiments on the sections below.

5.2.1. Experimental protocol for KNN

We carried out experiments on a 10-class handwritten numeral problem. The data were extracted from *NIST SD*19, essentially as in Ref. [30], based on the ensembles of KNNs generated by the Random Subspaces method. We used nearest neighbor classifiers ($K = 1$) for KNN, each KNN classifier having a different feature subset of 32 features extracted from the total of 132 features. Four databases were used: the training set with 5000 samples ($hsf\_\{0–3\}$) to create 100 KNN in Random Subspaces, we use relatively small size of data set to better observe the impact of EoC. The optimization set containing 10 000 samples ($hsf\_\{0–3\}$) was used for genetic algorithm (GA) searching for ensemble selection. To avoid overfitting during GA searching, the selection set containing 10 000 samples ($hsf\_\{0–3\}$) was used to select the best solution from the current population according to the objective function defined, and then to store it in a separate archive after each generation. The same selection set was also used for training fusion functions, including PFM transformation and the NB fusion function. Note that with 100 classifiers and 10 classes, BKS and WER would require constructing a table with 10^{101} cells, which is impossible to realize. Using the best solution from this

Table 5
Mean recognition rates of ensembles selected by compound diversity functions and combined with various fusion functions

O.F. → /F.F. ↓	MVE	CFD	COR	DM	DF	DIFF	EN	GD	INT	KW	Q
MAJ (%)	96.45	96.22	96.29	96.19	96.20	96.23	96.18	96.19	96.22	96.20	96.20
W-MAJ (%)	96.47	96.24	96.25	96.21	96.20	96.25	96.22	96.25	96.26	96.18	96.24
NB (%)	96.27	95.78	95.77	95.79	95.76	95.80	95.75	95.75	95.81	95.74	95.79
PFM–MAJ (%)	96.94	96.88	96.88	96.84	96.82	96.87	96.85	96.86	96.87	96.82	96.86
PFM–IRR–MAJ (%)	96.94	96.88	96.87	96.84	96.82	96.87	96.85	96.86	96.87	96.82	96.86
PFM–DIV–MAJ(%)	96.95	96.89	96.88	96.86	96.81	96.87	96.87	96.87	96.87	96.84	96.86
PFM–MAX (%)	79.63	77.56	77.53	78.06	78.97	78.28	78.07	77.88	78.06	78.17	78.09
PFM–MIN (%)	78.00	70.76	70.28	71.29	71.88	69.99	70.66	70.29	70.81	71.28	70.64
PFM–SUM (%)	96.43	96.21	96.21	96.17	96.17	96.21	96.19	96.21	96.22	96.16	96.21
PFM–PROD (%)	71.04	70.37	69.99	70.55	70.90	69.73	70.06	69.68	69.97	70.64	69.89

The accuracy of the PFM–DIV–MAJ is the mean value of the results applying 10 different diversity measures. All variances are smaller than 0.01%. O.F.: Objective functions; F.F.: Fusion functions.

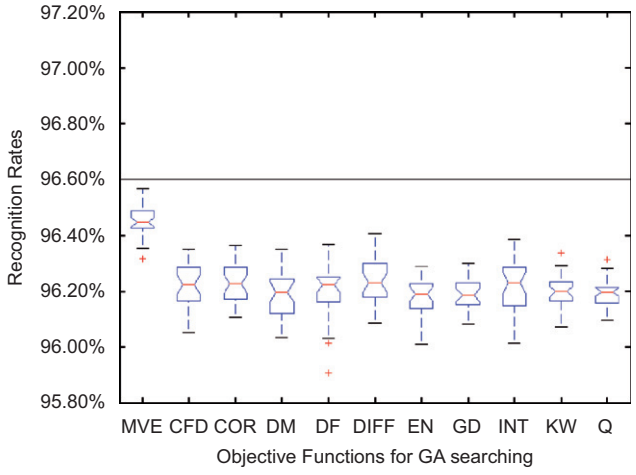

Fig. 2. The recognition rates achieved by EoCs selected by 10 compound diversity functions and majority voting error (MVE) using the simple MAJ as fusion function.

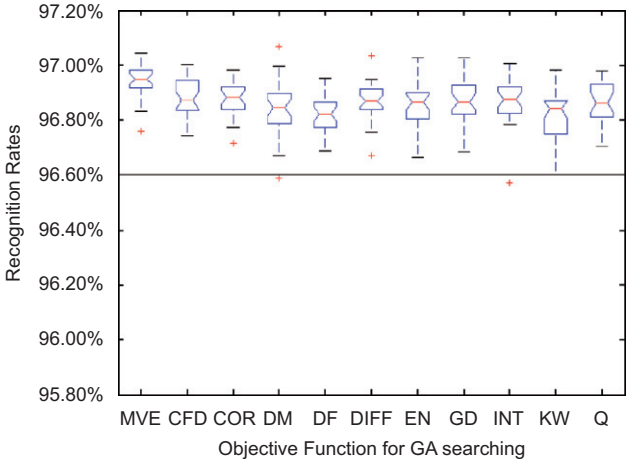

Fig. 3. The recognition rates achieved by EoCs selected by 10 compound diversity functions and majority voting error (MVE) using PFM–MAJ as fusion function.

archive, the test set containing 60 089 samples (*hsf*_{7}) was used to evaluate the EoC accuracies.

We need to address the fact that the classifiers used were generated with feature subsets having only 32 features out of a total of 132. The weak classifiers can help us better observe the effects of EoCs. If a classifier uses all available features and all training samples, a much better performance can be observed [27,28,31]. But, since this is not the objective of this paper, we focus on the improvement of EoCs by optimizing fusion functions on combining classifiers. The benchmark KNN classifier uses all 132 features, and so, with $K = 1$ we can have 93.34% recognition rates. The combination of all 100 KNN by simple MAJ gives 96.28% classification accuracy and gives 96.96% by PFM–MAJ. The possible upper limit of classification accuracy (the oracle) is defined as the ratio of samples which are classified correctly by at least one classifier in a pool for all samples. The oracle is 99.95% for KNN.

For evaluating classifier combinations, we first need to go through the process of ensemble selection, because one of the most important requirements of EoCs is that they contain diverse classifiers. We tested two kinds of different objective functions in this section. The majority voting error (MVE) was tested because of its reputation as one of the best objective functions in selecting classifiers for ensembles [14], it evaluates directly the global EoC performance by MAJ rule. In addition, we also tested 10 different traditional diversity measures and 10 different compound diversity measures which combine the pairwise diversity measures and individual classifier performance to estimate ensemble accuracy, but did not use the global EoC performance.

These objective functions are evaluated by GA searching. We used GA because the complexity of population-based searching algorithms can be flexibly adjusted depending on the size of the population and the number of generations with which to proceed. Moreover, because the algorithm returns a population of the best combinations, it can potentially be exploited to prevent generalization problems [14]. GA was set with 128 individuals in the population and 500 generations, which means that

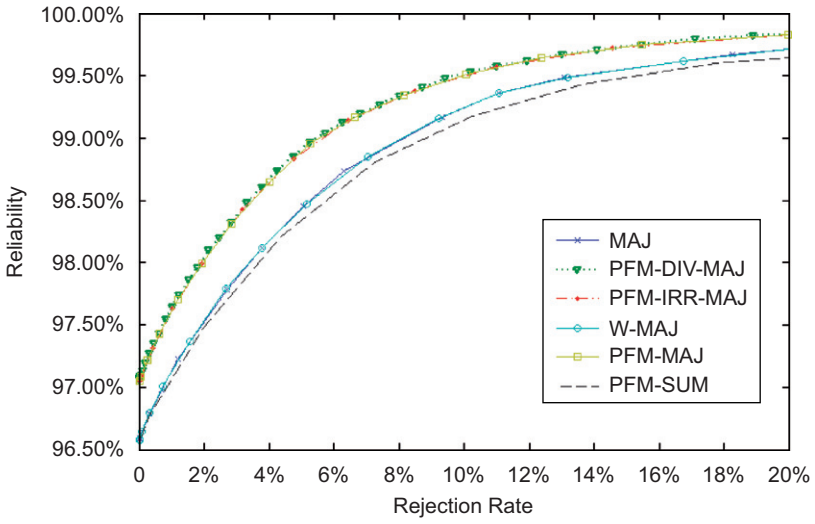

Fig. 4. The rejection curve of ensemble of KNNs selected by majority voting error (MVE) with evaluated fusion functions: MAJ, W-MAJ, PFM–SUM, PFM–MAJ, PFM–IRR–MAJ and PFM–DIV–MAJ. The accuracy of the PFM–DIV–MAJ is the mean value of the results applying 10 different diversity measures.

64 000 ensembles were evaluated in each experiment. The mutation probability is 0.01 with 11 different objective functions MVE and 10 compound diversity functions [32], including the disagreement measure (DM) [18], the double-fault (DF) [33], Kohavi–Wolpert variance (KW) [12], the interrater agreement (INT) [34], the entropy measure (EN) [3], the difficulty measure (DIFF) [35], generalized diversity (GD) [13], coincident failure diversity (CFD) [13], Q-statistics (Q) [36], and the correlation coefficient (COR) [3]), and with 30 replications. A threshold of three classifiers was applied as the minimum number of classifiers for an EoC during the whole searching process (Table 5). To summarize, 10 different fusion functions were tested.

We observe that, although traditional fusion functions like the MAJ, the W-MAJ and the NB have stable performances, the use of the PFM–MAJ, the PFM–IRR–MAJ and the PFM–DIV–MAJ can lead to a better performance (Table 5). Note that in this 10-class problem with 100 classifiers, it is impossible to apply the BKS.

We can observe that the advantage of using the PFM–MAJ instead of the MAJ is very clear (Figs. 2 and 3). By contrast, the PFM–MAX, the PFM–MIN and the PFM–PROD do not bring about any improvements. This is not surprising, since the MAX, the MIN, and the PROD rules have been regarded as suboptimal fusion functions compared with the SUM or the MAJ [14]. Given that 100 classifiers generate 4950 classifier pairs, an extremely biased value of the probability from any classifier pairs can affect the results seriously with the PFM–MAX, the PFM–MIN or the PFM–PROD rules.

The other fusion function that performs well and in a stable fashion is the PFM–SUM, the results of which are close to those achieved by the simple MAJ, but not yet as good as the PFM–MAJ. The PFM–SUM apparently outperforms the PFM–PROD in this respect (Table 5). A similar statement can be found in Ref. [15], where the authors suggest that the SUM is to be preferred over the PROD in the case where *a posteriori* probabilities are not well estimated. We thus suggest that the use of the PFM–MAJ or the PFM–SUM is more adequate than the PFM–MAX, the PFM–MIN or the PFM–PRO.

Until recently, there have been few other fusion functions that perform better than simple MAJ for crisp class label output classifiers. But, when PFM transformation is carried out, and those classifier pairs from ensembles are evaluated by the PFM–MAJ, we observe an improvement in the recognition rates of EoCs, the results achieved by the PFM–MAJ being a notch above those of the simple MAJ. This affirms the improvement brought about by the PFM (see Figs. 2 and 3).

We select the six best fusion functions for applying the rejection mechanism. In Fig. 4, we can observe that the MAJ and the W-MAJ have very similar performances, but the PFM–MAJ, the PFM–IRR–MAJ and the PFM–DIV–MAJ apparently outperform the MAJ and the W-MAJ. The advantage of the PFM–MAJ over the simple MAJ might be due to the exploration of the interaction of classifiers from the PFM. Using the information from the PFM, the system can achieve more accurate results. Interestingly, the performance of the PFM–SUM is not as good as the PFM–MAJ. This might indicate that the PFM might need more training samples to have a better estimation of the probability if we want to improve the performance of the PFM–SUM.

6. Discussion

For EoCs, the ideal is to obtain the probability $P(l|c(1),\ldots,c(i),\ldots,c(L),x)$ for the whole data set X, where l is the possible class label and $c(1),\ldots,c(i),\ldots,c(L)$ are decisions of individual classifiers $f(1),\ldots,f(i),\ldots,f(L)$, respectively. But, in reality, this approach might not work owing to the limitation with respect to the number of samples. Instead of estimating $P(l|c(1),\ldots,c(i),\ldots,c(L),x)$, the proposed method

deals with the probability $P(l|c(i), c(j), x)$ from pairwise confusion matrices on an evaluated class l, and thus is much more applicable, while at the same time taking into account classifier interaction.

When no class probability outputs are provided, most fusion functions, such as MAX, MIN, SUM and PRO, cannot be applied. The few available fusion functions are the simple MAJ, W-MAJ, NB or BKS, WER. However, for high-class problems and large size ensembles, there is no way to use BKS or WER, e.g. a 10-class problem with 100 classifiers requires the construction of a table with 10^{101} cells. Nevertheless, with PFM, we do not need as many samples as with BKS, PFM is a cube with 10^3 cells in this case, a size which is quite a reasonable and modest.

Furthermore, we show that all kinds of fusion functions are applicable. The result is encouraging. On the tested the UCI machine learning problems, the PFM–MAJ usually outperforms the simple MAJ as a fusion function for combining classifiers. We also note that the best fusion function seems to be problem-dependent, the PFM–DIV–MAJ, the PFM–IRR–DIV, the PFM–SUM, the PFM–MAX, the PFM–MIN and the PFM–MAX can slightly outperform the PFM–MAJ in some cases. Although we cannot figure out the best fusion function for the PFM, this shows that the use of the PFM allows the application of other continuous-valued fusion functions, and there will be many more choices of fusion functions for combining classifiers with only crisp class outputs.

To demonstrate that the advantages of PFM are not limited by the random classifier selection on the UCI machine learning repository, we apply the ensemble selection scheme with 10 compound diversity functions [32] on the *NIST SD*19 database. We can observe that the advantage of using the PFM–MAJ instead of the MAJ is very clear (Figs. 2 and 3).

The key element that makes an ensemble of classifier pairs outperform an EoC is that the use of the PFM takes the interaction into consideration. The pairwise manner may still be sub-optimal, but, if the class dimension is low and we have few classifiers and a large number of samples, PFM can be upgraded to the third degree, i.e. we can obtain the probabilities of any class label l by calculating $P(l|c(i), c(j), c(h), x)$ based on three classifier outputs $c(i), c(j), c(h)$. This would require the construction of four-dimensional confusion matrices and allow us to interpret the interaction of three classifiers at the same time. The use of diversity could further improve the recognition rates slightly in some cases, but not significantly.

7. Conclusion

In this paper, we propose a pairwise fusion matrix (PFM) transformation for classifier combination. PFM has some advantages:

(1) It transforms crisp class label outputs into class probability outputs.
(2) It is suited to most kinds of existing fusion functions for combining classifiers.
(3) It takes into account the interaction of classifiers in a pairwise manner.
(4) Because of its pairwise nature, it does not need too many samples for training compared with BKS or WER.

The experiment reveals that the performance of PFM is encouraging. Intuitively, the PFM can also be used for other trained fusion functions, such as NB or DT [22]. This will require another training, but we are interested in investigating the potential use of PFM in improving the performance of trained fusion functions.

Another possible improvement scheme would be the use of PFM–MAJ directly as an objective function for ensemble selection. In the same way that the simple MAJ is used for ensemble selection (i.e. MVE) and for classifier combination, one can also apply the PFM–MAJ for both ensemble selection and classifier combination.

The use of diversity might slightly improve the methods for classifier combination in some problems, but the effect is not significant. We suggest that more attention be paid to the possibility of using diversity for classifier combination in the future.

Acknowledgment

This work was supported in part by Grant OGP0106456 to Robert Sabourin from the NSERC of Canada.

References

[1] L.K. Hansen, C. Liisberg, P. Salamon, The error-reject tradeoff, Open Syst. Inf. Dyn. 4 (1997) 159–184.
[2] J. Kittler, M. Hatef, R. Duin, J. Matas, On combining classifiers, IEEE Trans. Pattern Anal. Mach. Intell. 20 (3) (1998) 226–239.
[3] L.I. Kuncheva, C.J. Whitaker, Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy, Mach. Learn. 51 (2) (2003) 181–207.
[4] D. Opitz, R. Maclin, Popular ensemble methods: an empirical study, J. Artif. Intell. Res. 11 (1999) 169–198.
[5] E. Pekalska, M. Skurichina, R.P.W. Duin, Combining dissimilarity-based one-class classifiers, In: International Workshop on Multiple Classifier Systems (MCS), 2004, pp. 122–133.
[6] C.A. Shipp, L.I. Kuncheva, Relationships between combination methods and measures of diversity in combining classifiers, Int. J. Inf. Fusion 3 (2) (2002) 135–148.
[7] L. Xu, A. Krzyzak, C.Y. Suen, Methods of combining multiple classifiers and their applications to handwriting recognition, IEEE Trans. Syst. Man Cybern. 22 (3) (1992) 418–435.
[8] G.I. Webb, Z. Zheng, Multistrategy ensemble learning: reducing error by combining ensemble learning techniques, IEEE Trans. Knowl. Data Eng. 16 (8) (2004) 980–991.
[9] H. Zouari, L. Heutte, Y. Lecourtier, A. Alimi, Building diverse classifier outputs to evaluate the behavior of combination methods: the case of two classifiers, In: International Workshop on Multiple Classifier Systems (MCS), 2004, pp. 273–282.
[10] R.E. Banfield, L.O. Hall, K.W. Bowyer, W.P. Kegelmeyer, A new ensemble diversity measure applied to thinning ensembles, in: International Workshop on Multiple Classifier Systems (MCS 2003), 2003, pp. 306–316.
[11] Y.S. Huang, C.Y. Suen, A method of combining multiple experts for the recognition of unconstrained handwritten numerals, IEEE Trans. Pattern Anal. Mach. Intell. 17 (1995) 90–93.

[12] R. Kohavi, D.H. Wolpert, Bias plus variance decomposition for zero-one loss functions, In: Proceedings of the International Machine Learning Conference (ICML), 1996, pp. 275–283.
[13] D. Partridge, W. Krzanowski, Software diversity: practical statistics for its measurement and exploitation, Inf. Software Technol. 39 (1997) 707–717.
[14] D. Ruta, B. Gabrys, Classifier selection for majority voting, Int. J. Inf. Fusion (2005) 63–81.
[15] D.M.J. Tax, M. Van Breukelen, R.P.W. Duin, J. Kittler, Combining multiple classifiers by averaging or by multiplying, Pattern Recognition 33 (9) (2000) 1475–1485.
[16] R.P. W. Duin, The combining classifier: to train or not to train?, in: 16th International Conference on Pattern Recognition (ICPR), vol. 2, 2002, pp. 20765.
[17] L.I. Kuncheva, A theoretical study on six classifier fusion strategies, IEEE Trans. Pattern Anal. Mach. Intell. 24 (2) (2002) 281–286.
[18] T.K. Ho, The random space method for constructing decision forests, IEEE Trans. Pattern Anal. Mach. Intell 20 (8) (1998) 832–844.
[19] A. Grove, D. Schuurmans, Boosting in the limit: maximizing the margin of learned ensembles, in: Proceedings of the Fifteenth National Conference on Artificial Intelligence, 1998, pp. 692–699.
[20] L.I. Kuncheva, M. Skurichina, R.P.W. Duin, An experimental study on diversity for bagging and boosting with linear classifiers, Int. J. Inf. Fusion 3 (2) (2002) 245–258.
[21] R.E. Schapire, Y. Freund, P. Bartlett, W.S. Lee, Boosting the margin: a new explanation for the effectiveness of voting methods, Ann. Stat. 26 (5) (1998) 1651–1686.
[22] L.I. Kuncheva, Combining Pattern Classifiers. Methods and Algorithms, Wiley, New York, 2004.
[23] K.D. Wernecke, A coupling procedure for discrimination of mixed data, Biometrics 48 (1992) 97–506.
[24] S. Raudys, Experts boasting in trainable fusion rules, IEEE Tran. Pattern Anal. Mach. Intell. 25 (9) (2003) 1178–1182.
[25] K. Turner, J. Ghosh, Error correlation and error reduction in ensemble classifiers, Connection Sci. 8 (3–4) (2006) 385–404.
[26] K. Kimura, S. Inoue T. Wakabayashi, S. Tsuruoka, Y. Miyake, Handwritten numeral recognition using autoassociative neural networks, In: Proceedings of the International Conference on Pattern Recognition (ICPR), vol. 1, 1998, pp. 166–171.
[27] J. Milgram, M. Cheriet, R. Sabourin, Estimating accurate multi-class probabilities with support vector machines, in: International Joint Conference on Neural Networks 2005 (IJCNN), 2005, pp. 1906–1911.
[28] L.S. Oliveira, R. Sabourin, F. Bortolozzi, C.Y. Suen, Automatic recognition of handwritten numerical strings: a recognition and verification strategy, IEEE Trans. Pattern Anal. Mach. Intell. 24 (11) (2002) 1438–1454.
[29] R.P.W. Duin, Pattern recognition toolbox for Matlab 5.0+. 2003. Available free at: ⟨ftp://ftp.ph.tn.tudelft.nl/pub/bob/prtools⟩.
[30] G. Tremblay, R. Sabourin, P. Maupin, Optimizing nearest neighbour in random subspaces using a multi-objective genetic algorithm, In: Proceedings of the 17th International Conference on Pattern Recognition (ICPR), 2004, pp. 208–211.
[31] P.V.W. Radtke, R. Sabourin, T. Wong, Intelligent feature extraction for ensemble of classifiers, In: 8th International Conference on Document Analysis and Recognition (ICDAR), 2005, pp. 866–870.
[32] A. Ko, R. Sabourin, A. Britto Jr, Combining diversity and classification accuracy for ensemble selection in random subspaces, in: IEEE World Congress on Computational Intelligence (WCCI 2006)–International Joint Conference on Neural Networks (IJCNN), 2006.
[33] G. Giacinto, F. Roli, Design of effective neural network ensembles for image classification purposes, Image Vision Comput. 19 (9–10) (2001) 699–707.
[34] J.L. Fleiss, B. Levin, M.C. Paik, Statistical Methods for Rates and Proportions, second ed., Wiley, New York, 2003.
[35] L.K. Hansen, P. Salamon, Neural network ensembles, IEEE Trans. Pattern Anal. Mach. Intell 12 (1990) 993–1001.
[36] A.A. Afifi, S.P. Azen, Statistical Analysis: A Computer Oriented Approach, second ed., Academic Press, New York, 1979.

About the author—ALBERT HUNG-REN KO received M.Sc. A degree in Artificial Intelligence and Pattern Recognition from the Universite Pierre et Marie Curie in 2002. In 2003 he started his Ph.D. degree in Pattern Recognition in Ecole de Technologie Superieure, Universite du Quebec. His research interests are Ensemble Classification Methods, Small World Structure and Neural Networks.

About the author—ROBERT SABOURIN received B. ing, M.Sc.A, Ph.D. degrees in Electrical Engineering from the Ecole Polytechnique de Montreal in 1977, 1980 and 1991, respectively. In 1977, he joined the physics department of the Universite de Montreal where he was responsible for the design and development of scientific instrumentation for the Observatoire du Mont Megantic. In 1983, he joined the staff of the Ecole de Technologie Superieure, Universite du Quebec, Montreal, P.Q, Canada, where he is currently a professeur titulaire in the Departement de Genie de la Production Automatisee. In 1995, he joined also the Computer Science Department of the Pontificia Universidade Catolica do Parana (PUC-PR, Curitiba, Brazil) where he was coresponsible since 1998 for the implementation of a Ph.D. program in Applied Informatics. Since 1996, he is a senior member of the Centre for Pattern Recognition and Machine Intelligence (CENPARMI). His research interests are in the areas of handwriting recognition and signature verification for banking and postal applications.

About the author—ALCEU DE SOUZA BRITTO JR received M.Sc. degree in Industrial Informatic from the Federal Center for Technological Education of Parana (Brazil) in 1996, and Ph.D. degree in Computer Science from Pontifical Catholic University of Parana (PUC-PR, Brazil). In 1989, he joined the Computer Science Department of the Ponta Grossa University (Brazil). In 1995, he also joined the Computer Science Department of the PUC-PR. His research interests are in the areas of document analysis and handwriting recognition.

About the author—LUIZ OLIVEIRA received the B.S. degree in Computer Science from UnicenP, Curitiba, PR, Brazil, the M.Sc. degree in electrical engineering and industrial informatics from the Centro Federal de Educacao Tecnologica do Parana (CEFET-PR), Curitiba, PR, Brazil, and Ph.D. degree in Computer Science from Ecole de Technologie Superieure, Universite du Quebec in 1995, 1998, and 2003, respectively. His current interests include Pattern Recognition, Neural Networks, Image Analysis, and Evolutionary Computation.