ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

A THESIS PRESENTED TO THE
ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

IN PARTIAL FULFILLMENT OF THE THESIS
REQUIREMENT FOR THE DEGREE OF THE
PH.D. ENGINEERING

BY

ALBERT HUNG-REN KO

STATIC AND DYNAMIC SELECTION OF ENSEMBLE OF CLASSIFIERS

MONTREAL, OCTOBER 12, 2007

THIS THESIS WAS EVALUATED

BY A COMMITTEE COMPOSED BY :

Prof. Robert Sabourin, thesis director
Department of automated manufacturing engineering at École de technologie supérieure

Prof. Maarouf Saad, committee president
Department of electrical engineering at École de technologie supérieure

Prof. Laurent Heutte, external examiner
LITIS, Université de Rouen

Prof. Eric Granger, examiner
Department of automated manufacturing engineering at École de technologie supérieure

Prof. Alceu de Souza Britto, Jr., invited examiner
PPGIa, Pontifical Catholic University of Parana

THIS THESIS WAS DEFENDED IN FRONT OF THE EXAMINATION

COMMITTEE AND THE PUBLIC

ON OCTOBER 12, 2007

AT THE ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

ACKNOWLEDGMENTS

Thanks to all the friends and colleagues in LIVIA. They make nice tea, good coffee; they are skillful soccer players and wonderful chess partners. They improved my French, English, Spanish and Portuguese.

Thanks to my professor, Robert Sabourin. A demanding supervisor but an understanding teacher, a workaholic and a down-to-earth scientist, he read and directed all my works tirelessly. I would not achieve all these works without him.

A special thank is to Professor Alceu de Souza Britto, Jr. The EoHMM system is built on his HMM framework. He shares his codes with us and gives valuable comments on all my works.

Another thanks to Cynthia Orman. She is an English specialist and proof-read all my writings tirelessly.

Most of all, I want to thank my families in Taiwan, who supported me even 15,000 kms away, both financially and psychologically.

STATIC AND DYNAMIC SELECTION OF ENSEMBLE OF CLASSIFIERS

Albert Hung-Ren KO

ABSTRACT

This thesis focuses on different techniques of ensemble of classifier (EoC) methods that will help improve pattern recognition results.

Pattern recognition can, in general, be regarded as a problem of classification, where different patterns are presented and we need to classify them into specified classes. We create classifiers to perform the classification task. One way to improve the recognition rates of pattern recognition tasks is to improve the accuracy of individual classifiers, and another is to apply ensemble of classifiers (EoC) methods. EoC methods use multiple classifiers and combine their outputs. In general, the combined results of these multiple classifiers can be significantly better than those of the single best classifier. In this thesis, we only look into the techniques that improve EoC accuracy and not those that improve the accuracy of a single classifier.

Three major topics are associated with EoCs: ensemble creation, ensemble selection and classifier combination. In this thesis, we propose a new ensemble creation method for an ensemble of hidden markov models (EoHMM), three methods for ensemble selection for different circumstances, and a classifier combination method.

First and foremost, we propose compound diversity functions (CDF), which combine diversities with the performance of each individual classifier, and show that there is a strong correlation between the proposed functions and ensemble accuracy. We will demonstrate that most compound diversity functions are better than traditional diversity measures.

We also propose a pairwise fusion matrix (PFM) transformation, which produces reliable probabilities for the use of a classifier combination and can be amalgamated with most existing 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 that a PFM can work on problems where a Behavior Knowledge Space (BKS) might not be applicable.

Also proposed here is a new scheme for the optimization of codebook sizes for HMMs and the generation of HMM ensembles. By using a pre-selected clustering validity index, we show that HMM codebook size can be optimized without training HMM classifiers. Moreover, the proposed scheme yields multiple optimized HMM classifiers, and each individual HMM is based on a different codebook size.

Two other alternative ensemble selection methods are also proposed here: a dynamic ensemble selection method, and a classifier-free ensemble selection method. The former applies different ensembles for test patterns, and the experimental results suggest that in some cases it performs better than both static ensemble selection and dynamic classifier selection. The latter explores the idea of "data diversity" for data subset selection. We try to select adequate feature subsets for Random Subspaces, and only use the select data subsets to create classifiers.

The main objective of the proposed methods is to offer applicable approaches that might advance the state of the art. But EoC optimization is a very complex issue and is related to a number of varied processes, and our contribution is intended merely to provide an improved understanding of the use of EoCs.

SELECTION STATIQUE ET DYNAMIQUE DES ENSEMBLES DE CLASSIFICATEURS POUR LA RECONNAISANCE DE CHIFFRES MANUSCRITS

Albert Hung-Ren KO

SOMMAIRE

Cette thèse porte sur différents aspects concernant la création des ensembles de classificateurs (EoC) pour la mise en oeuvre de systèmes de reconnaissance de formes robustes.

La reconnaissance de formes peut être vue comme un problème de classification o des objets inconnus (patterns) doivent être associés à une classe d'appartenance. Afin de réaliser cette tâche, des classificateurs doivent être sélectionnés suite au processus d'apprentissage sur une base de données représentative du problème de reconnaissance. Une approche classique consiste à choisir le classificateur le plus performant sur une base de validation; une autre approche consiste à choisir et à combiner un ensemble de classificateurs. Il a été montré dans la littérature qu'en général, les EoC généralisent mieux que les classificateurs indivituels sur des nouvelles données. Dans cette thèse, plusieurs aspects traitant de la création des EoC sont analysés et plusieurs méthodes novatrices sont proposées afin d'obtenir des EoC les plus performants.

Trois mécanismes fondamentaux régissent la création des EoC : la génération des classificateurs individuels, la sélection des classificateurs les plus diversifiés et finalement la fusion des classificateurs pour former des EoC. Nous présentons dans cette thèse une nouvelle méthode pour la génération de HMM pour la création d'ensembles de HMM (EoHMM), trois nouvelles méthodes de sélection et une nouvelle méthode de fusion.

Dans un permier temps, une nouvelle fonction objective CFD est proposée pour la sélection des classificateurs pertinents. Cette fonction est basée sur les performances individuelles des classificateurs de l'ensemble et d'une mesure de diversité mesurée entre les pairs de classificateurs. Nous avons montré expérimentalement que la mesure de diversité proposée est supérieure aux mesures de diversité publiées dans la littérature pour la sélection des classificateurs.

Ensuite une nouvelle fonction de fusion basée sur une matrice de transformation pairwise (PFM) permet l'estimation fiable des probabilités a postériori dans les cas o le problème de reconnaissance comporte un grand nombre de classes. La transformation proposée a l'avantage d'être indépendante du type de sorties des classificateurs (étiquettes, scores, probabilités a posteriori, etc) et celle-ci est bien adaptée pour les bases d'apprentissage de petite taille. Nous avons montré empiriquement que la nouvelle fonction de fusion PFM montre en moyenne une meilleure performance que le vote majoritaire (MAJ), et se

comporte avantageusement par rapport à la méthode BKS dans plusieurs cas o le nombre de classes est très important.

Une nouvelle méthode pour la création des ensembles de HMM (EoHMM) est également proposée. Cette approche est basée sur le choix des N meilleurs codebooks choisis à partir de l'indice de validité des partitions XB, mesuré sur des partitions différentes de la base d'apprentissage. Un avantage de la méthode proposée est que le choix des meilleurs codebooks est effectué sans recourir à l'apprentissage des HMM. Le choix des codebooks pertinents est non supervisé et chaque modèle de l'ensemble est alors estimé sur un codebook comportant un nombre de centres différent.

Finalement deux nouvelles méthodes de sélection sont également proposées : la première est une nouvelle méthode pour la sélection dynamique des EoC basée sur le concept des Oracles (KNORA) et la deuxième repose sur le choix des sous-espaces de représentation basé sur une mesure de diversité entre les partitions obtenues dans ces sous-espaces. Cette dernière approche permet de choisir les espaces de représentation des classificateurs individuels indépendemment du choix de la machine d'apprentissage.

Les ensembles de classificateurs constituent une nouvelle approche pour la conception de systèmes de classification robustes. Cette thèse apporte quelques solutions novatrices pour tenter de faire avancer notre compréhension dans ce domaine de recherche en pleine expansion.

SELECTION STATIQUE ET DYNAMIQUE DES ENSEMBLES DE CLASSIFICATEURS POUR LA RECONNAISANCE DE CHIFFRES MANUSCRITS

Albert Hung-Ren KO

RÉSUMÉ

Les ensembles de classificateurs (EoC) permettent la mise en oeuvre de systèmes de reconnaissance de formes robustes. Nous présentons dans cette thèse plusieurs solutions novatrices pour tenter de solutionner trois problèmes fondamentaux reliés à la conception des EoC : la génération des classificateurs, la sélection et la fusion.

Une nouvelle fonction de fusion (Compound Diversity Function - CDF) basée sur la prise en compte de la performance individuelle des classificateurs et de la diversité entre pairs de classificateurs est proposée au chapitre un pour la sélection statique des ensembles. Un résultat important est la démonstration de l'existence d'une corrélation entre différentes versions de CDF et la performance globale de l'ensemble. De plus, nous avons montré que les variantes de CFD sont en général plus performantes pour la sélection statique des ensembles de classificateurs que les mesures de diversité publiées dans la littérature.

Le deuxième chapitre présente une nouvelle fonction de fusion basée sur les matrices de confusions "pairwise" (PFM), mieux adaptée pour la fusion des classificateurs en présence d'un grand nombre de classes. Cette méthode transforme les étiquettes des classes générées par les classificateurs en probabilités a postériori des classes. La méthode proposée est générale et s'applique à tous les types de classificateurs, peu importe la nature de la sortie (étiquettes, scores, probabilités à posteriori, etc). De plus, cette méthode est bien adaptée pour résoudre les problèmes de reconnaissance comportant un grand nombre de classes, et une base d'apprentissage de petite taille. Nous avons montré empiriquement que la nouvelle fonction de fusion PFM montre en général une meilleure performance que le vote majoritaire (MAJ), et se comporte avantageusement comparée à la méthode BKS dans plusieurs cas o le nombre de classes est très important.

Troisièmement, une nouvelle méthode est proposée pour générer des ensembles de Modèles de Markov Cachés (Hidden Markov Models - EoHMM) pour la reconnaissance des caractères manuscrits. Plusieurs hypothèses de codebooks sont générées à partir d'une mesure de validité des clusters. Le choix des codebooks est non supervisé, c'est-à-dire que le choix n'est pas basé sur la performance en généralisation des HMM mais a priori à partir de la qualité des partitions obtenues lors de la recherche du meilleur codebook. Nous avons observé que les modèles ainsi générés montrent une diversité d'opinions en généralisation ce qui permet la création de EoHMM performants. La validation de la méthode proposée sur la base de chiffres manuscrits NIST SD19 montre des résultats très encourageants.

Le chapitre quatre porte sur la sélection dynamique des ensembles de classificateurs. En effet, la sélection statique des ensembles de classificateurs suppose que le niveau de compétence du meilleur ensemble est élevé pour tous les exemples de test à classer. Cette remarque s'applique évidemment au choix du meilleur classificateur individuel. Une solution novatrice est proposée dans ce chapitre et repose sur le concept des Oracles associés aux données de la base de validation (KNORA). En effet, supposons une observation appartenant à la base de validation, la définition d'un Oracle réfère aux classificateurs individuels qui sont en mesure de classer correctement cette observation. La sélection dynamique consiste à localiser les observations de la base de validation qui sont dans le voisinage immédiat de l'exemple de test à classer et de constituer dynamiquement un ensemble de classificateurs défini par tous les oracles associés aux observations faisant parti de ce voisinage. Le principe de la méthode est simple et les résultats expérimentaux obtenus sont très prometteurs.

La méthode des sous-espaces aléatoires (Random Subspace Method - RSS) proposée par T.K. Ho permet la génération de pools de classificateurs diversifiés et bien adaptés pour la création des EoC. Actuellement il n'y a pas de méthode efficace pour la sélection des sous-espaces pertinents. Une nouvelle approche est proposée dans ce chapitre pour la sélection des sous-espaces de représentation à partir d'une mesure de diversité évaluée entre les paires de partitions. La première étape est de partitionner la base de validation en K clusters pour chaque sous-espace de représentation. L'hypothèse que nous posons est que la diversité entre les partitions dans les sous-espaces est reliée à la diversité d'opinions des classificateurs spécialisés dans ces mêmes sous-espaces de représentation. Nous avons montré expérimentalement que cette relation existe et que le choix des sous-espaces de représentation qui montrent une grande diversité permet de générer des pools de classificateurs adaptés pour la création des EoC. Un avantage important de la méthode proposée est que le choix des sous-espaces de représentation est indépendant du choix de la machine d'apprentissage.

Les méthodes proposées dans les cinq chapitres ont été soumis dans des journaux spécialisés et reconnus dans notre domaine de recherche (Pattern Recognition, International Journal on Pattern Recogniton and Artificial Intelligence, Pattern Analysis and Application et TPAMI). De plus, plusieurs communications dans les conférences internationales ont également été présentées (GECCO2006, IJCNN2006, ICPR2006, MCS2007 et ICDAR2007). Les ensembles de classificateurs constituent une nouvelle approche pour la conception de systèmes de classification robustes. Cette thèse apporte quelques solutions novatrices pour tenter de faire avancer notre compréhension dans ce domaine de recherche en pleine expansion.

TABLE OF CONTENT

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OF ABBREVIATIONS

ABS	Absolute Optima
ALL-HMM	Ensemble of both Column and Row Hidden Markov Model Classifiers
BKS	Behavior Knowledge Space
CDF	Compound Diversity Function
CDF-CFD	Compound Diversity Function using Coincident Failure Diversity
CDF-COR	Compound Diversity Function using Correlation Coefficient
CDF-DF	Compound Diversity Function using Double Fault
CDF-DIFF	Compound Diversity Function using Difficulty Measure
CDF-EN	Compound Diversity Function using Entropy Measure
CDF-GD	Compound Diversity Function using Generalized Diversity
CDF-INT	Compound Diversity Function using Interrater Agreement
CDF-KW	Compound Diversity Function using Kohavi-Wolpert Variance
CDF-Q	Compound Diversity Function using Q-Statistics
CFD	Coincident Failure Diversity
COL-HMM	Ensemble of Column Hidden Markov Model Classifiers
COR	Correlation Coefficient
DB	Davies-Bouldin Index
DF	Double Fault

DIFF	Difficulty Measure
DM	Disagreement Measure
DSC	Dempster-Shafer Combination
DT	Decision Template
EN	Entropy Measure
EoC	Ensemble of Classifiers
EoHMM	Ensemble of Hidden Markov Model Classifiers
E(2C)	Approximation of Disagreement Measure from Mirkin's Metric based on 2-Clusters Hypothesis
E(MC)	Approximation of Disagreement Measure from Mirkin's Metric based on Multi-Clusters Hypothesis
E(VI)	Approximation of Disagreement Measure from Mirkin's Metric based on the Variation of Information Hypothesis
GA	Genetic Algorithm
GD	Generalized Diversity
HMM	Hidden Markov Model
INT	Interrater Agreement
KNORA	K-Nearest Oracles
KNN	K-Nearest Neighbors
KNP	K-Nearest Prototypes
KW	Kohavi-Wolpert Variance

LCA	Local Class Accuracy
LDC	Normal Densities Based Linear Classifier
MAX	Maximum Rule for classifier combination
MAJ	Majority Voting Rule for classifier combination
MCS	Multiple Classifier System
ME	Mean Error
MiN	Minimum Rule for classifier combination
MLP	Multi-layer Perceptrons
MOGA	Multi-Objective Genetic Algorithm
MSE	Mean Square Error
MSE(2)	Mean Square Error for 2 Classifiers
MSE(L)	Mean Square Error for L Classifiers
MVE	Majority Voting Error
NB	Naive Bayes for classifier combination
NBC	Naive Bayes Classifier
NIST SD	NIST Scientific and Technical Databases
NNC	Neural Network Classifier with Back-propagation
NSGA2	Elitist Non-Dominated Sorting Genetic Algorithm.
OLA	Overall Local Accuracy
PBM	Pakhira-Bandyopadhyay-Maulik Index

PFM	Pairwise Fusion Matrix
PFM-DIV	Pairwise Fusion Matrix weighted by Diversity of Classifier-Pair
PFM-IRR	Pairwise Fusion Matrix weighted by Individual Classifier Recognition Rate
PFM-P	Pairwise Fusion Matrix weighted by Class Probabilities
PFM-MAX	Pairwise Fusion Matrix applying Maximum Rule
PFM-MAJ	Pairwise Fusion Matrix applying Majority Voting Rule
PFM-MIN	Pairwise Fusion Matrix applying Minimum Rule
PFM-SUM	Pairwise Fusion Matrix applying Sum Rule
PFM-PRO	Pairwise Fusion Matrix applying Product Rule
PPFM	Probability-Based Pairwise Fusion Matrix
PRO	Product Rule for classifier combination
PWC	Parzen Windows Classifier
Q	Q-Statistics
QDC	Quadratic Discriminant Classifier
REL	Relative Optima
RBN	Radial Basis Neural Network Classifier
RMSSTD	Root-Mean-Square Standard Deviation Index
ROW-HMM	Ensemble of Row Hidden Markov Model Classifiers
RS	R-Squared Index

SS	Sum of Squares
SUM	Sum Rule for classifier combination
SVM	Support Vector Machine
UCI	UCI Machine Learning Repository (University of California, Irvine)
WER	Wernecke's Method for classifier combination
W-MAJ	Weighted Majority Voting Rule for classifier combination
XB	Xie-Beni Index

LIST OF SYMBOLS

a_i	Recognition rate of classifier $f(i)$
amb_{ij}	Ambiguity between classifier $f(i)$ and classifier $f(j)$
$\bar{b}$	Average bias
b_i	Coefficient of classifier $f(i)$
$\bar{C}$	Cluster center of all data points
$\bar{c}$	Average covariance
c_i	Cluster centroid for cluster c_i
C_{00}	The number of data point pairs that are in different clusters under both clustering C_i and clustering C_k
C_{01}	The number of data point pairs that are in the same cluster under both clustering C_i and clustering C_k
C_{10}	The number of data point pairs that are in the same cluster under clustering C_i, but not clustering C_k
C_{11}	The number of data point pairs that are in the same cluster under clustering C_k, but not clustering C_i
$\|C_i\|$	The number of samples belonging to cluster c_i
$c(i)$	The class label output from classifier $f(i)$
$c(i)_T$	The number of classifiers voting for class i in a T-class problem
D	Data Observation

D_{nc}	Inter-cluster measure
$\bar{d}$	Average diversity
$d(c_i, c_j)$	Dissimilarity between cluster c_i and cluster c_j
d_j	Distance between the test sample and the training sample x_j
d_{ij}	Diversity between classifier $f(i)$ and classifier $f(j)$
$d_{ij,t}$	Distance between cluster c_i and cluster c_j
d_{min}	Minimum Inter-cluster distance
$diam(c_i)$	Diameter of cluster c_i
$div(f(i), f(j))$	Diversity between classifier $f(i)$ and classifier $f(j)$
$\widehat{div_{amb}}$	Compound diversity function for diversity measures that represent ambiguity between classifiers
$\widehat{div_{sim}}$	Compound diversity function for diversity measures that represent similarity between classifiers
$E(y\|x)$	Estimated probability of a pattern x belonging to class y
$F(C_i, C_k)$	Fowlkes-Mallows Index of clustering C_i and clustering C_k
$f(x, D)$	Probability estimated by a classifier f trained with dataset D of a pattern x belonging to an indicated class
f_{ens}	Probability estimated by an ensemble of classifiers trained with dataset D of a pattern x belonging to an indicated class
$f(i)(D_i)$	Classifier $f(i)$ trained with dataset D_i
$g(l\|x)$	Discriminant function for class l on sample x

$J(C_i, C_k)$	Jacard Index of clustering C_i and clustering C_k
$J_m(U, V)$	Sum of the squared error with the partition matrix U and the set of cluster centroids V
$K(C_i, C_k)$	Mirkin's Metric of clustering C_i and clustering C_k
L	The number of classifiers in an ensemble
l	Class label
l_{max}	Class label selected among all classes
M	The total number of classifiers in a pool
M_{ik}	The block of a contingency table with column-i and row-k
$m(T)$	The number of correct classifiers exceeding the threshold of being majority
m_{ik}	The value of a contingency table at the block column-i and row-k
N	The number of samples
$n(c(i), c(j))$	The number of samples on which classifier $f(i)$ votes on class $c(i)$ and classifier $f(j)$ votes on class $c(j)$
n_i	The number of samples in cluster c_i
nc	The number of clusters
O	Complexity
$P(l\|c(i), x)$	Probability of a pattern x belonging to class l when the classifier $f(i)$ votes on class $c(i)$

$P(l|c(i), c(j), x)$ Probability of a pattern x belonging to class l when the classifier $f(i)$ votes on class $c(i)$ and the classifier $f(j)$ votes on class $c(j)$

$P(l|c(1), \cdots, c(i), \cdots, c(L), x)$

Probability of a pattern x belonging to class l when classifier $f(1)$ votes on class $c(1)$, classifier $f(i)$ votes on class $c(i)$, and classifier $f(L)$ votes on class $c(L)$, etc

$P(y|x)$ Probability of a pattern x belonging to class y

p Significance value

p_i Classification accuracy of classifier $f(i)$

$R(C_i, C_k)$ Rand Index of clustering C_i and clustering C_k

$R(f(i))$ Individual classifier recognition rate of classifier $f(i)$

$R_{i,qt}$ Ratio of within-cluster scatter to between-cluster separation

S_i Variance of cluster c_i

SS_b Sum of squares between clusters

SS_t Total Sum of Squares

SS_w Sum of squares within clusters

$\rho(T)$ Threshold of the majority voting in a T-class problem

T The number of classes

t Class label

$u_{i,j}$ Membership value in a partition matrix between cluster c_i and cluster c_j

$\bar{v}$ Average variance

v_i	Cluster centroid of cluster c_i
$W_i(C_i, C_k)$	First Wallace Index of clustering C_i and clustering C_k
$W_k(C_i, C_k)$	Second Wallace Index of clustering C_i and clustering C_k
w_i	Weight of classifier $f(i)$
X	A set of samples; a set of data points
$\bar{X}$	Cluster center; centroid
$\bar{X}_i$	Mean of the cluster c_i
X_j	Sample in cluster c_x
x	Pattern sample; data point
y	Class label
z_i	Cluster centroid of cluster c_i

CHAPTER 1

INTRODUCTION

1.1 Background: Ensemble of Classifiers

Pattern recognition is a task which enables machines to recognize different patterns. In general, patterns with known labels (or classes) are used to train agents called classifiers. Once these classifiers have been trained, they can classify new patterns with unknown labels into certain classes, and thus recognize those patterns. In other words, classifiers are designed to find the relationship between pattern features and pattern labels.

There are various types of classification algorithms for classifiers, such as multi-layer perceptrons (MLP), hidden markov models (HMM), k-nearest neighbors (KNN) and support vector machines (SVM), among others. Due to the complexity of a problem, the feature dimension, the class dimension and the number of training samples available, some classification algorithms might perform better than others. When we consider selecting an adequate classification algorithm for a particular problem, the basic objective is twofold: To enhance accuracy to the fullest extent possible, and to reduce classifier training time as much as possible.

There are several ways to improve the accuracies of these classification algorithms. One is to use more than one classifier to carry out the pattern recognition tasks, and this is called a multiple classifier system (MCS) or an ensemble of classifiers (EoC). An MCS or an EoC aims to enhance recognition rates by employing multiple classifiers, rather than by improving the accuracy of a single classifier. It has been shown theoretically and experimentally that by combining the outputs of multiple classifiers we can achieve a better recognition rate (11; 56; 66; 77; 81; 103; 111).

Nevertheless, to create an EoC, we are still faced with several problems: How can we generate multiple classifiers? Then, once these classifiers have been generated, should we use all of them or should we select a sub-group of them? If we decide to select a sub-group, how do we go about it? Then, once the sub-group has been selected, how can we combine the outputs of these classifiers?

These problems have been investigated in the literature, and we present the state of the art in the next section.

1.2 State-of-the-Art of the Methodology

1.2.1 Ensemble Generation

There are several systematic methods for generating multiple classifiers which are currently popular. The idea is to use different datasets to train classifiers, so that these classifiers will behave differently. This gives us multiple diverse classifiers. We describe some basic ensemble generation methods below.

We can use different examples to train classifiers. Supposing we have a large database, for example, if we randomly select only two-thirds of the data points to train a classifier, very likely each classifier will have diverse training samples and thus behave differently. This ensemble generation method is called Bagging (63).

In order to generate different datasets for multiple classifier training more efficiently, we can also select the training samples in a more systematic manner. For example, we can set a probability for each training sample, and we select only two-thirds of all the samples. If a sample has a higher selection probability, then it is more likely to be selected to train classifiers. However, once we train a classifier, we check whether or not this classifier can correctly classify a particular sample. If a sample is correctly classified, it is assigned a lower selection probability. By contrast, if it is wrongly classified, it is assigned a higher

selection probability. We repeat this process for all the samples, which will have the effect of adjusting the selection probability of each sample. In this way, we can focus on more difficult samples. This ensemble generation method is called Boosting (31; 90).

We can also use all the samples, but only a part of their features, to train classifiers. Supposing that the data have a large feature dimension, we can only use a portion of its features to train classifiers. For example, if all the samples have 20 features, we may use different 5 features to train each classifier. This ensemble generation method is called Random Subspaces (49), and the size of the feature subspace is called its cardinality.

In general, once the classifiers have been generated, we need to collect the best of them in a sub-group. We discuss the process of selection in the next section.

1.2.2 Ensemble Selection

Not all the classifiers generated will be helpful for obtaining the best pattern recognition result. Some might have relatively low accuracy, and others might be identical and thus not very useful. For this reason, we need to select the best classifiers from the pool and form a sub-group of them. This selection process is called ensemble selection, because we select certain classifiers to construct an EoC. In general, we select one ensemble for all test patterns, which is referred to as static ensemble selection.

One way to perform static ensemble selection is to make use of the diversity among classifiers (11; 66; 80; 89). Diversity is important, because if all classifiers are the same, we cannot improve the pattern recognition results by combining them. In other words, they must give quite different outputs. Based on this concept, we can simply define a diversity of classifiers and then evaluate different EoCs by measuring their diversities. Finally, we select the EoC with the best diversity.

Another way to do this is to use the classifier combination results directly (5; 61; 89; 101). We select EoCs, combine their outputs and measure their recognition rates on an

independent validation dataset. If a particular EoC achieves the best recognition rates on this validation dataset, then we suppose that it will also be the best on the test dataset.

It has been demonstrated that the measure of the recognition results of EoCs is more reliable than the measure of their diversity (63; 66; 89). However, the fact that we use the recognition results for ensemble selection means that we must know how to combine classifiers before we select them. The problem is that, in general, we do not know the best way to combine these classifiers. Since classifier combination is not optimized, we doubt that ensemble selection based on one classifier combination method will be optimal.

Another interesting approach is to measure classifier accuracy based on the features of a sample, and select a single classifier with the best accuracy for this sample. This means that each sample can use different classifiers. This approach is known as dynamic classifier selection (12; 15; 14; 28; 44; 65; 107). Moreover, since only one classifier would be used for each sample, there is no need to proceed with classifier combination.

If we perform static ensemble selection, we need to combine the outputs of these classifiers. We present some known methods for classifier combination in the next section.

1.2.3 Classifier Combination

After an EoC has been selected, we need to combine the classifiers in the ensemble, and this process is called classifier combination. Many methods can be used to combine the outputs of classifiers (50; 56; 69; 81; 89; 92; 96; 104; 109; 111), and these are called fusion functions. In general, there are two types of fusion function: one which only requires the crisp class label outputs (for example, this sample belongs to class A, that sample belongs to class B), and the other which requires the probability outputs for each class (for example, this sample has a 90% probability of belonging to class A, and a 10% probability of belonging to class B).

For fusion functions which use the probability outputs for each class, we can simply combine their outputs by summing the probabilities for each class from all classifiers (the SUM rule), or we can combine their outputs by multiplying the probabilities for each class from all classifiers (the PRODUCT rule). We can also simply choose the class label with the maximum probability, either by referring to the maximum probability from all classifiers (the MAX rule) or by referring to the minimum probability from all classifiers (the MIN rule) (50; 56; 69; 81; 89; 92; 96; 104; 109; 111).

For fusion functions which use only the crisp class label outputs, the options are somewhat limited. The simplest way to combine them is to use the majority voting rule: each classifier has a vote on a sample, and the class that obtains the most votes wins (the MAJ rule).

Besides these simple fusion functions, there are a number of trained fusion functions that use another independent database to make up the combination rules, such as the Behavior-Knowledge Space (BKS), the Decision Template (DT), Naive Bayes (NB) (50; 69; 92; 104), etc. These will be discussed later in this thesis, following a short discussion on some of the potential problems and drawbacks of the current methods for ensemble creation, ensemble selection and classifier combination.

1.3 Problem Statement

Although there are a number of useful methods proposed in the literature for ensemble creation, ensemble selection and classifier combination, our understanding of the ensemble remains limited. Below are some of the limitations and potential disadvantages of current methods:

- Ensemble Generation

  In general, ensemble generation methods use a part of data subset to train classifiers, however :

a. The Random Subspaces method requires a minimum number of features, and is therefore only adequate for problems with high feature dimension.

b. If the number of available samples is small, then Bagging or Boosting might encounter "the dimensional curse" for classifier training.

c. The reduction of features or training samples might not be desirable for some complex classification algorithms.

- Ensemble Selection

  In order to select the best ensemble from a classifier pool, different objective functions have been proposed :

  a. The use of diversity for ensemble selection does not perform well.

  b. In order to use a fusion function (such as majority voting error) for ensemble selection, we should first define it, and there is no guarantee that the fusion function chosen will be optimal for the problem at hand.

  c. The ensemble selection process is mainly static; that is, we select one ensemble for all test patterns. Again, this is sub-optimal.

  d. Dynamic classifier selection does not consider the use of the ensemble, which might further boost its performance and stability.

  e. In order to carry out ensemble selection, we need to train classifiers. Since not all the classifiers trained will be used, the time spent for additional classifier training is wasted.

  f. If the size of classifier pool is large, then ensemble selection occurs in a large search space. This is particularly time-consuming.

- Classifier Combination

  Once an ensemble has been selected, we need a fusion function to combine its classifiers :

  a. Most simple fusion functions require the class probability outputs from the classifiers, which are not adequate for classifiers with only class label outputs.

  b. Most trained fusion functions will require a significant number of training samples. This causes problems for small data.

  c. Some trained fusion functions, such as BKS, can be applied only for problems with small class dimensions.

As we can see from the problems described above, there is still much room for improvement and innovation in the field of EoC. The objective of our work is to propose applicable methods with a view resolving, at least partly, some of these problems. We remind readers, however, that EoC optimization is a very complex issue. It is related to a number of varied processes, and our contribution constitutes only part of an improved understanding of the use of EoCs.

1.4 Objectives and Contributions

We propose three new methods for ensemble selection for different contexts, a new ensemble creation scheme for HMMs and a new classifier combination method for classifiers. Our objective is to partly resolve some of the difficulties associated with EoCs presented in the previous section. It is important to mention that we do not assume that these methods are the best choices for all problems, since the best method is usually problem-dependent, given that the most adequate ensemble method often depends on the feature dimension and

the features of the classes and classifiers, on data size, on problem complexity and on the choice of classification algorithm. We offer alternative ways to employ an EoC system, rather than to achieve an optimization of all factors involved in EoC selection, which is nearly impossible. The methods we propose make the following contributions:

- Ensemble Generation:

  We propose an ensemble generation method that does not require using data subset for HMMs :

  a. Ensemble of HMM classifiers based on the Clustering Validity Index.

     Besides the traditional Bagging, Boosting and Random Subspaces ensemble creation methods, we propose a new ensemble creation method for HMMs. In general, HMMs need sufficient samples for training to enable them to perform well. But the fact that these ensemble creation methods use only data subsets could cause problems for HMM training. We thus propose a method for creating an ensemble of HMMs which not only employs all data points and all features, but also offers diversity among classifiers.

- Ensemble Selection:

  We make three major contributions concerning ensemble selection:

  a. Compound Diversity Functions for Ensemble Selection.

     Our first contribution is to combine diversity and classifier accuracy for ensemble selection. This is a more general ensemble selection method, and is not based on any one classifier combination method. We will show that this method has a strong theoretical basis and performs better than the traditional ensemble selection based on diversity among classifiers. Moreover,

since we do not fix any classifier combination method for ensemble selection, it is possible to perform fusion function selection and further optimize EoC performance.

b. From Dynamic Classifier Selection to Dynamic Ensemble Selection.
Our second contribution is to select ensembles of classifiers dynamically. All the methods in the literature are aimed at selecting one EoC for all samples, but, in fact, different samples might need different EoCs so that they can be more adequately classified. Based on this concept, we propose a new dynamic ensemble selection method in our work, and compare it with traditional static ensemble selection and dynamic classifier selection.

c. The Implication of Data Diversity for Classifier-free Ensemble Selection in Random Subspaces.
Our third contribution is to select EoCs without using any classifiers. This classifier-free method is only for use with the Random subspaces ensemble generation method. Remember that different classifiers are generated with all samples, but only a part of the features is used in the Random Subspaces method. Since we generate different classifiers based on different feature subsets, then, if we can select adequate feature subsets, we are actually selecting adequate classifiers. We thus propose a method for feature subset selection on Random Subspaces, which will also constitute a classifier-free ensemble selection method. With this approach, we can reduce the time spent in useless classifier training and also reduce the ensemble selection search space.

- Classifier Combination:
We also propose a transformation matrix that is applicable for all kinds of fusion functions :

a. Pairwise Fusion Matrix for Combining Classifiers.

   As we mentioned above, there are very few fusion functions for the crisp class label outputs. We thus present a new fusion function that can transform crisp class label outputs into class probability outputs. Once we have obtained the class probability outputs, we can apply many more fusion functions to combine classifiers. This method is thus applicable for all kinds of fusion functions. Furthermore, this method requires many fewer training samples and can be applicable for problems with high dimensional class as well.

The proposed methods are all strongly related. They represent solutions for different types of problems, but they are not necessarily mutually exclusive. For example, dynamic ensemble selection can be applied with the pairwise fusion matrix. Likewise, compound diversity functions can be used on ensembles of HMM classifiers.

1.5 Organization of the Thesis

This thesis is organized as follows:

a. Compound Diversity Functions for Ensemble Selection

   A new ensemble selection scheme is presented in chapter 2. It has been submitted to the International Journal of Pattern Recognition and Artificial Intelligence, and was presented at the International Joint Conference on Neural Networks (IJCNN 2006), along with experiments measuring the correlations between CDF and ensemble accuracy, and at the International Conference on Pattern Recognition (ICPR 2006), along with experiments focusing on ensemble performance comparison. In this work, we propose combining diversity and classifier accuracy for ensemble selection.

b. Pairwise Fusion Matrix for Combining Classifiers

   We introduce a new approach for combining classifiers chapter 3. It has been ac-

cepted by Pattern Recognition, vol. 40, 2007, and was presented at the International Workshop on Multiple Classifier Systems (MCS 2007). We present here a transformation method that is applicable on all kinds of fusion functions to combine classifiers. Since PFM and CDF are very general, widely applicable and mutually compatible, CDF has also been tested in some PFM experiments. Their combination was presented at the Genetic and Evolutionary Computation Conference (GECCO 2006).

c. Ensemble of HMM classifiers based on the Clustering Validity Index
A new ensemble of HMM creation methods is introduced in chapter 4. It has been submitted to the International Journal of Pattern Analysis and Application and is currently under revision. It was also presented at the International Workshop on Multiple Classifier Systems (MCS 2007). In this work, we present a new ensemble of HMM classifier creation method based on various codebook sizes. We will create ensemble of HMM classifiers, perform ensemble selection with CDF and classifier combination with PDF, and compare the results with traditional techniques.

d. From Dynamic Classifier Selection to Dynamic Ensemble Selection
We present a new dynamic ensemble selection method, K-Nearest Oracles KNORA) in chapter 5. The paper has been submitted to Pattern Recognition, and it was also presented at the International Workshop on Multiple Classifier Systems (MCS 2007). In this work, we present an innovative dynamic ensemble selection method as an alternative to static ensemble selection. The combination with PDF is compared with traditional static ensemble selection methods and with dynamic classifier selection schemes.

e. The Implication of Data Diversity for a Classifier-free Ensemble Selection in Random Subspaces
The classifier-free ensemble selection method is presented in chapter 6. It has been

submitted to the IEEE Transactions on Pattern Analysis and Machine Intelligence. This is a special ensemble selection method to be used only for the Random Subspaces ensemble creation method. Note that this classifier-free ensemble selection method is not applicable on our HMM handwritten numeral recognition system. Our purpose in presenting this work here is to demonstrate the possibility of performing "data selection", which has never been mentioned in the literature, but will be of great interest to develop in the future. Its combination with PDF is investigated and compared with other ensemble selection techniques.

Most of our topics are strongly interrelated. Several of them can be applied together, and others can serve as post-processing methods. Below is a global view of the organization of our work (Fig. 1):

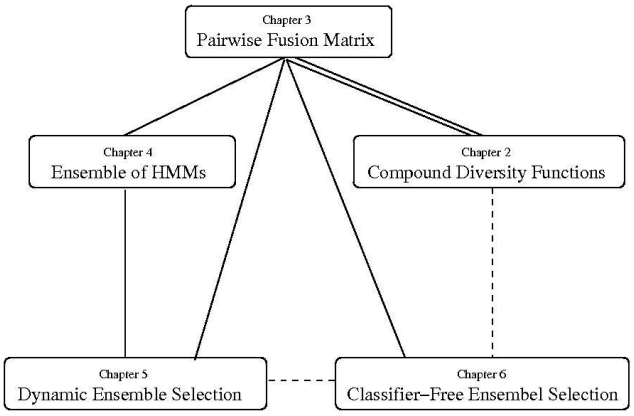

Figure 1 The map of relationship between the proposed methods. The solid lines indicate that the methods are compatible and can be used together, and the dash lines means that the application as post-processing is possible. The double line between CDF and PFM indicates that both are pairwise based

Those interested in the whole EoC system can begin at chapter 2 and read through to the end of chapter 4. These chapters address the ensemble creation, ensemble selection and classifier combination processes, and thus offer a global view of an EoC. Note that chapter

3 discusses some techniques described in chapter 2, and chapter 4 requires reading parts of chapter 2 and chapter 3.

Those already familiar with EoCs may read chapter 2, chapter 3 and chapter 5, which offer quite different and innovative approaches to ensemble selection and classifier combination. The material in chapter 5 is independent of that in both chapter 2 and chapter 3. Consequently, readers interested only in dynamic selection can go to chapter 5 directly.

Chapter 6 is geared to advanced readers who not only understand EoC, but also the Multi-Objective Genetic Algorithm (MOGA). Those who have no background knowledge, but are interested, might find it helpful to read K. Debs Multi-Objective Optimization using Evolutionary Algorithms (13), because some techniques applied in our work have been represented in this book.

Finally, we remind readers that chapter 4 describes a special ensemble method for HMMs, and so to fully appreciate this material, it is important to have some basic knowledge of HMMs. For those who are interested in the topic, we recommend L. R. Rabiners work, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition" (83). For those who are interested in our baseline HMM system, Alceu de Souza Britto Jr.s thesis, "A Two-Stage HMM-Based Method for Recognizing Handwritten Numeral Strings" (8), might be helpful. Note that the framework of chapter 4 is based on his work.

CHAPTER 2

COMPOUND DIVERSITY FUNCTIONS FOR ENSEMBLE SELECTION

An effective way to improve a classification method's performance is to create ensembles of classifiers. Two elements are believed to be important in constructing an ensemble: a) the performance of each individual classifier; and b) diversity among the classifiers. Nevertheless, most works based on diversity suggest that there exists only weak correlation between classifier performance and ensemble accuracy. We propose compound diversity functions which combine the diversities with the performance of each individual classifier, and show that there is a strong correlation between the proposed functions and ensemble accuracy. Calculation of the correlations with different ensemble creation methods, different problems and different classification algorithms on 0.624 million ensembles suggests that most compound diversity functions are better than traditional diversity measures. The population-based Genetic Algorithm was used to search for the best ensembles on a handwritten numerals recognition problem and to evaluate 42.24 million ensembles. The statistical results indicate that compound diversity functions perform better than traditional diversity measures, and are helpful in selecting the best ensembles.

2.1 Introduction

The purpose of pattern recognition systems is to achieve the best possible classification performance. A number of classifiers are tested in these systems, and the most appropriate one is chosen for the problem at hand. Different classifiers usually make different errors on different samples, and this means that by combining classifiers, we can arrive at an ensemble that makes more accurate decisions (11; 56; 66; 77; 81; 103; 111). In order to have classifiers with different errors, it is advisable to create diverse classifiers. For this purpose, diverse classifiers are grouped together into what is known as an Ensemble of Classifiers (EoC). There are several methods for creating diverse classifiers,

among them Random Subspaces (49), Bagging and Boosting (31; 63; 90). The Random Subspaces method creates various classifiers by using different subsets of features to train them. Because problems are represented in different subspaces, different classifiers develop different borders for the classification. Bagging generates diverse classifiers by randomly selecting subsets of samples to train classifiers. Intuitively, based on different sample subsets, classifiers would exhibit different behaviors (See appendix 1). 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. With this mechanism, most created classifiers will focus on hard samples and can be more effective.

There are two levels of problems in optimizing the performance of an EoC. First, how are classifiers selected, given a pool of different classifiers, to construct the best ensemble? Second, given all the selected classifiers, what is the best rule for combining their outputs? These two problems are fundamentally different, and should be solved separately to reduce the complexity of optimization of EoCs; the former focuses on ensemble selection (5; 11; 61; 66; 89; 80; 101) and the latter on ensemble combination, i.e. the choice of fusion functions (56; 81; 89; 96; 111). For ensemble selection, the problem can be considered in two steps: (a) find a pertinent objective function for selecting the classifiers; and (b) use a pertinent searching algorithm to apply this criterion. Obviously, a correct criterion is one of the most crucial elements in selecting pertinent classifiers (11; 66; 80; 89). It is considered that, in a good ensemble, each classifier is required to have different errors, so that they will be corrected by the opinions of the whole group (56; 63; 66; 88; 89). This property is regarded as the diversity of an ensemble.

Diversity is important for ensemble selection and cannot be substituted by fusion functions. There are several reasons for this: First, for a large number of classifiers, fusion functions need to take into account all classifier outputs for each evaluation (5), whereas pairwise diversity measures can be calculated beforehand, and evaluating them is less

time-consuming and more effective. Second, classifiers can be created and ensembles can be trained along with diversity (30; 73). Third, we need to optimize fusion functions in order to combine classifiers (56), since, without knowing the best fusion functions, it would be premature to use them for ensemble selection. Given that different fusion functions need to be evaluated, any pre-selected fusion function might not be optimal for the ensemble selection. According to the 'no free lunch' theorem (105; 106), it is understandable that a search algorithm based on one fusion function might not be better than another search algorithm based on a more common objective function. Based on these arguments, we consider ensemble selection and ensemble combination as two different problems, each of which should be solved separately.

Nevertheless, there is no universal definition of diversity, and therefore a number of different diversity measures have been proposed (1; 25; 29; 47; 49; 61; 66; 80; 101). What is more, it has been observed that, even with so many different diversity measures, clear correlations between ensemble accuracy and diversity measures cannot be found (11; 63; 66), leading some researchers to consider diversity measures to be unnecessary for ensemble selection (89). To summarize, the concept of diversity does help, but both theoretical and experimental approaches showing that strong correlations between diversity measures and ensemble accuracy are lacking. Given the challenge of using diversity for ensemble selection, we argue that the lack of correlation between ensemble accuracy and diversity does not imply that there is no direct relationship between them, but that diversity should be taken into account with the performance of individual classifiers. We suggest that such compound diversity functions can give the best correlation with ensemble accuracy. Here are the key questions that need to be addressed:

a. Diversity is important, but it has only a weak correlation with the ensemble accuracy. Can we combine the diversity with the classifier accuracies to achieve a higher correlation with the ensemble accuracy?

b. Is there any effect on such a correlation, e.g. from the number of classes or the number of classifiers?

c. Can the diversity combined with the classifier accuracy be effective for ensemble selection?

To answer these questions, we derive compound diversity functions by combining diversities and the performances of individual classifiers, and we show that with such functions there are strong correlations between the diversity measures and ensemble accuracy. Furthermore, we demonstrate the impact on the correlation between the accuracy and the diversity with different ensemble creation methods, with different number of classifiers and with different number of classes. However, the problem of EoC optimization is very complex. In addition to diversity issues, it is also related to fusion functions for classifier combination and to searching algorithms for ensemble selection. The contribution of this chapter constitutes only part of an improved understanding of the use of diversity for ensemble selection.

The chapter is organized as follows. In the next section, we investigate the dilemma of the lack of correlation between diversity and ensemble accuracy. In section 3, we give the reason that why the compound diversity functions might work. In section 4, we discuss how the number of classifiers and the number of classes might influence the correlation between ensemble accuracy and compound diversity functions. Section 6 presents basic diversity measures that would be tested in the experiments. Correlations with ensemble accuracy are measured on 0.624 million ensembles in section 6. In section 7, we use the proposed compound functions as objective functions for ensemble selection among 42.24 million ensembles. A discussion and our conclusion are contained in the final sections.

2.2 Dilemma of the Ambiguity towards the Ensemble Accuracy

In this section, we adopt the framework established in (11) to discuss the impediment to using the ambiguity to estimate ensemble accuracy. For readers not familiar with the work in (11; 62), we present a short introduction here, but the original papers offer far more details. The main point is to decompose the mean square error of an ensemble into an ambiguity part and a non-ambiguity part, and we can find the variance terms in both the ambiguity part and the non-ambiguity part. As a result, when we try to maximize the ambiguity among classifiers, we will also affect the non-ambiguity part. That is the reason that an increase in the diversity will not necessarily guarantee a decrease in the global ensemble error.

To start, we need to introduce the concept of the bias-variance decomposition (10; 11; 18; 27; 53). Briefly speaking, attempts to reduce the bias component will cause an increase in variance, and vice versa.

Suppose that the response variable is binary, i.e., $y \in \{0, 1\}$, the probability of a sample x belonging to a class y can be $P(y|x)$, and the classification task is to estimate this probability $E\{y|x\} = P(y|x)$ based on a sequence of the N observation $D = \{(x_1, y_1), (x_2, y_2), \cdots (x_N, y_N)\}$. Assume that we have a classifier f trained with a particular dataset D, the probability of a data point x belonging to a class predicted by the classifier f can be written as $f(x, D)$. To measure the effectiveness of the $f(x, D)$ as a predictor of the $E\{y|x\}$, we can simply calculate its mean square error (MSE) (62):

$$
\begin{aligned}
& E\{(f(x, D) - E\{y|x\})^2\} \\
= & (E\{f(x, D)\} - E\{y|x\})^2 + E\{(f(x, D) - E\{f(x, D)\})^2\} && (2.1) \\
& or\ MSE\{f\} = bias(f)^2 + var(f) && (2.2)
\end{aligned}
$$

where $E\{f(x, D)\}$ is the expectation of the classifier $f(x, D)$ with the respect to the training set D, i.e., the average over the ensemble of the possible D. We can deduct that:

$$bias(f) = E\{f(x, D)\} - E\{y|x\} \tag{2.3}$$

$$var(f) = E\{(f(x, D) - E\{f(x, D)\})^2\} \tag{2.4}$$

This form can be further decomposed into bias-variance-covariance (11; 101). For an ensemble with L classifiers, the averaged bias of the ensemble members is defined as :

$$\bar{b} = \frac{1}{L}\sum_{i}^{L}(E\{f_i(x, D_i)\} - E\{y|x\}) \tag{2.5}$$

where D_i is the dataset used to train the classifier f_i. We note that $E\{f_i(x, D_i)\}$ is the average over the ensemble of the possible D, and thus all classifiers will have the same $E\{f(x, D)\}$. We just keep the notation for the clarity and for the consistency with (11). Then, the averaged variance of the ensemble members will be :

$$\bar{v} = \frac{1}{L}\sum_{i}^{L}(E\{(f_i(x, D_i) - E\{f_i(x, D_i)\})^2\}) \tag{2.6}$$

and the averaged covariance of the ensemble members will be:

$$\bar{c} = \frac{1}{L(L-1)}\sum_{i}^{L}\sum_{j\neq i}^{L} E\{(f_i(x, D_i) - E\{f_i(x, D_i)\})(f_j(x, D_j) - E\{f_j(x, D_j)\})\} \tag{2.7}$$

If we decompose the mean square error for this ensemble of L classifiers, we get :

$$MSE(L) = E\{((\frac{1}{L}\sum_{i}^{L} f_i(x, D_i)) - E\{y|x\})^2\} \tag{2.8}$$

$$= \bar{b}^2 + \frac{1}{L}\bar{v} + \frac{L-1}{L}\bar{c} \tag{2.9}$$

To determine the link between $MSE(L)$ and the ambiguity, which measures the amount of variability among classifier outputs in ensembles, we need to apply ambiguity decomposition. It has been proved (62) that, at a single data point, the quadratic error of the ensemble f_{ens} is guaranteed to be less than or equal to average quadratic error of the individual classifiers (62):

$$(f_{ens} - E\{y|x\})^2 = \sum_i^L w_i(f_i(x, D_i) - E\{y|x\})^2 - \sum_i^L w_i(f_i(x, D_i) - f_{ens})^2 \quad (2.10)$$

where w_i is the weight of classifier $f_i(x, D_i)$ in the ensemble, and $0 \leq w_i \leq 1$. If every classifier $f_i(x, D_i)$ has the same output, then the second term is 0, and f_{ens} would be equal to the average quadratic error of the individual classifiers. Note that the ensemble function is a convex combination ($\sum_i^L w_i = 1$):

$$f_{ens} = \sum_i^L w_i f_i(x, D_i) \quad (2.11)$$

For the $MSE(L)$ of this ensemble of classifiers, suppose that every classifier has the same weight, i.e. $\forall i, w_i = \frac{1}{L}$, so f_{ens} is merely the average function of all individual classifiers $f_{ens} = \bar{f}$. Consequently the ambiguity decomposition can be written as :

$$(\bar{f} - E\{y|x\})^2 = \frac{1}{L}\sum_i^L (f_i(x, D_i) - E\{y|x\})^2 - \frac{1}{L}\sum_i^L (f_i(x, D_i) - \bar{f})^2 \quad (2.12)$$

Note that its expectation is exactly eq. 2.8 and eq. 2.9

$$E\{\frac{1}{L}\sum_i^L (f_i(x, D_i) - E\{y|x\})^2 - \frac{1}{L}\sum_i^L (f_i(x, D_i) - \bar{f})^2\} = \bar{b}^2 + \frac{1}{L}\bar{v} + \frac{L-1}{L}\bar{c} \quad (2.13)$$

The ambiguity is the second term on the left-hand side in eq. 2.13, and it can be written as (62):

$$E\{(\frac{1}{L}\sum_{i}^{L}(f_i(x,D_i)-\bar{f})^2)\}$$

$$=\frac{1}{L}\sum_{i}^{L}E\{(f_i(x,D_i)-E\{f_i(x,D_i)\})^2\}-E\{(\bar{f}-E(\bar{f}))^2\} \tag{2.14}$$

$$=\bar{v}-var(\bar{f})=\bar{v}-\frac{1}{L}\bar{v}-\frac{L-1}{L}\bar{c} \tag{2.15}$$

The first term of the left-side in eq. 2.13 is the sum of averaged bias and averaged variance of classifiers:

$$E\{\frac{1}{L}\sum_{i}^{L}(f_i(x,D_i)-E\{y|x\})^2\}=\bar{b}^2+\bar{v} \tag{2.16}$$

As stated in (11), the term $\bar{v}$, the average variance, exists in both the ambiguity part and the non-ambiguity part of $MSE(L)$. This means that we cannot simply maximize the ambiguity without affecting the bias component of $MSE(L)$. When we try to maximize the ambiguity among classifiers, we actually maximize the difference between its variance $\bar{v}$ and its covariance $\bar{c}$. If the term $\bar{v}$ increases, the non-ambiguity part of $MSE(L)$ will increase too. This is why, in general, an increase in the diversity measure will not necessarily guarantee a decrease in the global ensemble error. We need to mention that the above discussion is with respect to a single data point, but the results can generalize to the full space (11).

2.3 Proposed Compound Diversity Functions

The above section shows that the $MSE(L)$ can be decomposed into an ambiguity part and a non-ambiguity part, and because the variance terms exist in both parts, there is no

easy solution to minimize the $MSE(L)$ by simply maximizing the ambiguity. In this section, however, we will show that in some certain circumstances the $MSE(L)$ can have another form of the decomposition. Based on this decomposition, we propose an indirect approximation of the $MSE(L)$ with only the average errors of individual classifiers and the diversities of classifier-pairs. The proposed approximation might thus help reduce the $MSE(L)$ for the ensemble selection. First, suppose that we have an ensemble with only 2 classifiers $f_i(D_i), f_j(D_j)$, and that classifiers $f_i(D_i)$ and $f_j(D_j)$ have the recognition rates a_i and a_j on a data set X, respectively, and the average error of classifier $f_i(D_i)$ is $(1 - a_i)$, and the average error of classifier $f_j(D_j)$ is $(1 - a_j)$ and the diversity d_{ij} is measured between them. With only two classifiers, we get $L = 2$ in eq. 2.6 and eq. 2.7. As a result, on any data point $x \in X$, the ambiguity between $f_i(x, D_i)$ and $f_j(x, D_j)$ is exactly half of the difference between their variance and covariance in eq. 2.15:

$$\begin{aligned} amb_{ij} &= \frac{1}{2}(\bar{v} - \bar{c}) \\ &= \frac{1}{4}(E\{(f_i(x, D_i) - E\{f_i(x, D_i)\})^2\} + E\{(f_j(x, D_j) - E\{f_j(x, D_j)\})^2\} \\ &\quad -2 \cdot E\{(f_i(x, D_i) - E\{f_i(x, D_i)\}) \cdot (f_j(x, D_j) - E\{f_j(x, D_j)\})\}) \end{aligned} \tag{2.17}$$

If we use $L = 2$ in eq. 2.9 and replace $\frac{1}{2}(\bar{v} - \bar{c})$ by amb_{ij}, we can write $MSE(2)$ as :

$$MSE(2) = \bar{b}^2 + \frac{1}{2}(\bar{v} + \bar{c}) = amb_{ij} + \bar{b}^2 + \bar{c} \tag{2.18}$$

As a result of this decomposition, there are basically two $MSE(2)$ terms, the first being the ambiguity of the ensemble, and the second being the sum of the averaged covariance and the averaged bias of individual classifiers. Using the eq. 2.17, we can write the above equation as :

$$MSE(2) = \bar{b}^2 + \bar{v} - \frac{1}{2}(\bar{v} - \bar{c}) = \bar{b}^2 + \bar{v} - amb_{ij} \tag{2.19}$$

where $amb_{ij} = \frac{1}{2}(\bar{v} - \bar{c})$. The point is that we have the term $\bar{b}^2 + \bar{v}$ instead of $\bar{b}^2 + \bar{c}$, and one way to approximate the $\bar{b}^2 + \bar{v}$ of the ensemble is through the $var(f) + bias(f)^2$ of each individual classifier f, which is exactly the MSE of each individual classifier. Despite this, we do not have its exact value of the $var(f) + bias(f)^2$ of the classifier f on each data point. However, we have the average of its zero-one loss error (18) on the whole data set X, i.e. $(1 - a_i)$. The behavior of a zero-one loss error is much more complicated, and up to now there has simply been no clear analog of the bias-variance-covariance decomposition when we have a zero-one loss function (11; 18). Nevertheless, it is still reasonable to assume that the larger the MSE of a classifier on each data point x, the larger its average zero-one loss error on the whole data set X should be. We need to draw some assumptions to get the reasonable approximation here. First, we want to approximate the value of $\bar{b}^2 + \bar{v}$ in the eq. 2.18, but what we know is the average error rate $(1 - a_i)$ of any given classifier f_i. So suppose that :

a. For any classifier f_i, $(1 - a_i) \approx \alpha_i(var(f_i) + bias(f_i)^2)$.

b. All classifiers in the ensemble have similar $MSE(f)$.

The first assumption gives that $(1 - a_i) \approx \alpha_i(var(f_i) + bias(f_i)^2)$ for f_i and $(1 - a_j) \approx \alpha_j(var(f_j) + bias(f_j)^2)$ for f_j. Still, owing to the lack of exact values for α_i and α_j, there is no easy solution to the approximation of the sum of averaged bias and averaged variance. But, if the second assumption stands, i.e., these individual classifiers have a similar $MSE(f)$, and one could obtain a reasonable approximation of $(\bar{b}^2 + \bar{v})$ by calculating the geometric mean of individual classifier's $(var(f) + bias(f)^2)$. As a result, the term $\bar{b}^2 + \bar{v}$ might be approximated by the error rates of individual classifiers based on the above assumptions :

$$(\bar{b}^2 + \bar{v}) \approx \gamma((1 - a_i) \cdot (1 - a_j))^{\frac{1}{2}} \tag{2.20}$$

Now, we want to approximate the value of the ambiguity amb_{ij} in the eq. 2.18 with the diversity measures. Again, we need to suppose that :

- The diversity measures represent approximations of the ambiguity among classifiers, i.e., $d_{ij} \propto amb_{ij}$, $0 \leq d_{ij} \leq 1$.

Using the assumption, the term d_{ij} has a high correlation with $amb_{ij} = \frac{1}{2}(\bar{v} - \bar{c})$, and the approximation of $\frac{1}{2}(\bar{v} - \bar{c})$ can be written as :

$$amb_{ij} \approx \delta \cdot d_{ij} \tag{2.21}$$

For an approximation to $MSE(2)$, i.e. $\bar{b}^2 + \bar{v} - amb_{ij}$, given the approximation $(\bar{b}^2 + \bar{v})$ as $\gamma \cdot ((1 - a_i) \cdot (1 - a_j))^{\frac{1}{2}}$, and the approximation of their diversity $(\bar{v} - \bar{c})$ as $\delta \cdot d_{ij}$, we could not achieve any exact solution due to the lack of values γ and δ. Again, we need to make some assumptions to go further :

- The ambiguity term and the non-ambiguity term have similar weights in $MSE(2)$.

Based on this assumption, the value $MSE(2)$ can be approximated as the product of the error rates of each classifier and their pairwise diversity. Given $0 \leq d_{ij} \leq 1$, we have $0 \leq 1 - d_{ij} \leq 1$, and we define an index for the approximation of $MSE(2)$ as :

$$\widetilde{MSE}_{ij} \equiv (1 - d_{ij}) \cdot ((1 - a_i) \cdot (1 - a_j))^{\frac{1}{2}} \tag{2.22}$$

For multiple classifiers, the direct approximation of $MSE(L)$ is much more complex and its term of covariance cannot easily be substituted. Still, we can regard multiple classifiers as a network of classifier-pairs, and we might use the average error of each individual classifier and the diversity between each classifier-pair for an indirect approximation of $MSE(L)$. Given the number of selected classifiers $L \geq 2$, and we have $\widetilde{MSE(L)} \approx$

$(\prod_{i=1}^{L}(1-a_i))^{\frac{1}{L}}(\prod_{i,j=1,i\neq j}^{L}(1-d_{i,j}))^{\frac{1}{L\times(L-1)}}$. By calculating their product, we can get an approximation of ensemble accuracy without any consideration for the type of fusion functions. It is important to note that different diversity measures are supposed to have different sorts of relationships with ensemble accuracy. Some diversity measures measure the ambiguity among classifiers, where positive correlation with ensemble accuracy is expected; others actually measure the similarity among classifiers, where there would be a negative correlation between them and ensemble accuracy. In the case where the diversity measures represent the ambiguity, we combine the diversity measures with the error rates of each individual classifier :

$$\widehat{div_{amb}} = (\prod_{i=1}^{L}(1-a_i))^{\frac{1}{L}}(\prod_{i,j=1,i\neq j}^{L}(1-d_{i,j}))^{\frac{1}{L\times(L-1)}} \tag{2.23}$$

where a_i is the correct classification rate of classifier f_i, and $d_{i,j}$ is the measured diversity between classifier f_i and classifier f_j. Apparently we have $\frac{L\times(L-1)}{2}$ diversity measures on different classifier-pairs. Here, $1-a_i$ is the error rate of classifier-i, and $(1-d_{i,j})$ can be interpreted as the similarity between classifier f_i and classifier f_j. Thus, $\widehat{div_{amb}}$ is, in fact, an estimation of the likelihood of a common error being made by all classifiers. In other word, we expect $\widehat{div_{amb}}$ to have negative correlation with ensemble accuracy. However, if the diversity measures represent the similarity, the proposed compound diversity function should be :

$$\widehat{div_{sim}} = (\prod_{i=1}^{L}(1-a_i))^{\frac{1}{L}}(\prod_{i,j=1,i\neq j}^{L}d_{i,j})^{\frac{1}{L\times(L-1)}} \tag{2.24}$$

where $d_{i,j}$ should be interpreted as the similarity between f_i and f_j in this case. So, $\widehat{div_{sim}}$ ought to mean the likelihood of a common error being by all the classifiers. We expect negative correlation between the $\widehat{div_{sim}}$ and ensemble accuracy. While it is true that these approximations lead to strong correlations with $MSE(L)$ for a fixed number of classifiers L, the bottom line is that the ensemble selection will result in the minimization of L for the

proposed compound diversity function, if L is set as a free parameter. This is substantiated below: Suppose that there are a total of M classifiers in the pool, and we intend to select a subset of L classifiers, $L \leq M$, which can construct an EoC with the best accuracy by a simple majority voting rule (88; 89; 92). For the pairwise diversity measures, suppose that for all classifiers $f_1 \sim f_M$, we measure the diversity d_{ij} on $\frac{M(M-1)}{2}$ classifier-pairs $c_{ij}, 1 \leq i, j \leq M, i \neq j$. Intuitively, there exists at least one classifier-pair $\widehat{c_{ij}}$ with the maximum pairwise diversity $\widehat{d_{ij}}$ that is larger than or equal to any pairwise diversity of other classifier-pairs d_{ij}, for $1 \leq i, j \leq M, i \neq j$. As a consequence, the maximum pairwise diversity $\widehat{d_{ij}}$ of classifier-pair $\widehat{c_{ij}}$ is larger than the diversities of any other selected L classifiers, given that $2 \leq L \leq M$:

$$\forall L, \ \ \widehat{d_{ij}} \geq E\{d_{ij}\} = d_L \tag{2.25}$$

where $E\{d_{ij}\}$ is the mean of the pairwise diversities of L selected classifiers. This means that if we use pairwise diversity as an objective function for ensemble selection, and if the number of classifiers is set as a free parameter, it's quite possible that we will get only one classifier-pair. The proposed compound functions are based on diversity measured in a pairwise manner, even taking into account the individual classifiers' error rates, ensembles with fewer classifiers are more likely to be favored in the ensemble selection. With regard to this effect, functions with various number of classifiers shall be rescaled by [1]:

$$\widehat{div_{amb}} = \frac{L}{L-1}(\prod_{i=1}^{L}(1-a_i))^{\frac{1}{L}}(\prod_{i,j=1,i\neq j}^{L}(1-d_{i,j}))^{\frac{1}{L\times(L-1)}} \tag{2.26}$$

$$\widehat{div_{sim}} = \frac{L}{L-1}(\prod_{i=1}^{L}(1-a_i))^{\frac{1}{L}}(\prod_{i,j=1,i\neq j}^{L}d_{i,j})^{\frac{1}{L\times(L-1)}} \tag{2.27}$$

[1] In practice, when L is large, it is possible that we need to multiply a coefficient η on the compound diversity functions, so that the lower bound of evaluated compound diversity values will not exceed machine capacity and precision.

2.4 Concern about the Number of Classes and the Number of Classifiers

The measures $\widehat{div_{sim}}$ and $\widehat{div_{amb}}$ are supposed to have a strong correlation with the MSE of the ensemble, but this MSE never reaches 100% correlation with ensemble error, for several reasons: First, the ensemble error is a zero-one loss error, while the MSE of the ensemble is based on bias, variance and covariance terms. Second, ensemble error is influenced by the way classifiers are combined, i.e. by the choice of fusion functions, while the MSE of the ensemble does not take fusion functions into consideration when combining ensembles. Third, ensemble error is involved in more complicated situations and is related to other concerns, such as the number of classes and the number of classifiers (see the following discussion). For these reasons then, it is not hard to see why $\widehat{div_{sim}}$ and $\widehat{div_{amb}}$ will not be perfectly correlated with the ensemble error. However, we need to know more about what its limitations are.

Given the complexity of the problem of ensemble selection, and the various *ad hoc* methods for combining classifiers, it is impossible at this stage to create a flawless and complete framework for understanding the limitations of the estimation of ensemble accuracy with compound diversity functions. With this in mind, we set up some preconditions for a special case study as the first step towards gaining these understandings. We suppose that each classifier produces labels of samples as outputs, and we need to fix a fusion function for combining classifiers in an ensemble in our case study. A number of different fusion functions can be used (56), but for, simplicity and effectiveness (89), suppose that a simple majority voting rule (88; 89; 92) constitutes the fusion function of ensemble outputs. Based on these conditions, we wish to know whether or not:

a. Given an ensemble of classifiers, is it possible that some classifiers make more (or less) error without changing the ensemble outputs?

b. Given an ensemble of classifiers, is it possible that some classifier-pairs have greater (or less) diversity without changing the ensemble outputs?

c. If the above two concerns are true, how different can they be while maintaining the same ensemble outputs?

It is not hard to answer the first two questions. When a simple majority voting rule is used, a correct ensemble output depends on the proportion of classifiers correctly classifying this sample. For a sample x in a T-class problem, suppose that the correct class is $i, 1 \leq i \leq T$. The ensemble will give correct output only under the condition $\forall j,\ c(i)_T > c(j)_T$, for $1 \leq i, j \leq T, i \neq j$, where $c(i)_T$ is the number of classifiers making a decision on class i, and $c(j)_T$ is the number of classifiers making a wrong decision on another class j, in a T-class problem. Under the condition $\forall j, c(i)_T > c(j)_T$, the $c(i)_T$ can decrease, and the $c(j)_T$ can increase, and the ensemble can still give the correct output.

A similar reasoning can apply to diversities, because the change in the error rates of each individual classifier will eventually affect the diversities among them. It is apparent that the different error rates of individual classifiers and the different diversities among them can achieve the same ensemble outputs by a simple majority voting rule. We know that there is an unavoidable systematic estimation bias on the correlation measurement with ensemble accuracy for this fusion function. In fact, since this problem results from classifiers combining by a simple majority voting rule, and not from a particular ensemble selection criterion, the effect will occur for any objective functions on ensemble selection.

The third question depends on the nature of the pattern recognition problems and cannot be easily estimated. It is impossible to say in what way this estimation bias will affect the correlation between compound diversity functions and ensemble accuracy. But among those problems are two elements resulting in this estimation bias on correlation measurements between $\widehat{div_{sim}}$ / $\widehat{div_{amb}}$ and ensemble accuracy:

a. the number of classes of the problem

b. the number of classifiers selected from the pool to construct the ensemble

As we mentioned before, an ensemble can maintain the same outputs under the condition that $\forall j, c(i)_T \geq c(j)_T$. For a given sample in a T-class problem, suppose that the ensemble output remains the same. We define a margin $m(T), m(T) \geq 0$ to be the number of correct classifiers exceeding the threshold of being majority (31; 77; 90):

$$m(T) = c(i)_T - \rho(T) \tag{2.28}$$

where $\rho(T)$ is the threshold of the majority voting in a T-class problem. Usually $\rho(T)$ represents the second most popular vote (31):

$$\rho(T) = \max c(j)_T, 1 \leq j \leq T, j \neq i \tag{2.29}$$

Intuitively, given that the output of the ensemble remains unchanged, we still have :

$$c(i)_T \geq \rho(T), 1 \leq i \leq T \tag{2.30}$$

Given that all classifiers have choices on T classes, we can expect both $c(i)_T$ and $\rho(T)$ to decrease when T increases. The larger the number of classes is, the fewer votes are obtained for each class. We describe the details in the appendix 2 for interested readers.

As we can see in Fig. 2, for a 10-class problem, class i received the majority vote, but the margin $m(10)$ with the second most popular voted class j is very small. This means that the ensemble can change its decision with several different votes, therefore the measured error rates and diversities are more accurate in estimating ensemble accuracy. By contrast, for a 3-class problem, the margin $m(3)$ between $c(i)_3$ and $c(j)_3$ is huge, which means that more classifiers are allowed to change their individual outputs while the ensemble can

still maintain the same outputs. In this case, the estimation will be much worse and the correlation with ensemble accuracy will have deteriorated. The margin $m(T)$ is propor-

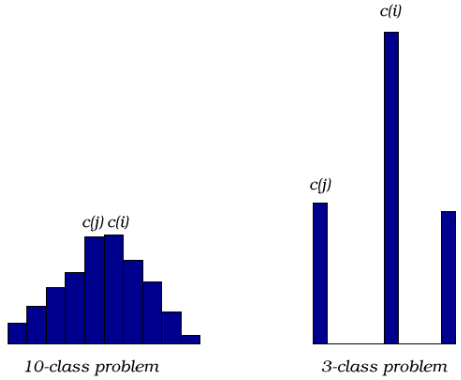

Figure 2 Distribution of 100 votes in ensembles: (a) 10-class problem; (b) 3-class problem

tional to this estimation bias. From the eq. 2.5 in the appendix 2, we note that it is also proportional to the number of classifiers of ensemble L. This indicates that the estimation bias in the correlation measurement between ensemble accuracy and $\widehat{div_{sim}}$ / $\widehat{div_{amb}}$ will become larger when more classifiers are used. This estimation bias results directly from the nature of a zero-one loss error, and from the simple majority voting rule for combining classifiers. No matter which objective function for ensemble selection is used, we will encounter a loss of correlation with ensemble accuracy. The influence of the number of classes affects not only the margin of the majority voting, but also the sensitivity of the whole voting network as well, especially in the measure of diversity. Fig. $3.a$ shows that, on an ensemble of 7-classifiers, there are two groups of classifiers with different opinions in a 2-class problem ($C1 \sim C4$, and $C5 \sim C7$), and the majority voting rule needs at least 4 votes from classifiers for a decision to be made. By contrast, in a 6-class problem, the majority could be represented with only 2 votes (Fig.$3.b$), we have 6 groups with different outputs ($C1$ agrees with $C2$, but $C3$, $C4$, $C5$ and $C6$ all differ from one another). Note that we have the same margin of 1 vote in both cases. If we consider the majority class

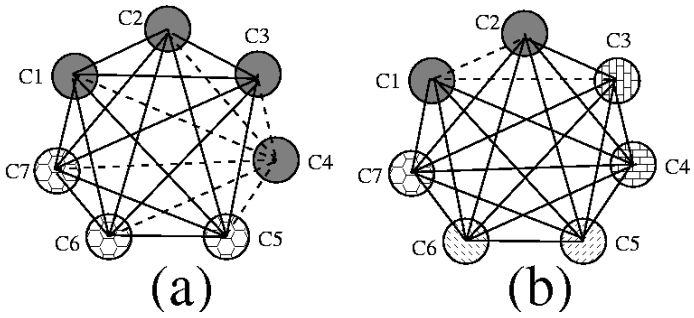

Figure 3 An ensemble of 7 classifiers (C1 $\sim$ C7); the shadowed circles represent the classifiers needed to achieve the majority, the solid lines represent the pairwise diversities among classifiers, and the dashed lines represent the required modified-pairwise-diversities so that the majority of votes could be shifted into another class: (a) at least 4 votes needed in 2-class problems; 6 modified pairwise-diversities needed for majority-shifting; (b) at least 2 votes needed in 6-class problems; 2 modified pairwise-diversities needed for majority-shifting. This figure serves only as an example. For details, please see appendix 2

shifting into another class, 6 pairwise diversities have to be modified in 2-class problems (i.e., if $C4$ agrees with $C5$, $C6$, $C7$, diversities must change between $C4$ and all other classifiers); and only 2 pairwise diversities need to be modified in 6-class problems (i.e., if C1 agrees with C3, diversities change only between $C1$ and $C3$, $C1$ and $C2$). This indicates that a large number of diversity changes in low-class problems may not affect the final output, but in high-class problems a slight change in diversity may lead to another final decision. Thus, the measure $\widehat{div_{sim}}$ / $\widehat{div_{amb}}$ is much more sensitive to ensemble behavior in high-class problems than it is in low-class problems.

This suggests that the implementation of proposed compound diversity functions should be much more effective dealing with high-class problems. Moreover, the fewer classifiers are selected in an ensemble, the more accurate the correlation between ensemble accuracy and compound diversity functions shall be.

2.5 Diversity Measures

Before we carried out the correlation measurements, we need to introduce some diversity measures that would be evaluated in our experiments. The traditional concept of diversity is composed of the terms of correct / incorrect classifier outputs. By comparing these correct / incorrect outputs among classifiers, their respective diversity can be calculated. In general, there are two kinds of diversity measures (See appendix 3 and 4):

a. Pairwise diversity measures

Diversity is measured between two classifiers. In the case of multiple classifiers, diversity is measured on all possible classifier-pairs, and global diversity is calculated as the average of the diversities on all classifier-pairs. That is, given L classifiers, $\frac{L\times(L-1)}{2}$ pairwise diversities $d_{12}, d_{13}, ..., d_{(L-1)L}$ will be calculated, and the final diversity $\bar{d}$ will be its average (66):

$$\bar{d} = 2 \times \frac{\sum_{ij} d_{ij}}{L \times (L-1)}, i \leq j \tag{2.31}$$

This type of diversity includes: Q-statistics (1; 5), the correlation coefficient (66), the disagreement measure (49) and the double fault (29).

b. Non-Pairwise diversity measures

There are others diversities that are not pairwise, i.e. they are not calculated by comparing classifier-pairs, but by comparing all classifiers directly. This type of diversity includes: the Entropy measure (66), Kohavi-Wolpert variance (61), the measurement of interrater agreement (5; 25), the measure of difficulty (47), generalized diversity (80) and coincident failure diversity (80).

Most research suggests that neither type of diversity is capable of achieving a high degree of correlation with ensemble accuracy, as only very weak correlation can be observed (66). As we see in the section 4, the proposed compound diversity functions might represent

better correlations with the ensemble accuracy. To verify its usefulness, we carried out the experiments of the correlation measurements in the next section.

2.6 Correlations between Diversity and Ensemble Accuracy

To make sure that the normalized compound diversity function is valid for the estimation of ensemble accuracy, we tested it on problems extracted from UCI machine learning repository. There are several requirements for the selection of pattern recognition problems. First, we should test three types of ensemble creation method: Random Subspaces, Bagging, and Boosting. Thus the databases must have a large feature dimension for Random Subspaces. Second, to avoid the dimensional curse during training, each database must have sufficient samples of its feature dimension. Third, to avoid identical samples being trained in Random Subspaces, only databases without symbolic features are used. Fourth, 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 four databases selected from the UCI data repository (See Table I).

Table I

UCI data for ensembles of classifiers

Database	Classes	Train	Test	Features	Random Subspace	Bagging	Boosting
Wisconsin Breast-Cancer	2	284	284	30	5	66 %	66 %
Satellite	6	4435	2000	36	4	66 %	66 %
Image Segmentation	7	210	2100	19	4	66 %	66 %
Letter Recognition	26	10007	9993	16	12	66 %	66 %

For each of 4 databases, for each of 3 ensemble creation methods (Random Subspaces, Bagging, and Boosting), and for each of 3 classification algorithms, 18 classifiers were

generated as the pool for base classifiers. Classifiers were then selected from this pool to construct ensembles. The three different classification algorithms used in our experiments are Naive Bayesian Classifiers (NBC), Quadratic Discriminant Classifiers (QDC), and 5-Layer Neural Network Classifiers (NNC) with Back-Propagation (19). To better understand the influence of the number of classifiers on the correlation between diversity and ensemble accuracy, ensembles were composed from $3 \sim 15$ classifiers. In total, we evaluated 13 different numbers of classifiers for ensembles. All correlations are measured for ensembles with the same number of classifiers, then the mean values of correlations from different numbers of classifiers are calculated. To obtain the most accurate measure, 50 ensembles were constructed with the same number of selected classifiers for each database, for each classification algorithm, for each ensemble method and for each different number of classifiers. We repeated this process 30 times to obtain a reliable evaluation. The simple majority voting rule is used as the fusion function for the evaluation of the global performances of related EoC. A total of $3 \times 3 \times 4 \times 13 \times 50 \times 30 = 0.702$ million ensembles should be evaluated. But, due to the dimensional curse, NNC did not have sufficient samples for training on the Image Segmentation problem or on the Satellite problem for Bagging or for Boosting. This occurred on $1 \times 2 \times 2 \times 13 \times 50 \times 30 = 0.078$ million ensembles, so in total $0.702 - 0.078 = 0.624$ million ensembles were evaluated in the experiment.

We measured ensemble accuracy correlation on 10 traditional diversity measures, including the disagreement measure (DM) (49), the double-fault (DF) (29), Kohavi-Wolpert variance (KW) (61), the interrater agreement (INT) (25), the entropy measure (EN) (66), the difficulty measure (DIFF) (47), generalized diversity (GD) (80), coincident failure diversity (CFD) (80), Q-statistics (Q) (1), and the correlation coefficient (COR) (66), as well as on 10 respective proposed compound diversity functions (eq. 2.26 $\&$ eq. 2.27). They are also compared with the Mean Classifier Error (ME) of individual classifiers. On all training databases, the proportion of selected samples in Bagging and Boosting is 66%.

For Random Subspaces, the sizes of subsets of features are decided under the condition that each classifier created must have recognition rates more than 50% .

2.6.1 Random Subspaces

In the Table II, we show the correlations between original diversity measures and ensemble accuracy, and the correlation between compound diversity functions and ensemble accuracy. NBC, QDC, and NNC are applied on all databases, and we show their average correlations.

First, we observe that in most cases the ME has an apparent correlation with ensemble accuracy. Furthermore, it shows that, in general, compound diversity functions give better results than the original diversity measures; it can also be perceived that, even though the correlation between ME and ensemble accuracy is weak, compound diversity functions still work well and present stronger correlations with ensemble accuracy than ME. Of all the diversity measures, Q, COR, INT and DIFF are not stable. By contrast, DM, DF, KW, EN, GD and CFD are quite reliable, as they always offer $43\% \sim 76\%$ of correlation with compound diversity functions. Note that in some cases (e.g., Wisconsin breast cancer), their correlation with ensemble accuracy is better than the correlation between ME and ensemble accuracy.

2.6.2 Bagging

The ensembles for the second experiment were created by Bagging. NBC and QDC are used on all the databases. But NNC is implemented on all of them except the Image Segmentation data and the Satellite data, given insufficient samples, because their high feature dimension caused the dimensional curse.

In Table III, there is a clear correlation between ME and ensemble accuracy, and it is quite strong. Of all the diversities, Q, COR, INT, and DIFF did not perform as well as

Table II

Correlation for the Random Subspaces method between ensemble accuracy and: (a) Mean Classifier Error; (b) the average of diversity measures; (c) the proposed compound diversity functions. The arrows indicate the expected correlations: $\downarrow$ for -1 and $\uparrow$ for 1

	Breast Cancer	Satellite	Image Segmentation	Letter Recognition
ME (Mean Classifier Error) ($\downarrow$)	-0.4447	-0.5820	-0.6147	-0.4680
Original Diversity Measures	**Breast Cancer**	**Satellite**	**Image Segmentation**	**Letter Recognition**
disagreement measure (DM) ($\uparrow$)	-0.0170	0.0779	-0.1860	-0.0577
double fault (DF) ($\downarrow$)	-0.3916	-0.1204	-0.4725	-0.3758
Kohavi-Wolpert variance (KW) ($\uparrow$)	-0.0170	0.0779	-0.1860	-0.0577
interrater agreement (INT) ($\downarrow$)	-0.3605	-0.0791	-0.0038	-0.0283
entropy measure (EN) ($\uparrow$)	-0.0170	0.0779	-0.1860	-0.0577
measure of difficulty (DIFF) ($\downarrow$)	0.2440	-0.1263	0.5518	0.1364
generalized diversity (GD) ($\uparrow$)	0.2893	0.0819	0.3547	0.1413
coincident failure diversity (CFD) ($\uparrow$)	0.2990	0.0807	0.3603	0.1526
Q-statistics (Q) ($\downarrow$)	-0.1705	-0.0811	0.1140	0.0460
correlation coefficient (COR) ($\downarrow$)	-0.3552	-0.0792	0.0120	-0.0266
Proposed Compound Diversity Functions	**Breast Cancer**	**Satellite**	**Image Segmentation**	**Letter Recognition**
disagreement measure (DM) ($\downarrow$)	-0.6379	-0.4563	-0.4310	-0.4449
double fault (DF) ($\downarrow$)	-0.4924	-0.4731	-0.5058	-0.4916
Kohavi-Wolpert variance (KW) ($\downarrow$)	-0.5407	-0.5337	-0.7616	-0.5014
interrater agreement (INT) ($\downarrow$)	-0.2416	-0.0462	-0.1010	-0.1496
entropy measure (EN) ($\downarrow$)	-0.6379	-0.4563	-0.4310	-0.4449
measure of difficulty (DIFF) ($\downarrow$)	-0.3292	-0.2877	0.0708	-0.1200
generalized diversity (GD) ($\downarrow$)	-0.4551	-0.4978	-0.5951	-0.4851
coincident failure diversity (CFD) ($\downarrow$)	-0.4264	-0.4561	-0.5292	-0.4490
Q-statistics (Q) ($\downarrow$)	-0.3362	-0.2355	-0.1224	-0.4410
correlation coefficient (COR) ($\downarrow$)	-0.2488	-0.0468	-0.0998	-0.1498

the others. The GD and CFD results are unstable; sometimes giving good correlation but sometimes not. DM, KW and EN are stable, though a little bit weaker than those in Random Subspaces. Since the selected databases have high feature dimension for the implementation of Random Subspaces, as a result, the effect of the dimensional curse might occur for Bagging and for Boosting. KW always performed at $43\% \sim 83\%$ on our compound diversity function.

Table III

Correlation for Bagging method between ensemble accuracy and: (a) Mean Classifier Error; (b) the average of pure diversity measures; (c) the proposed compound diversity functions. Note that the arrows indicate the expected correlations: $\downarrow$ for -1 and $\uparrow$ for 1

	Breast Cancer	Satellite	Image Segmentation	Letter Recognition
ME (Mean Classifier Error) ($\downarrow$)	-0.5516	-0.5151	-0.8113	-0.5906
Original Diversity Measures	**Breast Cancer**	**Satellite**	**Image Segmentation**	**Letter Recognition**
disagreement measure (DM) ($\uparrow$)	-0.2902	0.1309	-0.2306	0.1771
double fault (DF) ($\downarrow$)	-0.0409	-0.2131	-0.3520	-0.2603
Kohavi-Wolpert variance (KW) ($\uparrow$)	-0.2902	0.1309	-0.2306	0.1771
interrater agreement (INT) ($\downarrow$)	-0.0219	-0.1356	0.2298	-0.1340
entropy measure (EN) ($\uparrow$)	-0.2902	0.1309	-0.2306	0.1771
measure of difficulty (DIFF) ($\downarrow$)	0.4925	-0.2024	-0.3516	0.0224
generalized diversity (GD) ($\uparrow$)	-0.1122	0.1313	-0.2273	0.2149
coincident failure diversity (CFD) ($\uparrow$)	-0.1178	0.1314	-0.2321	0.2150
Q-statistics (Q) ($\downarrow$)	0.1068	-0.1283	-0.1692	0.0570
correlation coefficient (COR) ($\downarrow$)	-0.0058	-0.1386	-0.1686	-0.1309
Proposed Compound Diversity Functions	**Breast Cancer**	**Satellite**	**Image Segmentation**	**Letter Recognition**
disagreement measure (DM) ($\downarrow$)	-0.5269	-0.3689	-0.3700	-0.5656
double fault (DF) ($\downarrow$)	-0.3370	-0.4798	-0.6645	-0.5663
Kohavi-Wolpert variance (KW) ($\downarrow$)	-0.5431	-0.4384	-0.8329	-0.6005
interrater agreement (INT) ($\downarrow$)	-0.2086	-0.1798	-0.0050	-0.1443
entropy measure (EN) ($\downarrow$)	-0.5269	-0.3689	-0.3700	-0.5656
measure of difficulty (DIFF) ($\downarrow$)	-0.2359	-0.3978	-0.3873	-0.3256
generalized diversity (GD) ($\downarrow$)	-0.3331	-0.3962	-0.6721	-0.4922
coincident failure diversity (CFD) ($\downarrow$)	-0.2864	-0.3672	-0.3683	-0.4702
Q-statistics (Q) ($\downarrow$)	-0.5094	-0.4559	-0.1190	-0.4109
correlation coefficient (COR) ($\downarrow$)	-0.2014	-0.1867	-0.0846	-0.1450

We note that, in general, the correlations between the diversities and ensemble accuracy for Bagging are weaker than those for Random Subspaces. But, on high-dimension-class problems, (e.g. letter recognition data, image segmentation), the implementation of compound diversity functions is just as good for Bagging as for Random Subspaces. The advantage of compound diversity functions over the original diversity measures can be perceived in this case.

2.6.3 Boosting

The ensembles were created for the third experiment by Boosting, NBC and QDC are used on all databases, but NNC is used on all except the Image Segmentation data and the Satellite data, because, given insufficient samples, their high feature dimension caused the dimensional curse.

Table IV

Correlation for Boosting method between ensemble accuracy and: (a) Mean Classifier Error; (b) the average of pure diversity measures; (c) the proposed compound diversity functions. Note that the arrows indicate the expected correlations: $\downarrow$ for -1 and $\uparrow$ for 1

	Breast Cancer	Satellite	Image Segmentation	Letter Recognition
ME (Mean Classifier Error) (↓)	-0.4828	-0.5173	-0.3405	-0.6148
Original Diversity Measures	Breast Cancer	Satellite	Image Segmentation	Letter Recognition
disagreement measure (DM) (↑)	-0.1392	-0.2849	-0.2370	0.4086
double fault (DF) (↓)	-0.0047	0.3131	0.2549	-0.3408
Kohavi-Wolpert variance (KW) (↑)	-0.1392	-0.2849	-0.2370	0.4086
interrater agreement (INT) (↓)	-0.0538	0.1283	-0.1497	-0.3926
entropy measure (EN) (↑)	-0.1392	-0.2849	-0.2370	0.4086
measure of difficulty (DIFF) (↓)	0.3652	0.3505	0.2647	-0.1940
generalized diversity (GD) (↑)	-0.0576	-0.2949	-0.2410	0.4092
coincident failure diversity (CFD) (↑)	-0.0558	-0.3115	-0.2436	0.4109
Q-statistics (Q) (↓)	0.0873	0.1923	0.0471	-0.2980
correlation coefficient (COR) (↓)	-0.0638	0.1293	-0.1498	-0.3912
Proposed Compound Diversity Functions	Breast Cancer	Satellite	Image Segmentation	Letter Recognition
disagreement measure (DM) (↓)	-0.5599	-0.1080	-0.0219	-0.5410
double fault (DF) (↓)	-0.3878	-0.0462	0.0364	-0.5351
Kohavi-Wolpert variance (KW) (↓)	-0.5487	-0.4489	-0.3708	-0.5681
interrater agreement (INT) (↓)	-0.1807	0.0607	-0.0275	-0.3129
entropy measure (EN) (↓)	-0.5599	-0.1080	-0.0219	-0.5410
measure of difficulty (DIFF) (↓)	-0.2825	0.0729	0.0854	-0.4388
generalized diversity (GD) (↓)	-0.3459	-0.2538	-0.1226	-0.5226
coincident failure diversity (CFD) (↓)	-0.3182	-0.0660	-0.0008	-0.4693
Q-statistics (Q) (↓)	-0.5448	-0.1134	-0.0299	-0.3180
correlation coefficient (COR) (↓)	-0.1980	0.0611	-0.0272	-0.3130

On most of the databases, there is a strong correlation between ME and ensemble accuracy (Table IV). Interestingly, it is in Boosting that we see how the implementation of diversity really matters: the correlation by the proposed compound diversity function could be equivalent to or better than that of ME, which means that, for Boosting, the notion of diversity does help to obtain a strong correlation with ensemble accuracy. Nevertheless, we also perceive that the correlations between the diversities and ensemble accuracy are weaker for Boosting than those for Bagging and for Random Subspaces for low-dimension-class problems. But, when the number of classes is large (e.g. letter recognition data), the correlation on Boosting can be as good as that on Bagging, and the notion of diversity is quite well with compound diversity functions. In high-class-problems, the useful diversity measures appear to be DM, DF, KW, EN, DIFF, GD and CFD. They offer correlations between $46\% \sim 56\%$.

2.6.4 Discussion on the Correlation between Diversity and Ensemble Accuracy

In all three ensemble creation methods, we first note that the proposed compound diversity functions correlate much stronger with the ensemble accuracy than the traditional diversity measures. Second, comparison of the various ensemble creation methods suggests that, in Random Subspaces, the proposed compound diversity functions generally have the strongest correlations with ensemble accuracy, better than in Bagging or in Boosting. Nevertheless, considering the correlation with ensemble accuracy, compound diversity functions could perform better than ME in Boosting. This suggests that the issue of ensemble diversity is crucial in Boosting.

It is certain that the number of classifiers has an impact on the correlation between compound diversity functions and ensemble accuracy. We found the strongest correlation with ensemble accuracy on the minimum number of classifiers, i.e. when ensembles were constructed with only 3 classifiers. But this correlation could decrease to nearly 0 when the number of classifiers is close to the total number of classifiers available in the pool, as we

explained in the section 5. A typical example is shown in Fig. 4, and this tendency is observed on all our experimental problems. This is the reason why the measured average correlation is not too significant compared with the ME.

2.7 Ensemble Selection and Diversity as Objective Function

Even though the experiment shows that the compound diversity functions are strongly correlated with ensemble accuracy, it is important to show that such functions can be used as objective functions for ensemble selection. Thus we carried out a number of experiments using different diversities as objective functions for ensemble selection. These objective functions are evaluated by genetic algorithm (GA) searching. We used a GA because the complexity of population based searching algorithms can be flexibly adjusted depending on the size of the population and the number of the generations to proceed. Moreover, because the algorithm returns population of the best combination, it can be potentially exploited to prevent generalization problems (89). We tested 20 different diversities, including 10 compound diversity functions and 10 original diversity measures. Besides these 20 different objective functions, we also used the Mean Classifier Error (ME) and the error of ensembles applying the majority voting (MVE). We then compared their effectiveness as objective functions for the creation of the EoC.

2.7.1 Experimental Protocol for Ensemble Selection

We carried out experiments on a 10-class handwritten-numeral problem. The data was extracted from $NISTSD19$, essentially as in (99), 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 ($hsf\_\{0-3\}$) was used to create 100 KNN in Random Subspaces, and the optimization set containing 10000 samples ($hsf\_\{0-3\}$) was used for GA searching. To avoid overfitting during GA searching, the validation set containing 10000 samples ($hsf\_\{0-3\}$)

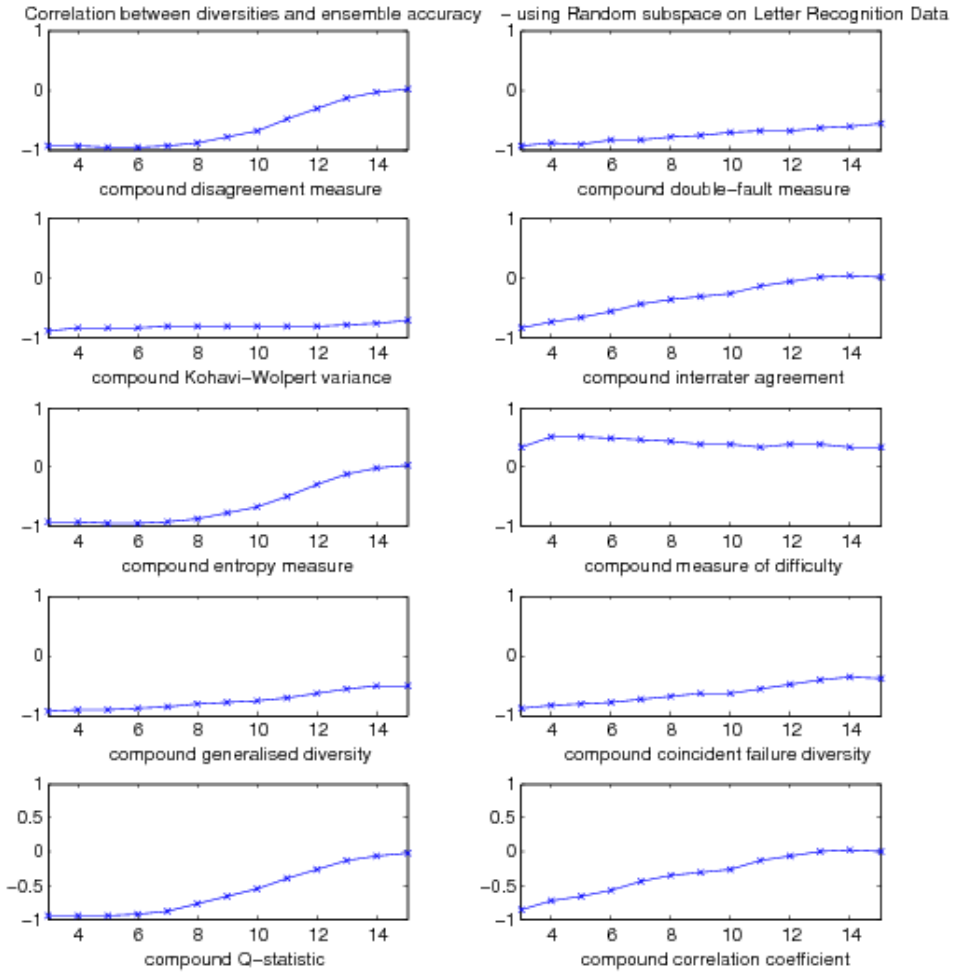

Figure 4 The correlations between the CDFs and the accuracy on the letter recognition problem extracted from the UCI machine learning database with the Random subspaces as the ensemble creation method. We can observe that the larger the ensemble size, the lower the correlation

was used to select the best solution from the current population according to the defined objective function, and then to store it in a separate archive after each generation. Using the best solution from this archive, the test set containing 60089 samples ($hsf\_\{7\}$) was used to evaluate the accuracies of EoC. We used GA as the searching algorithm, with 128 individuals in the population and with 500 generations, which means $64,000$ ensembles were evaluated in each experiment. The mutation probability is 0.01. With 22 different objective functions (Mean Classifier Error (ME), Majority Voting Error (MVE), 10 original diversity measures, and 10 compound diversity functions) and 30 replications, 42.24 million ensembles were searched and evaluated. A threshold of 3 classifiers was applied as the minimum number of classifiers for EoC during the whole searching process. Experimental results are reported in Table V.

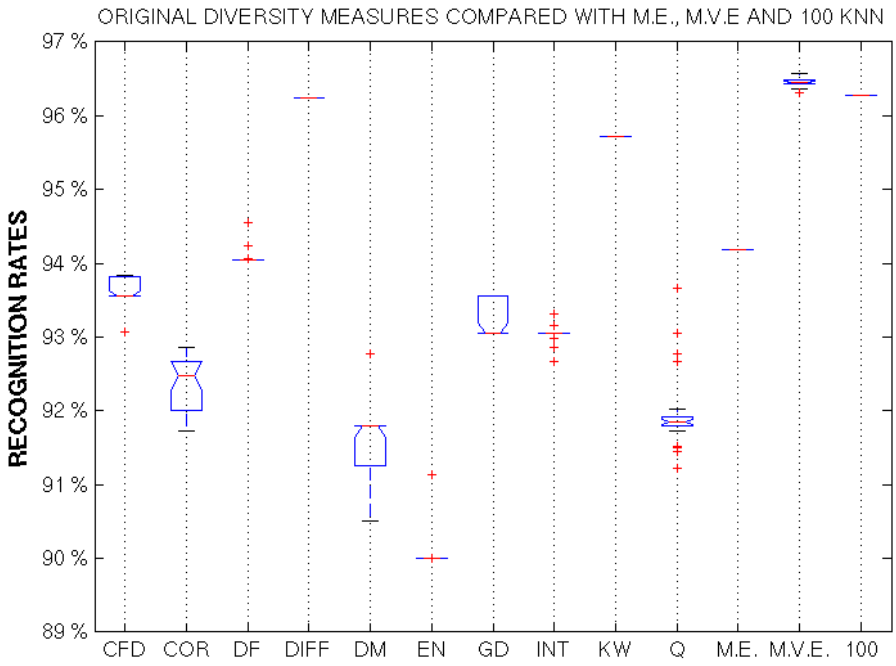

Figure 5 The recognition rates achieved by EoCs selected by original diversity measures, compared with the Mean Classifier Error (ME), Majority Voting Error (MVE), and the ensemble of all (100) KNN classifiers

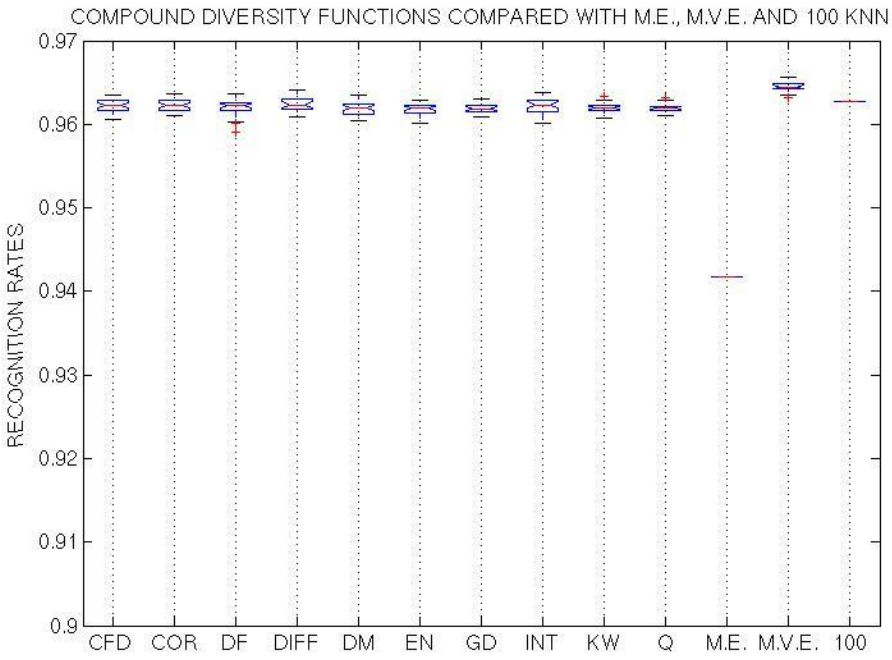

Figure 6 The recognition rates achieved by EoCs selected by compound diversity functions, compared with the Mean Classifier Error (ME), Majority Voting Error (MVE), and the ensemble of all (100) KNN classifiers

Table V

The recognition rates of the ensembles selected by different objective functions, including traditional diversity measures and compound diversity functions (CDF), on NIST SD19 handwritten numerals

	100 KNN	ME	MVE	
	96.28 ± 0.00 %	94.18 ± 0.00 %	96.45 ± 0.05 %	
DM	KW	EN	GD	CFD
91.56 ± 0.46 %	95.72 ± 0.00 %	90.04 ± 0.21 %	93.26 ± 0.25 %	93.66 ± 0.18 %
INT	DIFF	DF	Q.	COR
93.04 ± 0.11 %	96.24 ± 0.00 %	94.10 ± 0.13 %	91.96 ± 0.52 %	92.44 ± 0.37 %

CDF-DM	CDF-KW	CDF-EN	CDF-GD	CDF-CFD
96.19 ± 0.09 %	96.20 ± 0.06 %	96.18 ± 0.08 %	96.19 ± 0.05 %	96.22 ± 0.08 %
CDF-INT	CDF-DIFF	CDF-DF	CDF-Q.	CDF-COR
96.22 ± 0.09	96.23 ± 0.08 %	96.20 ± 0.10 %	96.20 ± 0.05 %	96.23 ± 0.07 %

First, we see that the use of traditional diversity measures does not always give satisfying performance. The results show that the selected ensembles perform poorly, most of them are even worse than those chosen by ME. Apparently there are many outliers indicated in the box plot (Fig. 5), which are values exceeding the distance of 1.5 interquartile range ($Q_U - Q_L$) from either end of the box, which means that searching by the traditional diversity measures could lead to great instability. This phenomenon is understandable, in light of the fact that the original diversity measures were designed to optimize diversity among classifiers, and they do not target ensemble accuracy directly. The result also confirms the lack of correlation between most diversity measures and ensemble accuracy.

As we predicted, all pairwise diversity measures will lead to the minimum number of classifiers, i.e., 3 classifiers in this experiment. Moreover, some non-pairwise diversity measures will lead to 3 classifiers, since it will not be easy to find an ensemble with greater diversity than the ensemble composed of the 3 most diverse classifiers. The only two diversity measures that can resist the minimum-converging tendency are KW, which always finds 17 classifiers for EoC, and DIFF with 21 classifiers. DIFF performs relatively well in this case, as had been shown in (92). It seems that DIFF, the minimization of the variance of the proportion of correct classifiers on all samples, encourages fairly distributed difficulty, instead of selecting the most diverse classifiers. To arrive at a fair distribution of difficulty, a number of classifiers would be required. Even DIFF did not have strong correlation with ensemble accuracy in our previous correlation measurement; it does guarantee a comparable performance in this case.

By contrast, the proposed compound diversity functions are much more stable (Fig. 6). Most EoCs selected by them are constructed by $35 \sim 60$ classifiers, which is about half the total of 100 classifiers. Compared with the EoCs found by MVE with $19 \sim 35$ classifiers, the sizes of EoCs selected by the compound diversity functions are larger, but the performances are quite stable. Though MVE is still clearly better with the significance $p < 0.01$, the differences in recognition rates with EoCs selected by MVE are usually less

than 0.3%. This indicates that the EoCs selected by the proposed compound functions are quite generalized and fit different fusion functions.

Finally, we point out that, among all diversity measures, the compound diversity functions always perform better than the original diversity measures. While most of the original diversity measures perform worse than ME, the use of the compound diversity functions gives much better results than ME. Furthermore, all compound diversity functions achieve similar performances; which should result from the strong correlations among most of them.

2.8 Discussion

Previous published studies suggested that diversity is not unequivocally related to ensemble accuracy, and it is our objective to demonstrate that the implementation of diversity can help in ensemble selection. As we can see in these experiments, there are correlations between the proposed compound diversity functions and ensemble accuracy. The result also suggests that DM, KW, EN, GD and CFD are stable for all ensemble creation methods. Performance depends strongly on the accuracy of individual classifiers, but, in general, an equivalent or stronger correlation could be achieved with compound diversity functions, especially with KW.

In contrast to the use of the original diversity measures, which show no strong intercorrelation (63), these compound diversity functions do have strong intercorrelations, except for COR, DIFF, INT, and Q. This means that most diversities have similar indication, and so the creation of new diversity measures might not be a priority, but rather consideration of how to use diversities for ensemble selection. With the Random Subspaces method, this correlation is stronger than it is in either Bagging or Boosting. In general, a decrease in correlation is observed when the number of selected classifiers increases, but this was not the case for high-class problems, as we predicted.

Based on GA searching, we see that the compound diversity functions apparently outperform the original diversity measures and the Mean Classifier Error as objective functions for ensemble selection, and even exceed the performance of the ensemble of all 100 KNN classifiers and reduce the number of classifiers by half. The proposed compound diversity functions do improve the performance of EoCs, and always perform better than the respective original diversity measures, their performances being much close to those ensembles obtained with the MVE objective function.

Recall that MVE is used both for ensemble selection and for classifier combination, and thus it is understandable that MVE will have the best performance as the objective function. But, it is possible that when different fusion functions are used, MVE will not be the best choice as an objective function. An ensemble combined with Decision Template (DT), for example, might not have the best performances when it is evaluated by MVE. The 'no free lunch' theorem (105; 106) has also supported the idea that no search algorithm will be optimal in all situations.

Given that these compound diversity functions do not take into account of any fusion functions, the ensemble outputs can be further optimized using various classifier-combining methods (56; 88; 89). This is an advantage for modular approaches to further optimize searching algorithms and fusion functions. All the compound diversity functions worked well for ensemble selection in our experiment, even some that had previously been measured and found to have weaker correlation with ensemble accuracy. This indicates a strong similarity among most of the compound diversity functions in the pattern recognition problems evaluated.

The result encourages further exploration of the implementation of compound diversity functions, and the pertinence of these functions for use with different searching algorithms. Moreover, it suggests that the problem resides in finding ways to amalgamate diversities and individual classifier errors, rather than allowing diversity measures to se-

lect EoCs single-handedly. Another advantage of compound diversity functions is that they can be calculated beforehand, since diversities are measured in a pairwise manner, and error rates are measured on each classifier; thus, for time-consuming searching methods, such as GA or exhaustive searching, ensemble accuracy can be estimated quickly by simply calculating the products of the diversity measures and individual classifier errors. Given L classifiers and N samples on a C-class problem, the copmplexity of the CDFs is $O(L + \frac{L(L-1)}{2})$, the complexity of non-pairwise traditional diversity measures is $O(LN)$, and the complexity of the MVE is $O(LNC)$. The CDFs thus has the lower cost for the ensemble selection.

2.9 Conclusion

Diversity used to be regarded as useful, but not unequivocally related to ensemble accuracy. In this exploratory work on diversity, we show that, with the proper compound diversity functions, there are strong correlations between the diversities and ensemble accuracy. Moreover, using population-based GA searching, the compound diversity functions do improve the recognition rates of the ensembles. We have drawn up some conclusion based on our experiments:

a. Diversities and the performances of individual classifiers should be taken into account together.

b. Compound diversity functions have stronger correlations with the ensemble accuracy than the traditional diversity measures.

c. Compared with MVE, compound diversity functions have lower cost for the ensemble selection.

d. In general, ensembles selected by different compound diversity functions have so far been found to have similar performances for GA searching, with the significance $p \geq 0.1$.

Given that this exploratory work has been accomplished with different ensemble creation methods, considering different numbers of classifiers of ensembles, evaluating millions of ensembles, but with a restricted number of classification algorithms, and in a limited number of problems, it will be advisable to carry out more experiments on ensemble selection, with more pattern recognition problems and more classification methods. The problems associated with optimizing ensembles include not only diversity, but also searching algorithms (89) and fusion functions (56).

At the next chapter, we will test different fusion functions on ensembles selected with the proposed compound diversity functions, compared with those selected with MVE. To further optimize the performance of an EoC, we will propose other fusion functions. These fusion functions are, interestingly, also based on a pairwise concept like compound diversity functions.

CHAPTER 3

PAIRWISE FUSION MATRIX FOR COMBINING CLASSIFIERS

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 produce 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.

3.1 Introduction

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 (20; 56; 64; 89; 109).

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 Random Subspaces (49), Bagging and Boosting (31; 63; 90). 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 (50; 56; 69; 81; 89; 92; 96; 104; 109; 111). In general, two kinds of fusion functions are available: (a) Fusion functions of label outputs, such as majority voting, Behavior Knowledge Space, Naive Bayes methods, etc.; and (b) Fusion functions of continuous-values outputs, which require the class probabilities 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 is suggested.

To improve the performance of the fusion functions of label outputs, the Behavior-Knowledge Space (BKS) (50) 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 (DT), Dempster-Shafer combination (DSC), Fuzzy

Integral, or multilayer perceptrons (MLP) (50; 69; 92; 104). 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 (87). It has, in fact, been suggested that, given insufficient training samples, simple fusion functions may outperform some trained fusion functions (87).

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:

a. Can label outputs classifiers apply continuous-valued fusion functions?

b. Can a trainable fusion function perform well without a large training dataset?

c. 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 3-dimension 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. 7) by which an ensemble of L classifiers is transformed into another ensemble of $\frac{L\times(L-1)}{2}$ 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 pairwise fusion matrix does not require as many samples needed for ensemble training as the BKS.

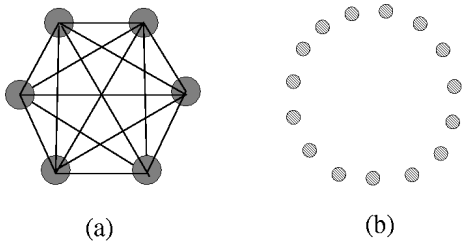

Figure 7 An example of pairwise confusion matrices transformation in a 6-classifier ensemble. (a) The original ensemble with 6 classifiers; and (b) the transformation yields to $\frac{6\times5}{2} = 15$ classifier pairs. Note that each classifier pair is equal to the link between two classifiers in (a)

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 chapter constitute only a small step towards a considerably improved understanding of classifier combination.

The chapter 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 Behavior Knowledge Space (BKS) in

section 4. Experimental results are compared in section 5. Discussion and our conclusion are presented in the remaining sections.

3.2 Fusion Functions for Label Outputs Classifier Combination

Several fusion functions of label outputs for combining classifiers have been proposed (20; 56; 64; 109). These directly compare the outputs from all individual classifiers in an ensemble. Some related theoretical studies are presented in (56; 64; 109). As stated in (64; 100), 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 (50; 69; 92; 104). We detail some popular fusion functions of label outputs in the section below.

3.2.1 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 $c(i), 1 \leq c(i) \leq T$, and we write $d_{i,t} = 1$ for $c(i) = t, 1 \leq t \leq T$ and zero otherwise. Consequently, we calculate the discriminant function for class $l, 1 \leq l \leq T$ as :

$$g(l|x) = \sum_{i=1}^{L} d_{i,l} \tag{3.1}$$

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

$$k = \arg \max_{l=1}^{T} g(l|x) \tag{3.2}$$

3.2.2 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 :

$$g(l|x) = \sum_{i=1}^{L} b_i d_{i,l} \tag{3.3}$$

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

$$\sum_{i=1}^{L} b_i = 1 \tag{3.4}$$

It has been suggested that if each classifier is independent from one another, than the coefficient b_i can be set as (69):

$$b_i \propto \log \frac{p_i}{1 - p_i} \tag{3.5}$$

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

3.2.3 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 $l, 1 \leq l \leq T$ 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 :

$$\tilde{P}(l|x) \propto \prod_{i=1}^{L} P(l|c(i), x) \tag{3.6}$$

$$k = \arg\max_{l=1}^{T} \tilde{P}(l|x) \tag{3.7}$$

This is the so-called naive nayes (NB) combination (92; 109). However, it is very unlikely that all classifiers in an ensemble will be mutually independent.

3.2.4 Behavior-Knowledge Space (BKS) and Wernecke's method (WER)

Some authors propose constructing a complex BKS table (50) 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 $P(l|c(1), \cdots, c(i), \cdots, c(L), x)$ for the whole data X, where l is a possible class label for a sample $1 \leq l \leq T$, and $c(i)$ is the decision of classifier $f(i)$ over the sample, with L classifiers $1 \leq i \leq L$. 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 Behavior-Knowledge Space (BKS) (50), or considering a 95% confidence interval of the probabilities in the BKS table, known as Wernecke's method (WER) (104). For BKS, the class is assigned by simply comparing the values in each cell in BKS table :

$$k = \arg\max_{l=1}^{T} P(l|c(1), \cdots, c(i), \cdots, c(L), x) \tag{3.8}$$

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 (69), as well as being too self-assured (87).

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.

3.3 The Concept of Pairwise Fusion Matrices

3.3.1 Pairwise Fusion Matrix Transformation (PFM)

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 Transformation (PFM)

makes use of pairwise estimation to solve this problem. If we only take classifier pairs into account, we need only 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:

$$P(l|c(i), c(j), x) = n(x \in l, c(i), c(j))/n(c(i), c(j)) \tag{3.9}$$

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, $1 \leq l \leq T$. The probability $P(l|c(i), c(j), x)$ is, in fact, the concept of a 3-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 $f(j)$:

$$P(l|c(j),x) = \sum_{i=1}^{T} P(l|c(i),c(j),x) \tag{3.10}$$

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 $\frac{L\times(L-1)}{2}$ classifier pairs, we can obtain $\frac{L\times(L-1)}{2}$ 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):

$$P(l|c(i),c(j),x) = \frac{1}{N}\sum_{x=1}^{N} P(l|c(i),x)\cdot P(l|c(j),x) \tag{3.11}$$

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-values 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.

3.3.2 Apply PFM on fusion functions of Continuous-values 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-values outputs:

a. PFM-Maximum Rule (PFM-MAX)

$$k = \arg \max_{l=1}^{T} \max_{i,j=1,i\neq j}^{\frac{L}{2}} P(l|c(i), c(j), x) \tag{3.12}$$

b. PFM-Minimum Rule (PFM-MIN)

$$k = \arg \max_{l=1}^{T} \min_{i,j=1,i\neq j}^{\frac{L}{2}} P(l|c(i), c(j), x) \tag{3.13}$$

c. PFM-Sum Rule (PFM-SUM)

$$k = \arg \max_{l=1}^{T} \frac{2}{L \times (L-1)} \sum_{i,j=1,i\neq j}^{\frac{L}{2}} P(l|c(i), c(j), x) \tag{3.14}$$

d. PFM-Product Rule (PFM-PRO)

$$k = \arg\max_{l=1}^{T} \prod_{i,j=1,i\neq j}^{\frac{L}{2}} P(l|c(i), c(j), x) \tag{3.15}$$

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.

3.3.3 Apply PFM on fusion functions of label outputs

Although one of the advantages of PFM lies on the use fusion functions of continuous-values outputs, PFM can apply on fusion functions of label outputs as well. Given that MAJ can outperform some fusion functions of continuous-values outputs (87), 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 $\frac{L\times(L-1)}{2}$ 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 :

$$\tilde{l} = \arg\max_{l=1}^{T} P(l|c(i), c(j), x) \tag{3.16}$$

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

$$g(\hat{l}|x) = \sum_{i,j=1;i\neq j}^{L} d_{i,j|\tilde{l}} \tag{3.17}$$

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

$$k = \arg\max_{\hat{l}=1}^{T} g(\hat{l}|x) \tag{3.18}$$

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 $1 \leq l \leq 10$, $P(3|c(1) = 3, c(2) = 8, x)$, $P(3|c(1) = 3, c(3) = 5, x)$ 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.

3.3.4 Other Alternatives for PFM

We have shown that PFM can apply on both label outputs and continuous-values 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:

a. 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 $1 \leq i, j \leq L, i \neq j$ and $1 \leq l \leq T$, 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).

$$\dot{P}(l|c(i), c(j), x) = P(l|c(i), c(j), x) * R(f(i)) * R(f(j)) \tag{3.19}$$

b. 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 the different properties of diversity measures, we make use of them in different ways (PFM-DIV):

$$\ddot{P}(l|c(i),c(j),x) = P(l|c(i),c(j),x) * R(f(i)) * R(f(j)) * div_{amb}(f(i),f(j)) \tag{3.20}$$

$$\ddot{P}(l|c(i),c(j),x) = P(l|c(i),c(j),x) * R(f(i)) * R(f(j)) * (1 - div_{sim}(f(i),f(j))) \tag{3.21}$$

c. 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):

$$\breve{P}(l|c(i),c(j),x) = P(l|c(i),c(j),x) * P(c(i)|x) * P(c(j)|x) \tag{3.22}$$

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.

3.4 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 3 classifiers f_i, f_j, f_k are constructed for BKS, the class l_{max} is selected among all classes $l, 1 \leq l \leq L$ as the ensemble output on a sample x if :

$$l_{max} = \arg\max_{l} n(l|c_i, c_j, c_k) \tag{3.23}$$

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 $l_{max}(c_j, c_k)$.

$$l_{max}(c_i, c_j) = \arg\max_{l} n(l|c_i, c_j) \tag{3.24}$$

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 $n(l|c_i, c_j)$:

$$n(l|c_i, c_j) = n(l|c_i, c_j, c_k) + n(l|c_i, c_j, \bar{c_k}) \tag{3.25}$$

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 :

$$l_{max}(c_i, c_j) = \arg\max_{l}(n(l|c_i, c_j, c_k) + n(l|c_i, c_j, \bar{c_k})) \tag{3.26}$$

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

$$n(l_{max}|c_i, c_j, c_k) + n(l_{max}|c_i, c_j, \bar{c_k}) > n(\bar{l_{max}}|c_i, c_j, c_k) + n(\bar{l_{max}}|c_i, c_j, \bar{c_k}) \quad (3.27)$$

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

$$n(l_{max}|c_i, c_j, c_k) - n(\bar{l_{max}}|c_i, c_j, c_k) > n(\bar{l_{max}}|c_i, c_j, \bar{c_k}) - n(l_{max}|c_i, c_j, \bar{c_k}) \quad (3.28)$$

Note that from the BKS, we already know that :

$$n(l_{max}|c_i, c_j, c_k) > n(\bar{l_{max}}|c_i, c_j, c_k) \quad (3.29)$$

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

$$n(l_{max}|c_i, c_j, c_k) - n(\bar{l_{max}}|c_i, c_j, c_k) > 0 \quad (3.30)$$

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.

3.5 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 combinaion. 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 continuos-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 chapter focuses thus on the comparison of PFM and other fusion methods of crisp label outputs, such as the Naive Bayes Combination (NB), the Behavior Knowledge Space (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. At this chapter 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 $NIST\ SD19$ with 100 classifiers. The experimental protocols and the results are shown in the following sections.

3.5.1 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 sim-

plify 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 VI). Among available samples, in general, 50% are used as a training data set, and 50% are used as a test data set, except for the Image Segmentation dataset, 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, Bagging and Boosting (63; 90). 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 available training data set. The cardinality of Random Subspace is set under the condition that all classifiers have recognition rates 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) (19). For each of 13 databases and for each of 3 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 $3 \sim 8$ classifiers. 30 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.

Table VI

UCI data for ensembles of classifiers

Database	Classes	Training Samples	Test Samples	Features	Random Subspace	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 Recogntion	26	10000	10000	16	12	66 %	66 %

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 (89). 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 (Table VII $\sim$ IX).

In previous studies, the BKS has been shown to be comparatively accurate when an ensemble of 3 classifiers is involved (31), but the BKS could be outperformed by most of the other fusion functions when more classifiers are involved (69). In our study, the BKS apparently performs very well in 2- and 3-class problems (Table VII $\sim$ IX). 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 one of the best fusion functions for classifiers with only crisp class label outputs

Table VII

Comparison of recognition rates of different fusion functions with Random Subspace on UCI machine learning problems. All numbers are in percents (%), the variances are indicated in parenthesis. Note that 3 classification algorithms were used and only average values are shown here

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) %
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	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) %	
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	**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) %	

(89), 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 (Table VII $\sim$ IX). 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

Table VIII

Comparison of recognition rates of different fusion functions with Bagging on UCI machine learning problems. All numbers are in percents (%), the variances are indicated in parenthesis. Note that 3 classification algorithms were used and only average values are shown here

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) %
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	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 %	
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	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 (-) %	

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

Table IX

Comparison of recognition rates of different fusion functions with Boosting on UCI machine learning problems. All numbers are in percents (%), the variances are indicated in parenthesis. Note that 3 classification algorithms were used and only average values are shown here

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 (-)** %	

3.5.2 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 simpli-

fied 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 $NIST\ SD19$. It is a 10-class problem and the problem includes more than 150000 samples for the training and the validation, 60089 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.

3.5.2.1 Experimental Protocol for KNN

We carried out experiments on a 10-class handwritten numeral problem. The data were extracted from $NIST\ SD19$, essentially as in (99), 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 10000 samples ($hsf\_\{0-3\}$) was used for genetic algorithm (GA) searching for ensemble selection. To avoid overfitting during GA searching, the selection set containing 10000 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 archive, the test set containing 60089 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 (76; 74; 85). But, since this is not the objective of this chapter, 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 to 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 2 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 (89), 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. More-

over, because the algorithm returns a population of the best combinations, it can potentially be exploited to prevent generalization problems (89). GA was set with 128 individuals in the population and 500 generations, which means that 64000 ensembles were evaluated in each experiment. The mutation probability is 0.01. With 11 different objective functions (Majority Voting Error (MVE) and 10 compound diversity functions (58), including the disagreement measure (DM) (49), the double-fault (DF) (29), Kohavi-Wolpert variance (KW) (61), the interrater agreement (INT) (25), the entropy measure (EN) (66), the difficulty measure (DIFF) (47), generalized diversity (GD) (80), coincident failure diversity (CFD) (80), Q-statistics (Q) (1), and the correlation coefficient (COR) (66)), and with 30 replications. A threshold of 3 classifiers was applied as the minimum number of classifiers for an EoC during the whole searching process (Tables X). To summarize, 10 different fusion functions were tested.

Table X

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

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%

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-

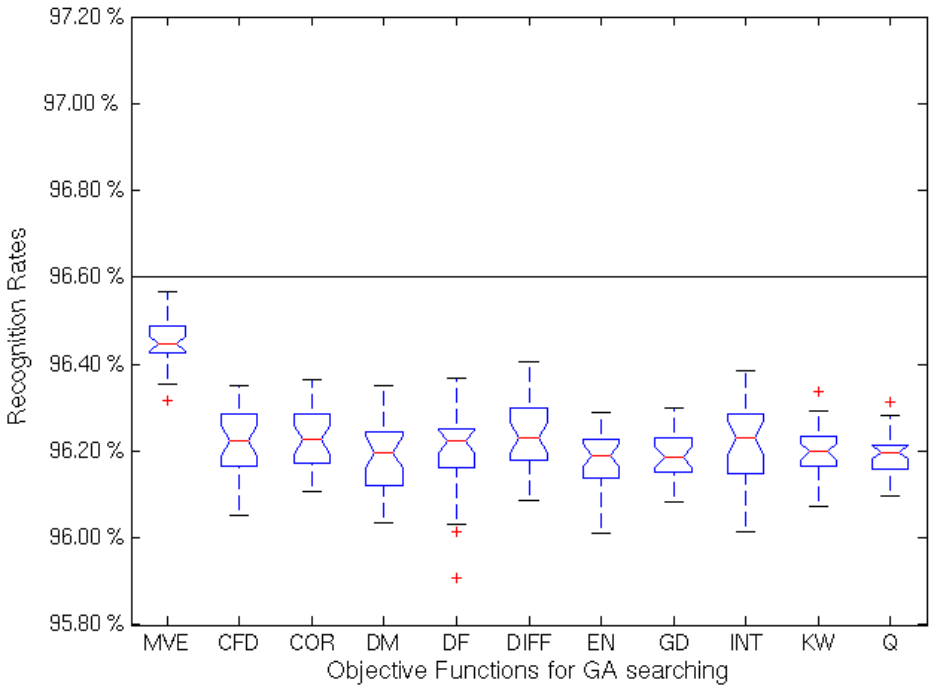

Figure 8 The recognition rates achieved by EoCs selected by 10 compound diversity functions and Majority Voting Error (MVE), using the simple MAJ as fusion function

DIV-MAJ can lead to a better performance (Table X). 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 (Fig. 8 & Fig. 9). 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 sub-optimal fusion functions compared with the SUM or the MAJ (56). Given that 100 classifiers generate 4950 classifier-pairs, an extremely biased value of the probability from any classifier-pairs can affects the results seriously with the PFM-MAX, the PFM-MIN or the PFM-PROD rules.

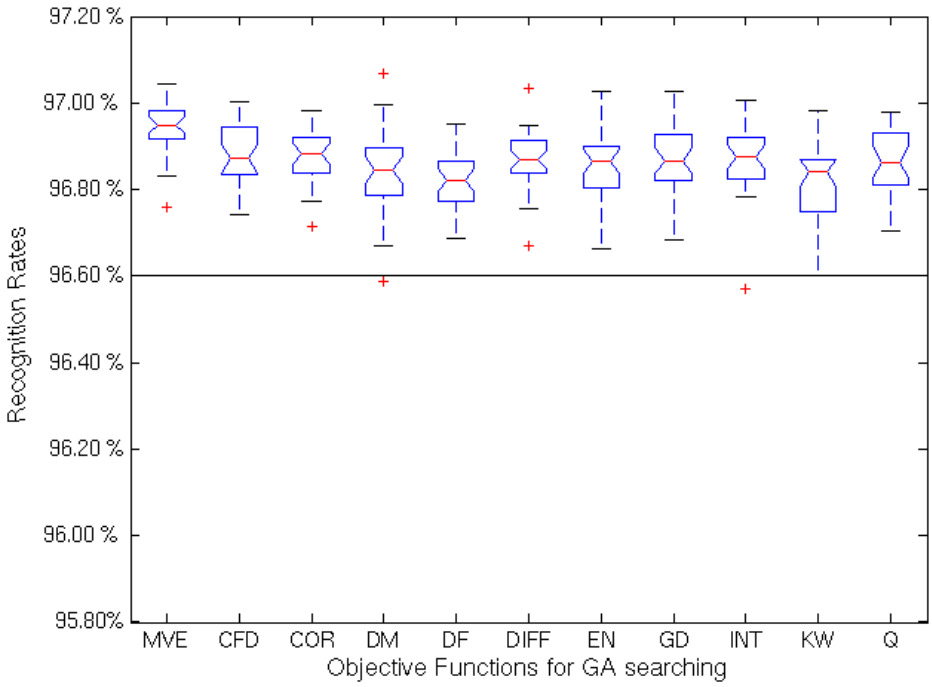

Figure 9 The recognition rates achieved by EoCs selected by 10 compound diversity functions and Majority Voting Error (MVE), using PFM-MAJ as fusion function

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 X). A similar statement can be found in (96), 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. 8 and 9).

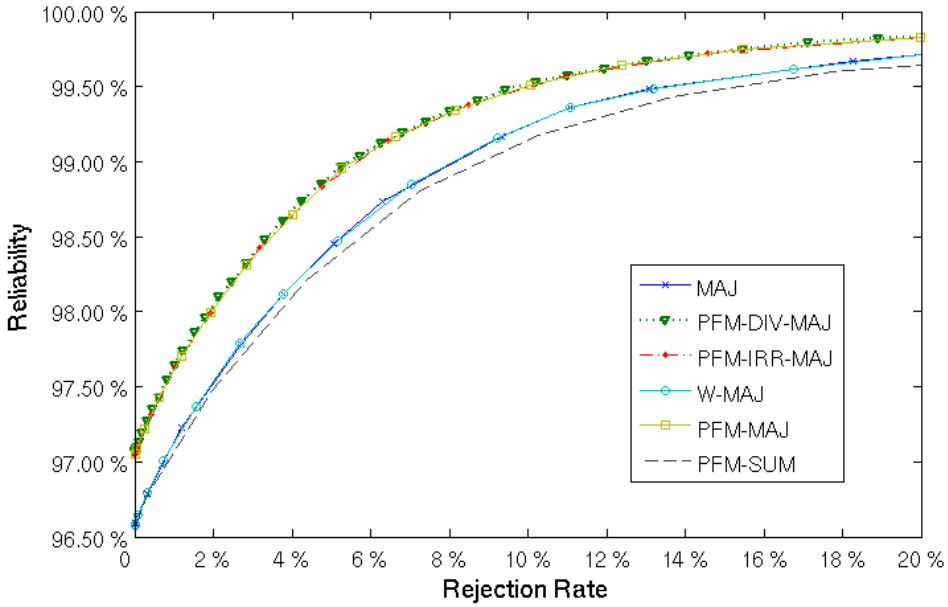

Figure 10 The rejection curve of ensemble of KNNs selected by Majority Voting Error (MVE), with evaluated fusion functions: MAJ, W-MAJ, PFM-SUM, PFM-MAJ, PFM-IRR-MAJ and PFM-DIV-MAJ. The accuracy of the PFM-DIV-MAJ is the mean value of the results applying 10 different diversity measures

We select the six best fusion functions for applying the rejection mechanism. In Figure 10, 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 pairwise fusion matrix, 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 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.

3.6 Discussion

For EoCs, the ideal is to obtain the probability $P(l|c(1), \cdots, c(i), \cdots, c(L), x)$ for the whole data set X, where l is the possible class label, and $c(1), \cdots, c(i), \cdots, c(L)$ are decisions of individual classifiers $f(1), \cdots, f(i), \cdots, f(L)$ respectively. But, in reality, this approach might not work owing to the limitation with respect to the number of samples. Instead of estimating $P(l|c(1), \cdots, c(i), \cdots, c(L), x)$, 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 is not limited by the random classifier selection on the UCI machine learning repository, we apply the ensemble selection scheme with 10 compound diversity functions (58) on the $NIST\ SD19$ database. We can observe that the advantage of using the PFM-MAJ instead of the MAJ is very clear (Fig. 8 & Fig. 9).

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 4-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.

3.7 Conclusion

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

a. It transforms crisp class label outputs into class probability outputs.

b. It is suited to most kinds of existing fusion functions for combining classifiers.

c. It takes into account the interaction of classifiers in a pairwise manner.

d. 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 Naive Bayes or Decision Tem-

plate (69). 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.

So far, we have already proposed a new ensemble selection scheme and a new classifier combination method. But still, we need to look back at one of the most essential element in an EoC, the process ensemble creation. At the next chapter, we propose a new ensemble creation method for an ensemble of HMM classifiers. We then apply different ensemble selection methods and classifier combination schemes, including those proposed in this thesis, and compare their results.

CHAPTER 4

ENSEMBLE OF HMM CLASSIFIERS BASED ON THE CLUSTERING VALIDITY INDEX FOR A HANDWRITTEN NUMERAL RECOGNIZER

A new scheme for the optimization of codebook sizes for HMMs and the generation of HMM ensembles is proposed in this chapter. In a discrete HMM, the vector quantization procedure and the generated codebook are associated with performance degradation. By using a selected clustering validity index, we show that the optimization of HMM codebook size can be selected without training HMM classifiers. Moreover, the proposed scheme yields multiple optimized HMM classifiers, and each individual HMM is based on a different codebook size. By using these to construct an ensemble of HMM classifiers, this scheme can compensate for the degradation of a discrete HMM.

4.1 Introduction

Random Subspace, Bagging and Boosting are general ensemble creation methods, and they can in most cases be applied to all kinds of classification algorithms to generate diverse classifiers for ensembles. However, there are some classification algorithms that might need to use all samples and all features for training, and thus cannot use Random Subspace, Bagging or Boosting for ensemble creation. Fortunately, there are some specialized ensemble creation methods which can be applied to these target classification algorithms. To be successful, these specialized ensemble creation methods must take into account the training process of the target classification algorithm, so that the classifiers created will be diverse enough to construct an ensemble.

One of such classification algorithm is the Hidden Markov Model (HMM). An HMM is one of the most popular classification methods for pattern sequence recognition, especially for speech recognition and handwritten pattern recognition problems (6; 16; 83; 84; 94). The objective of the HMM is to model a series of observable signals, and it is this ability

that makes the HMM a better choice for recognition problems than other classification methods. As a stochastic process, HMM is constructed with a finite number of states and a set of transition functions between two states or over the same state (6; 83; 94). Each state transmits some observations, according to a codebook which sets out corresponding emission probabilities. Such observations may be either discrete symbols or continuous signals. In a discrete HMM, a vector-quantization codebook is typically used to map the continuous input feature vector to the code word.

To perform vector-quantization to generate the codebook of an HMM, we first need to define the size of the codebook. An HMM codebook size optimization is, in general, performed by constructing a number of HMM classifiers and comparing their recognition rates on a validation data set. In other words, the process of codebook size optimization is always problem-dependent. Moreover, given the extremely time-consuming process of HMM training, HMM codebook size optimization remains a major problem.

There are various methods for solving the HMM codebook size optimization problem, the difficulty being to define the "optimal" codebook. On the one hand, according to the "no-free-lunch" theory (105; 106), no search algorithm is capable of always dominating all others on all possible datasets. On the other hand, an optimal codebook is only optimal relative to a few other evaluated codebooks. For these reasons, we believe that it is in our interest to consider multiple optimal codebooks and to use them to construct an ensemble of HMM classifiers (EoHMM), rather than to select a single, supposedly optimal, codebook.

We note that the use of EoHMM has been emerging as a promising scheme for improving HMM performance (3; 32; 33; 36; 34; 35; 38). This is because an EoC is known to be capable of performing better than its best single classifier (11; 22; 63; 89). EoC classifiers can be generated by changing the training set, the input features or the parameters and architecture of the base classifiers(35). There are quite a few methods for creating HMM

classifiers, based on the choice of features (33) for isolated handwritten images, and both column HMM classifiers and row HMM classifiers can be applied to enhance performance (8; 9). The use of various topologies, such as left-right HMM, semi-jump-in, semi-jump-out HMM (36), and circular HMM (3) can also be applied.

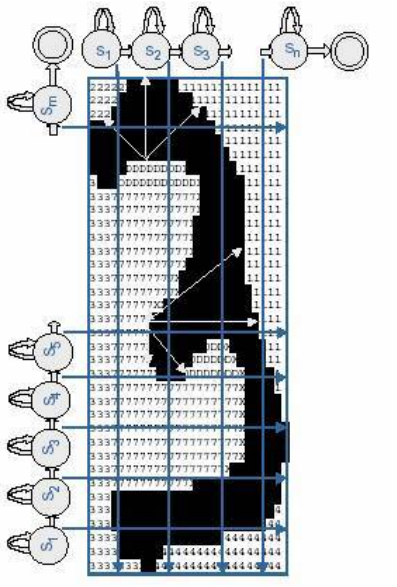

Figure 11 The benchmark HMM classifiers: For any character image, we scan the image from left to right, and obtain a sequence of columns as the observations; we then scan this image again from top to bottom, and obtain a sequence of rows as the observations. By this means, features are extracted from each column and each row, a column HMM classifier and a row HMM classifier are thus constructed for isolated handwritten numeral recognition

In our case, we want to create an EoHMM based on several codebooks. To do this, all the codebooks must be good and diverse, i.e. the symbols (codewords) that these codebooks present must be useful and different. The reason for this is quite simple: in order to obtain different and accurate HMM classifiers, we should avoid those that are identical or under-performing. The main question is, how can we select good and diverse codebook sizes for an EoHMM? In terms of a good size for a codebook, we note that discrete symbols in HMM are usually characterized as quantized vectors in the codebook by clustering, so

the fitness of the codebook is directly related to the fitness of the clustering, for which a number of validity indices have been proposed (4; 46; 45; 71; 78). This means that codebook size can actually be optimized by using clustering validity indices.

Nevertheless, in order for codebook sizes to be diverse, the clustering validity indices used must offer several adequate codebook sizes, and not just only a single optimal one. Because a data set usually consists of multiple levels of granularity (54; 91), if clustering validity indices can give multiple adequate codebook sizes for HMM, and if these HMM classifiers have diverse outputs, then it is possible to construct EoHMMs based on different codebook sizes. This mechanism will give the local optima of a selected clustering validity index. EoHMMs are then selected by various objective functions and combined by different fusion functions. Since EoHMMs are constructed with multiple codebooks, the degradation associated with a single vector quantization procedure can be improved by multiple vector quantization procedures, and by then classifier combination methods.

To clarify, we want to verify two assumptions in this work. Our first assumption is that a clustering validity index might have the property of being able to generate several codebook hypotheses. The second assumption is that the codebook hypotheses generated by one clustering validity index will contain enough diversity to construct a useful ensemble of EoHMMs. In this case, an EoHMM is constructed not based on different feature subspaces or on different samples, but on different representations in several symbol spaces. The key questions that need to be addressed are the following:

a. What are the basic properties of the clustering validity indices used in clustering?

b. Which clustering validity index performs better in the selection of codebook sizes for HMM?

c. Can the clustering validity index offer more than one hypothesis on HMM codebook sizes?

d. For HMM classifiers based on different codebook sizes selected by a clustering validity index, is the diversity among them strong enough to yield an EoHMM which performs well?

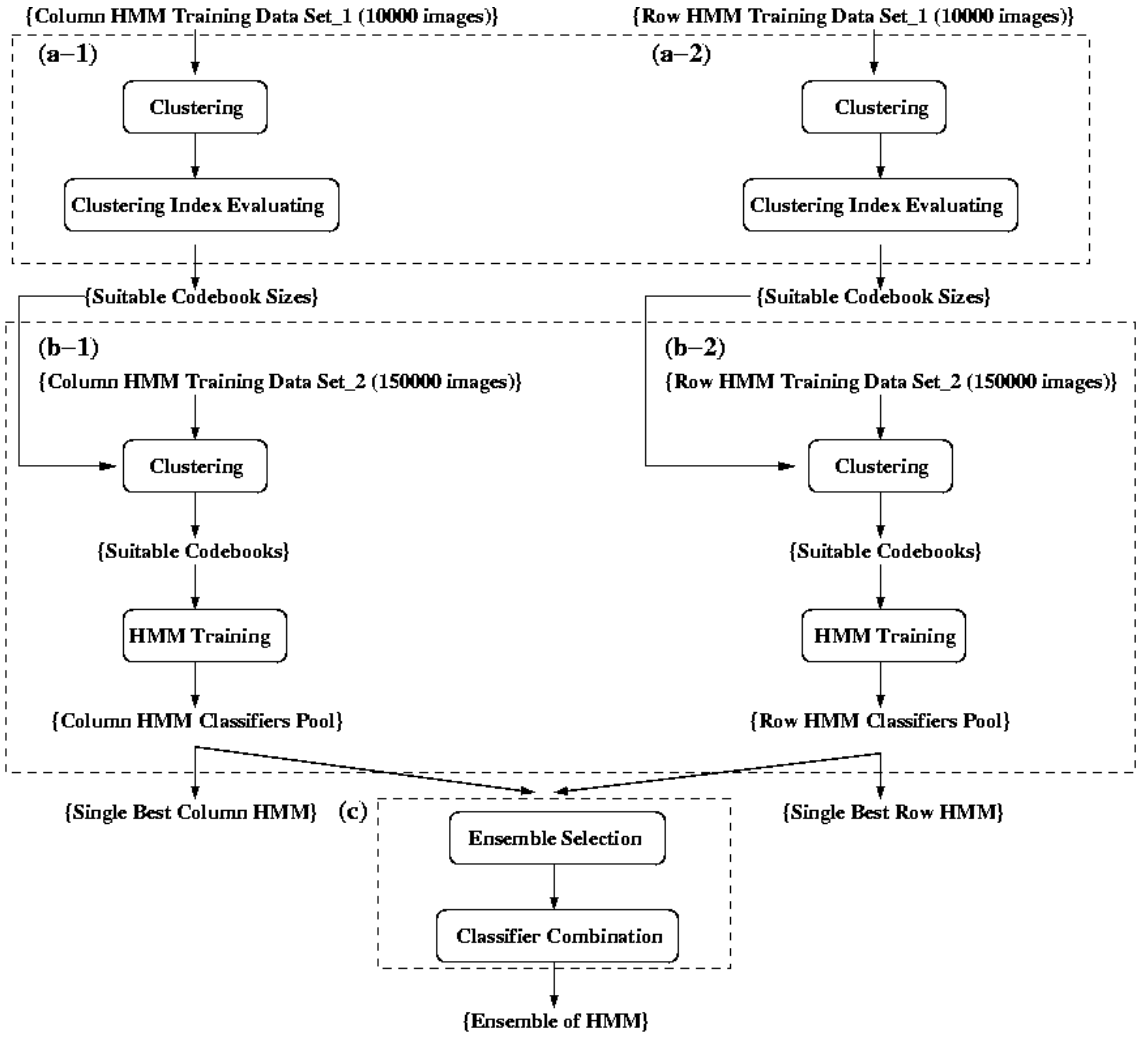

Figure 12 The EoHMM classification system approach includes: (a) the adequate codebook sizes searching; (b) codebooks generation and HMM classifiers training (c) EoHMM selection and combination. Both (a) and (b) were carried out separately on column and row HMM classifiers

To answer these questions, we carried out a general review on clustering validity indices, and applied the selected index for EoHMM construction. We used the HMM-based hand-

written numeral recognizer in (8; 9), which includes the numeral-string segmentation stage and the single-character verification stage. In this chapter, we focus on improving the verification stage to recognize the separated handwritten digits (Fig. 11). At this stage, column and row HMM classifiers are used to enhance classification accuracy, and the sum of the outputs from the single best column HMM and the single best row HMM constitutes the final decision. With this system, we were able to improve verification by constructing an EoHMM with different codebooks on both column HMM classifiers and row HMM classifiers, and then carrying out ensemble selection and classifier combination. It is important to note that HMM optimization is a very complex task, and there are still a great many issues associated with it. The analysis and the method presented therefore constitute only a small step towards a considerably improved understanding of HMM and EoHMM.

The chapter is organized as follows. In the next section, we introduce the basic concepts of clustering validity indices. Section 3 details the process of generation, selection and combination of HMM classifiers. In section 4, we report on experiments we carried out on the NIST SD19 handwritten numeral database. A discussion and a conclusion are presented in the final sections.

4.2 Clustering Validity Indices

In general, an HMM codebook is generated from a vector quantization procedure, and each code word can be actually regarded as a centroid of a cluster in feature space. The fitness of the clustering depends on a number of different factors, such as clustering methods and the number of clusters. For an adequate HMM codebook, there should be a means to select a better clustering. A clustering validity index is thus designed to evaluate the clustering results, and to assign a level of fitness to these results. Three types of clustering validity indices have been proposed in the literature, including external indices, internal indices and relative indices (45; 78). External indices are designed to test whether or not a data set is randomly structured; internal indices are used to evaluate the clustering results

by comparing them with a known partition; and relative indices are designed merely to find the best clustering results, that is, the most natural ones, regardless of sample labels. Given the fact that we have no known partition for a codebook and we are interested in finding natural clusters as code words for HMM, we focus on the known relative indices in this section, present their definitions and discuss their advantages and drawbacks in evaluating clustering. We must mention that a clustering validity index is not a clustering algorithm in and of itself, but a measure to evaluate the results of clustering algorithms and give an indication of a partitioning that best fits a data set. A clustering validity index is independent of clustering algorithms and data sets.

4.2.1 R-squared (RS) index

To explain RS index, we need to explain the $Sum\ of\ Squares(SS)$ measure used in this index. We have three kinds of SS:

a. SS_w: The sum of squares within the cluster.

Given a cluster c_x consisting of n samples, with the members $X_1, \cdots, X_n$, and the cluster center $\bar{X}$, define

$$SS_w(x) = \sum_{j=1}^{n} (X_j - \bar{X})^2 \tag{4.1}$$

and for nc clusters, suppose there are n_i samples for cluster c_i, and denote X_i as the centroid of the cluster c_i, and its members as X_{ij}, the total SS_w can be written as

$$SS_w = \sum_{i=1}^{nc} \sum_{j=1}^{n_i} (X_{ij} - \bar{X}_i)^2 \tag{4.2}$$

b. SS_b: The sum of squares between clusters

Given a data set d_x of nc cluster centroids $c_1, \cdots, c_{nc}$, and the center of all the data

$\bar{C}$, define

$$SS_b = \sum_{i=1}^{nc} (c_i - \bar{C})^2 \tag{4.3}$$

c. SS_t: The total sum of squares

$$SS_t = SS_w + SS_b \tag{4.4}$$

and RS (46; 45) is defined as the ratio of SS_b to SS_t. That is,

$$RS = \frac{SS_b}{SS_t} \tag{4.5}$$

Note that SS_b is a measure of difference between clusters, so that the more separated the two clusters, the greater SS_b will be. Moreover, SS_w is the compact measure of a single cluster. The smaller SS_w, the more compact this cluster will be. Given the same SS_w, RS is proportional to SS_b, and is the measure of distance between clusters. We can also write :

$$RS = \frac{SS_t - SS_w}{SS_t} \tag{4.6}$$

Given the same SS_b, RS can be regarded as a measure of compactness. To combine both effects, RS is a measure of homogeneity between clusters. The value of RS always being between 0 and 1. The process involves drawing the curve of RS while applying different numbers of clusters, and finding its "knee".

Given a number of clusters nc, a single RS takes into account the compactness of all clusters, as well as the distance between them. However, this distance measure is rough and indirect because it is based on the distance with respect to the mean value of all centroids. A single RS is unable to indicate how good the clustering is, but a series of

RS indices can. We expect to see a huge increase in RS value when the best number of clusters nc_{best} is applied (Fig. 17). Nevertheless, if the data are high-dimensional, and if some clusters are on the surface of a hyper-sphere the center of which is closed to the mean of all data, RS might not be very sensitive to them because the SS_b value is little changed.

4.2.2 Root-Mean-Square Standard Deviation (RMSSTD) index

$RMSSTD$ index is a measure based on sample variances and sample means. Supposing we have nc clusters in the data, and cluster c_i has n_i samples, $1 \leq i \leq nc$, then the mean of the cluster c_i is defined as :

$$\bar{X}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} X_j \tag{4.7}$$

where $X_j, 1 \leq j \leq n_i$, are samples of cluster c_i. Moreover, the variance of cluster c_i is defined as :

$$S_i^2 = \frac{1}{n_i - 1} \sum_{j=1}^{n_i} (X_j - \bar{X}_i) \tag{4.8}$$

Similarly, $RMSSTD$ (46; 45) is defined as :

$$RMSSTD = \left(\frac{\sum_{i=1}^{nc} \sum_{j=1}^{n_i} (X_{ij} - \bar{X}_i)}{\sum_{i=1}^{nc} n_i - 1}\right)^{\frac{1}{2}} \tag{4.9}$$

where $n_i, 1 \leq i \leq nc$ is the number of samples of cluster c_i, and $\bar{X}_i$ is the centroid of cluster c_i, $X_{ij}, 1 \leq j \leq n_i$ is a sample belonging to cluster c_i. From this, it is clear that $RMSSTD$ decreases when the number of clusters increases, because the more clusters it has, the smaller the variance will be for each cluster.

Like RS, the best clustering can be located on the "knee" of $RMSSTD$ curve (Fig. 16). However, there is a more serious problem with $RMSSTD$, in that it does not take into

account the distance between clusters, relying totally on SS_w and the number of clusters nc. This makes $RMSSTD$ less likely to detect the best number of clusters.

4.2.3 Dunn's Index

Assuming that the clustering process generates nc clusters, and that, for all clusters $c_1, \cdots, c_{nc}$, we define the dissimilarity of two clusters c_i, c_j, where $1 \leq i, j \leq nc, i \neq j$ as :

$$d(c_i, c_j) = \min_{x \in c_i, y \in c_j} d(x, y) \tag{4.10}$$

where x and y are any points in cluster c_i and c_j respectively, and $d(x, y)$ is the distance between x and y. We also define the diameter of a cluster c_i as :

$$diam(c_i) = \max_{x, y \in c_i} d(x, y) \tag{4.11}$$

Then, Dunn's index (4; 46; 45; 71; 78) is defined as :

$$Dunn's = \min_{i=1,\cdots,nc} \{ \min_{j=i+1,\cdots,nc} (\frac{d(c_i, c_j)}{\max_{k=1,\cdots,nc} \cdot diam(c_k)}) \} \tag{4.12}$$

It is clear that the larger $Dunn's$ index, the better the clustering results will be. The maximum of diameter $diam(c_k)$ might be larger than the dissimilarity $d(c_i, c_j)$. However, $Dunn's$ index is not an statistical clustering validity index. Given three clusters c_i, c_j, c_k, where $d(c_i, c_j)$ is defined by c_i, c_j and $diam(c_k)$ is defined by c_k, and $p(k \in (i \cup j)) \neq 0$, it is evident that $Dunn's$ index considers the distribution of none of the other clusters, with only the two following constraints:

a. For any other cluster $c_l, 1 \leq l \leq nc, l \neq k$, $diam(c_l) \leq diam(c_k)$.

b. For any other cluster pairs $c_m, c_n, 1 \leq m, n \leq nc, (m \cup n) \neq (i \cup j)$, $d(c_m, c_n) \geq d(c_i, c_j)$.

Considering the distribution of clusters c_i, c_j, c_k, suppose that the $diam(c_k)$ is defined by two points k_{d1}, k_{d2} in c_k. Given the condition that they do not generate diameters larger than $diam(c_k)$, then all the other points in c_k can change their position. A similar situation can be observed in c_i, c_j. Supposing that $d(c_i, c_j)$ is defined by a point i_d in c_i and another point j_d in c_j, then none of the other points in c_i, c_j are considered by $Dunn's$ index, on the condition that their distance is no shorter than $d(c_i, c_j)$. Another disadvantage of $Dunn's$ index is that, by measuring $d(c_i, c_j)$ and $diam(c_k)$, it actually requires calculation of the distance between any two data points. If the data set is large, the calculation of $Dunn's$ index will be highly complex and could be very time-consuming.

4.2.4 Xie-Beni (XB) index

XB index (4; 46; 45; 78) was originally a fuzzy clustering validity index. For a fuzzy clustering scheme, suppose we have the data set $X = \{x_i, 1 \leq i \leq N\}$, where N is the number of samples and the centroids v_j of clusters $c_j, 1 \leq j \leq nc$, where nc is the total number of clusters. We seek to define the matrix of membership $U = u_{ij}$, where u_{ij} denotes the degree of membership of the sample x_i in the cluster c_j. To define the XB index, first one must define the sum of squared errors for fuzzy clustering. The sum of squared errors is defined as

$$J_m(U, V) = \sum_{i=1}^{N} \sum_{j=1}^{nc} (u_{ij})^m \|x_i - v_j\|^2 \tag{4.13}$$

where $1 \leq m \leq \infty$. In general, we use J_1 for the calculation. U is a partition matrix of fuzzy membership $U = u_{ij}$, and V is the set of cluster centroids $V = v_i$. In addition, the minimum inter cluster distance d_{min} must also be defined, as

$$d_{min} = min_{i,j}\|v_i - v_j\| \tag{4.14}$$

Supposing that we have N samples on the total data, XB index can be defined as

$$XB = \frac{J_m}{N \times (d_{min})^2} \tag{4.15}$$

XB index is designed to measure the fitness of fuzzy clustering, but it is also suitable for crisp clustering. The XB index has been mathematically justified in (108). We note that, while u_{ij} is a 0 or 1 parameter, J_1 is exactly the same SS_w used in the RS and $RMSSTD$ indices. But, unlike these two indices, the XB index takes into account the total number of samples N. This does not normalize the XB index, but it does help to limit the increase in the index when the number of samples changes incrementally. We can also observe that the XB index uses the minimum distance d_{min} between the centroids of all cluster pairs, even though it is different from the distance $\min d_{c_i,c_j}$ used in $Dunn's$ index. The difference between d_{min} and $\min d_{c_i,c_j}$ could be regarded as the sum of the variances of cluster c_i and cluster c_j. From this point of view, we can say that the XB index is somehow a hybrid of the $RMSSTD$ index and $Dunn's$ index. The lower the value of the XB index, the better the clustering should be.

However, once XB index finds the nearest cluster pairs, it ignores the distribution of other clusters, on condition that the distances between any two of them are not less than d_{min} and all clusters maintain the same SS_w. The XB index has some advantages. It is a minimum-value-preferred index, and consequently we do not need to find the "knee", as in the RS or $RMSSTD$ indices. Moreover, unlike $Dunn's$ index, the XB index does

not evaluate the distance between any two data points, but rather the distance between any two clusters, and thus is much less complex than than $Dunn's$ index. This makes the XB index a better choice than $Dunn's$ index or the $RMSSTD$ and RS indices.

4.2.5 PBM index

Like the XB index, the PBM index (78) is suitable for both fuzzy clustering and crisp clustering. Supposing that we have a data set with N samples $X = \{x_1, \cdots, x_N\}$, and nc clusters $c_i, 1 \leq i \leq nc$, with respect centroids $v_i, 1 \leq i \leq nc$ and a given a matrix of membership $U = \{u_{ij}\}$ to denote the degree of membership of the sample x_i in the cluster c_j, we define the measure of within-cluster scatter E_{nc} as :

$$E_{nc} = \sum_{i=1}^{nc} \sum_{j=1}^{n_i} u_{ij} \|x_j - v_i\| \tag{4.16}$$

Then we define the inter-cluster measure D_{nc} as :

$$D_{nc} = \max_{i,j}^{nc} \|v_i - v_j\| \tag{4.17}$$

The final PBM index is thus defined by :

$$PBM = (\frac{1}{nc} \times \frac{E_1}{E_{nc}} \times D_{nc})^2 \tag{4.18}$$

where E_1 is a constant for a given data set, we could simply set E_1 equal to 1. What we notice first is that, as with most of the other indices, the PBM index uses the within-cluster measure. In fact, E_{nc} is equal to J_1 in the XB index, equal to SS_w in the RS and $RMSSTD$ indices. Considering the influence of the number of clusters, the PBM index takes a step which is similar to that taken with the $RMSSTD$ index, which is to divide the value $\frac{D_{nc}}{E_{nc}}$ by the number of clusters nc.

However, unlike $Dunn's$ index and the XB index, the PBM index uses the maximum distance between centroids of all the cluster pairs. As a result, the larger the PBM index, the more compact each cluster will be. But, the use of the maximum distance between centroids is less relevant to clustering fitness. The main problem of clustering is to make sure that two closest clusters are well separated, and not that the two clusters furthest apart can be further separated.

4.2.6 Davies-Bouldin (DB) index

The Davies-Bouldin (DB) index (4; 46; 45; 71; 78) is a function of the ratio of the sum of within-cluster scatter to between-cluster separation. The scatter within the i_{th} cluster is computed as :

$$S_{i,q} = (\frac{1}{|C_i|} \sum_{x \in C_i} \{\|x - z_i\|^q\})^{\frac{1}{q}} \tag{4.19}$$

where $|C_i|$ is the number of samples belonging to cluster C_i, and z_i is the centroid of cluster C_i. Usually, we use $q = 2$ for the DB index, and the distance between cluster C_i and C_j is defined as :

$$d_{ij,t} = (\sum_{s=1}^{p} \|z_{is} - z_{js}\|^t)^{\frac{1}{t}} = \|z_i - z_j\| \tag{4.20}$$

where $S_{i,q}$ is the q_{th} root of the q_{th} moment of the points in cluster i with respect to their mean, and is a measure of the dispersion of the points in cluster i. $S_{i,q}$ is the average Euclidean distance of the vectors in class i from the centroid of class i. $d_{ij,t}$ is the Minkowski distance of order t between the centroids that characterize clusters i and j. p is the dimension of features, and, in general, $t = 2$ is used for $d_{ij,t}$. Subsequently, the measurement

based on the ratio of within-cluster scatter to between-cluster separation can be obtained :

$$R_{i,qt} = \max_{j,j\neq i}\{\frac{S_{i,q} + S_{j,q}}{d_{ij,t}}\} \tag{4.21}$$

The Davies-Bouldin index is then defined as :

$$DB = \frac{1}{K}\sum_{i=1}^{K} R_{i,qt} \tag{4.22}$$

where K is the number of clusters. In practice, we set $q = 1$, $t = 1$, so that :

$$S_i = (\frac{1}{|C_i|}\sum_{x\in C_i}\{\|x - z_i\|_2\}) \tag{4.23}$$

$$d_{ij} = (\|z_i - z_j\|) = \|z_i - z_j\| \tag{4.24}$$

$$R_i = \max_{j,j\neq i}\{\frac{S_i + S_j}{d_{ij}}\} \tag{4.25}$$

It is clear that S_i is the average of Euclidean distance of the vector X in cluster C_i with respect to its centroid z_i, and d_{ij} is the Euclidean distance between the centroids z_i and z_j of the clusters C_i and C_j respectively. The smaller the DB index, the better the clustering is supposed to be. Thus, the DB index has two elements: the compactness of each cluster pairs $S_{i,q}+S_{j,q}$, and the inter-cluster scatter d_{ij}. Moreover, the DB index is somehow more significant than all the other indices we have mentioned previously, for the following three reasons:

a. The measure of compactness of each cluster pairs $S_{i,q} + S_{j,q}$ is similar to the sum of the within-cluster scatter used in other indices. But, instead of summing them all, this index tackles each cluster pairs separately and is more adequate.

b. The use of the inter-cluster scatter d_{ij} for each single cluster, instead of just the minimum or maximum functions, makes this index more sensitive.

c. The compactness of each cluster pairs $S_{i,q} + S_{j,q}$ and the inter-cluster scatter d_{ij} are combined by calculating their ratio. This feature gives the index a significant capacity to find a point of compromise between two clustering criteria: the distance between clusters of different clusters and the compactness of single clusters.

Based on these properties, we can say that the DB index should perform better than all other indices mentioned previously. However, the DB index does have its own drawbacks, a potential problem being that it uses the operations such as $R_{i,qt}$, the maximum of all cluster pairs for a certain cluster, and $\sum_{i=1}^{K} R_{i,qt}$, the summation of all the maximum values obtained on all clusters, to take into account all clusters separation. However, this process allows just one extremely bad cluster separation to overwhelm all the other good cluster separations. Other than this, the calculation of the DB index on single cluster pairs is convincing.

4.2.7 clustering validity index for Codebook Size Selection

Among the above clustering validity indices, Dunn's index, the DB index and the XB index are considered as the most adequate ones. However, the drawback of Dunn's index is its high calculation complexity. The derivation of the DB index has convincing theoretical support, but its problem is that it sums all the maximum values obtained on all clusters, which means that one extremely bad cluster separation may overwhelm all the other good cluster separations. In contrast, the XB index uses only the minimum distance between

centroids of cluster pairs, focusing on the nearest cluster pairs and ignores the distribution of other clusters.

However, to obtain a group of potentially adequate codebook sizes, the applied clustering validity index is not only supposed to find a single best number of clusters, but also several best numbers of clusters. In other words, the clustering validity index used must have several optima that can depict a data set at multiple levels of granularity (54; 91). This property is important because the best number of clusters depends on different hierarchical levels. An adequate clustering validity index should not only offer different clusterings, but also a reasonable distinction among them. When HMM classifiers are trained with the same features and with the same samples, the distinction among the codebooks is the only possibility that results in diversity among classifiers and boosts the EoHMM performance.

The XB index is found to have this desirable property in our problem (Fig. 13). The plot of XB index values versus the numbers of clusters gives a lot of minima with XB index values smaller than those of their neighbours, and these are actual optima for codebook sizes and are thus adequate for the construction of an EoHMM. In the next section, we detail the process for construction of EoHMMs based the on XB index, and the ensemble selection and classifier combination schemes considered.

4.2.8 Generation of HMM classifiers

Given a data set of $X = \{x_i, 1 \leq i \leq N\}$, where N is the number of samples, and defining a possible range M for the numbers of clusters j, $1 \leq j \leq M$, the cluster index should give the fitness $F_t(j)$ for these M clusterings, with $1 \leq j \leq M$. Due to the tremendous size of data set, we can use a smaller data set with N_s samples extracted from N samples for clustering goodness evaluation, $N_s = \eta\dot{N}$, where η is the proportion of samples used. Assuming that we intend to select L best clusterings, then these clusterings could be selected with clustering validity index values $F_t(j)$, $1 \leq j \leq L$. These selected numbers of clusterings then serve as the sizes of the codebook of HMM classifiers. The

selected codebook sizes are used again for the clustering on all N samples, with the result that the respective codebooks are generated. Each HMM is then trained with a different codebook. This pool of HMM classifiers must go further through the ensemble selection process to decide which classifiers are adequate for construction of an ensemble. Then the selected classifiers would be combined according to a fusion function.

Given the various scheme of objective functions for ensemble selection and the fusion functions for classifier combination, it is of the great interest to test these schemes on real problem. We perform the experiment on a benchmark data base in the next section.

4.3 Experiments with EoHMMs

The experimental data was extracted from $NIST\ SD19$ as a 10-class handwritten numeral recognition problem. As a result, there is an HMM model for each class, and 10 HMM models for an HMM classifier. Five databases were used: the training set with 150000 samples ($hsf\_\{0-3\}$) was used to create 40 HMM classifiers, 20 of them being column HMM classifiers and other 20 being row HMM classifiers. The large size of the data set for training can lead to a better recognition rate for each individual classifier. For codebook size selection evaluated by clustering validity indices, due to the extremely large data set (150000 images are equivalent to 5048907 columns and 6825152 rows, with 47 features per column or per row), we use only the first 10000 images from the training data set to evaluate the quality of the clustering, and they are equal to 342910 columns and 461146 rows. The sampling may present a slight bias in clustering, but, because even the sampled data set contains 0.34 millions column samples and 0.46 millions row samples, we believe it is large enough to evaluate the quality of the clustering and discover the multiple-level granularity of the data set. Note that, at the clustering evaluation stage, we only examined the different numbers of clusters with the clustering validity index to select several suitable codebook sizes for an EoHMM. Then, the codebooks were generated with the whole training set, according to the previously selected codebook sizes. The training validation

set of 15000 samples was used to stop HMM classifiers training once the optimum had been achieved. The optimization set containing 15000 samples ($hsf\_\{0-3\}$) was used for GA searching for ensemble selection. To avoid overfitting during GA searching, the selection set containing 15000 samples ($hsf\_\{0-3\}$) was used to select the best solution from the current population according to the defined objective function and then to store it in a separate archive after each generation. The selection set is also used for the final validation of HMM classifiers. Using the best solution from this archive, the test set containing 60089 samples ($hsf\_\{7\}$) was used to evaluate the accuracies of EoC.

Each column HMM used 47 features obtained from each column, and each row used 47 features obtained from each row (See Fig. 11). The features were extracted by the same means described in (8; 9), and K-Means was used for vector-quantization to generate codebooks for the HMM. The number of HMM states was optimized by the method described in (102). The HMMs were trained by Baum-Welch algorithm (83; 84). The benchmark HMM classifiers used 47 features, with the codebook size of 256 clusters (8; 9). For benchmark column HMM, we have a recognition rate of 97.60%, and for benchmark row HMM the classification accuracy was about 96.76%, while the combination of the benchmark column HMM and the benchmark row HMM achieved a rate of 98.00%. 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 the pool to all samples. The oracle achieved a rate of 99.76% on the test set, considering the pool of the whole HMM classifiers. For combining classifiers, 12 different fusion functions were tested.

4.3.1 Behaviors of clustering validity indices in HMM features

To decide on suitable codebook sizes of HMM, we carried out clusterings on HMM features. Due to the large data size, it is clear that we could not use all the training set to do the clusterings, all with different numbers of clusters. As a result, the first 10000 images in

training set were used for clustering, these images containing 342910 columns and 461146 rows.

Before we constructed the EoHMM, we performed K-Means clusterings with different numbers of clusters on HMM features, and showed the properties of clustering validity indices in this problem. Processing clusterings from 3 clusters to 2048 clusters for the clustering task, we showed the relationship between the XB index and the number of clusters for column HMM features, and many minima can be observed (Fig. 13(a)). The optimum codebook size defined by the XB index value is 1893 clusters, and, with this codebook size, the column HMM classifier can achieve 98.92% recognition rate on the validation set, and 98.32% on the test set. A similar tendency can be observed in row HMM features (Fig. 13(b)). This property, as we argued, is important to get multiple levels of granularity of the data set, and it offers codebook sizes for HMMs with the potential to perform well.

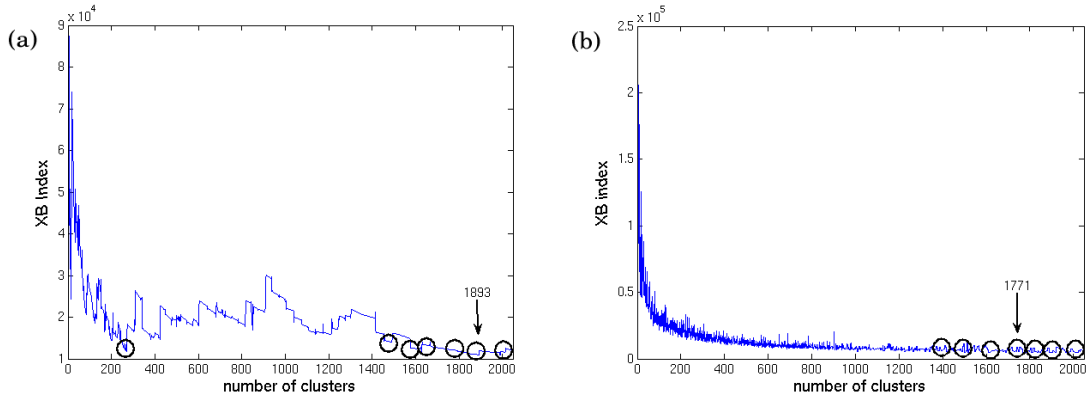

Figure 13 The relationship between XB index and the number of clusters for: (a) HMM column features; (b) HMM row features. The circled areas indicate the places where the best 40 optima were found. The arrow indicates the smallest XB value with the respective number of clusters. Note that clusterings were carried out on the first 10000 images of the training data set. (See Table XI for details)

In contrast, the relationship between the DB index and the number of clusters was much more ambiguous. In general, for column HMM features, the curve reached its minimum at 5 and maximum at 132, then decreased almost constantly (Fig. 14(a)). Apparently, a simple 5-cluster optimum is not useful for the codebook, as the corresponding column HMM can achieve a classification accuracy of only 71.69% on validation set, and 69.43% on the test set. Moreover, most of the optima selected by the DB index will contain fewer than 132 clusters.

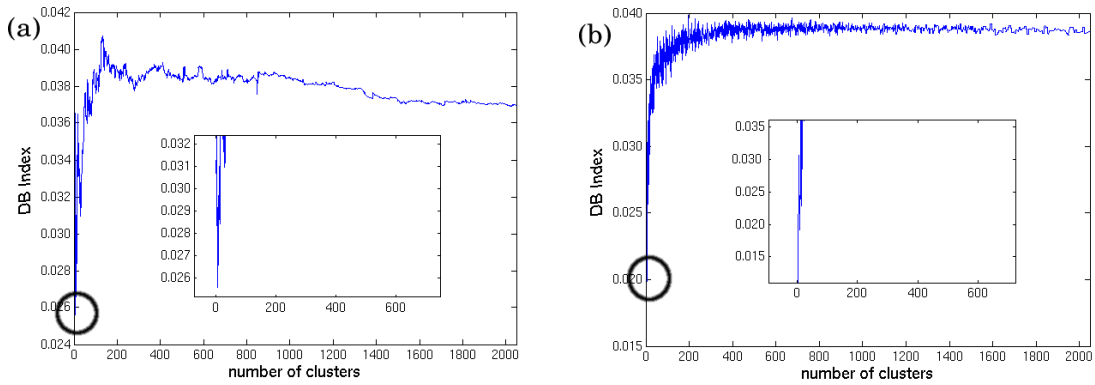

Figure 14 The relationship between DB index and the number of clusters for: (a) HMM column features; (b) HMM row features. Optima are minima in DB index, we enlarge the part where the optimum is located. Note that clusterings were carried out on the first 10000 images of the training data set

As we stated previously, the PBM index is less convincing. The PBM index suggests that the best clustering is with 3 clusters for column HMM (Fig. 15(a)), which can achieve a recognition rate of only 63.49% on the validation set, and 61.72% on the test set. Note that the maximum value in PBM represents the optimum. After slight variation in the beginning, the curve decreases continuously. The PBM thus encourages the use of small codebook sizes, which cannot lead to any useful results for this problem.

For RS and RMSSTD, the optima are located on the knees of the curves, but it might not be easy to find out these knees. For RS, the so called knee might be found at roughly 140

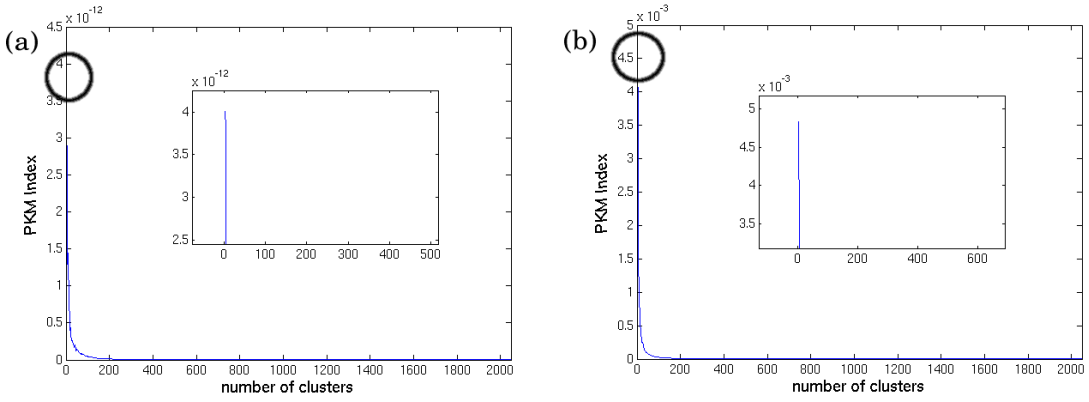

Figure 15 The relationship between PBM index and the number of clusters for: (a) HMM column features; (b) HMM row features. The optimum has the maximum value in PBM index, we enlarge the part where the optimum is located. Note that clusterings were carried out on the first 10000 images of the training data set

clusters for column HMM (Fig. 17(a)), where column HMM achieved 98.14% recognition rate on the validation set, and 97.36% on the test set. For RMSSTD, the knee is roughly at 131 clusters for column HMM (Fig. 16(a)), with which column HMM can achieved a 98.08% classification accuracy on the validation set, and 97.12% on the test set. But the disadvantage common to the RS and RMSSTD indices is that they give only one optimum solution, and there is no way to find multiple optima, which makes it impossible to use them for the construction of an EoHMM. Finally, we must mention that, given the size of the data set, it is impossible to evaluate $Dunn's$ index, because $Dunn's$ index has to calculate the distances between 342910^2 sample pairs for column HMM and 461146^2 sample pairs for row HMM.

4.3.2 The Multiple Levels of Granularity in Codebook Size Selection

These observations indicate that XB index has the properties desired for HMM codebook size selection. Note that, in order to construct an EoHMM which performs well, we need to select several fit codebooks, and, moreover, these codebooks must lead to diverse HMM

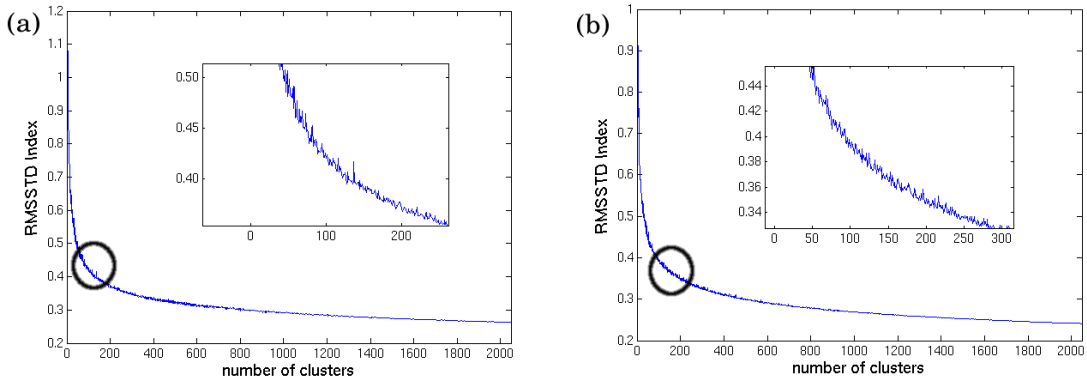

Figure 16 The relationship between RMSSTD index and the number of clusters for: (a) HMM column features; (b) HMM row features. The optimum is located on the "knee" of the curve in RMSSTD index, we enlarge the part where the optimum is located. Note that clusterings were carried out on the first 10000 images of the training data set

classifiers so that the combination of these HMM classifiers can actually achieve even better performance. As we observed in the previous section, the XB index not only finds fit codebooks, but it also reveals the multiple granularity of the data set. Moreover, its calculation is much less time-consuming than Dunn's index. All these advantages favour the use of the XB index.

Intuitively, because the clusterings with different granularity levels are located in different neighbourhoods, it is very unlikely that the codebook size optima found in a single neighbourhood can represent the concept of the multiple-level granularity. For this reason, it is important to have clusterings in different neighbourhoods. To satisfy this condition, we may simply require the selected clusterings have non-adjacent numbers of clusters.

Although the multiple-level granularity implicates the diversity related to different partitions between clusterings, we still need to confirm that the concept of the multiple-level granularity can also lead to better EoHMM performance, i.e., the optima found in different neighbourhoods can lead to better EoHMM performance than those found in the same

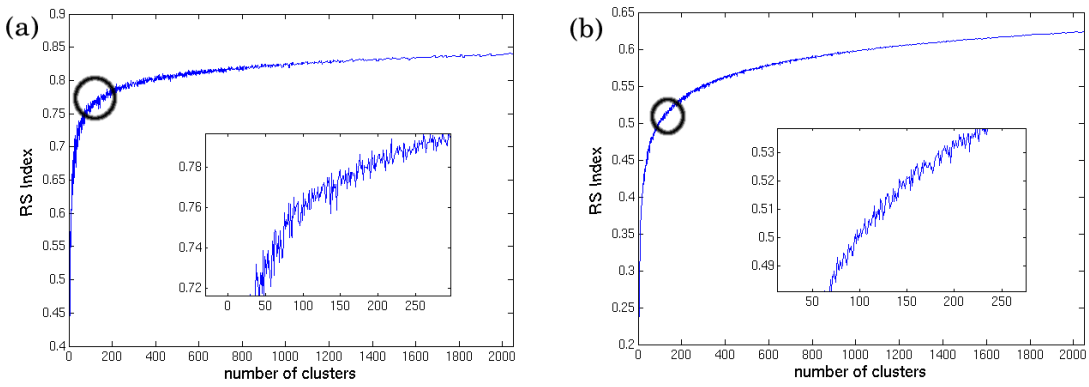

Figure 17 The relationship between RS index and the number of clusters for: (a) HMM column features; (b) HMM row features. The optimum is located on the "knee" of the curve in RS index, we enlarge the part where the optimum is located. Note that clusterings were carried out on the first 10000 images of the training data set

neighbourhood. Thus, we investigated and compared the performances of EoHMMs constructed by codebook sizes selected by the XB index optima in the same neighbourhood and those in different neighbourhoods.

We performed clusterings on the first 10000 images in the training set with numbers of clusters from $3 \sim 2048$. For HMM column features, the best codebook sizes defined by the XB index were 1893, 1892, 1891, 1890 and 1889 clusters, with an XB index of 10943, 10949, 10955, 10961, 10967 respectively. Note that these optima were selected by absolute minima in the XB index, and no multiple levels of granularity were involved. Consequently all selected codebook sizes are in the same neighbourhood.

However, if we require that all optima have an XB index value smaller than those of their neighbours, i.e., if we require simply that for any selected number of cluster $nc \geq 2$, its XB value $XB(nc)$ must be smaller than those of its two nearest neighbours, $XB(nc) < XB(nc+1)$, $XB(nc) < XB(nc-1)$, then we can obtain codebook sizes in different neighbourhoods. Under this condition, the clusterings with 1892, 1891, 1890 and 1889

clusters do not qualify. In contrast, we will have the following best codebook sizes, as defined by the XB index: 1893, 1991, 1986, 1965 and 2012 clusters, with an XB index of 10943, 10982, 11478, 11498 and 11605 respectively. Note that, in this case, the optima were selected by relative minima in XB index, i.e. we required that these minima be the smallest in their neighbourhoods, and thus we took into account of multiple levels of granularity.

The same process was carried out for HMM row features, and the best codebook sizes defined by the XB index were 1771, 1770, 1769, 1768 and 1767 clusters, with an XB index as 4565, 4569, 4572, 4574 and 4577 respectively. If we require that all optima have an XB index value smaller than those of their neighbours, we will have the following best codebook sizes, as defined by the XB index: 1771, 1809, 2022, 1975 and 1978 clusters, with an XB index of 4565, 4675, 4741, 4764 and 4782 respectively.

We then construct 2 basic EoHMMs on both the column HMM features and the row HMM features. One EoHMM was constructed with codebook sizes with XB indices that are the absolute minima, while another EoHMM was constructed with codebook sizes with XB index values that are relative minima, i.e., their XB indices are smaller than their neighbours. We then evaluated the performance of these two EoHMM on both the column HMM feature and the row HMM feature.

Table XI

Comparison classification accuracy with ensembles composed of 5 absolute optima (ABS) and of 5 relative optima (REL) in terms of XB index. Results are shown on test set and validation set. The number of classifiers is shown in parenthesis

	COL-ABS (5)	COL-REL (5)	ROW-ABS (5)	ROW-REL (5)
Validation Set	99.12 %	**99.13 %**	98.80 %	**98.88 %**
Test Set	98.49 %	**98.54 %**	97.92 %	**98.14 %**

Even though the ensembles are constructed with a small number of classifiers, we can observe that optima found in different neighbourhoods by XB index are slightly better than those found in the same neighbourhoods (Table XI). Note that all HMM classifiers are trained with the same number of samples and the whole feature set, and they are different from one another only in the codebooks. We can expect that the difference will be more apparent when more HMM classifiers are used. To prove that an EoHMM constructed with optima found in different neighbourhoods by the XB index can significantly enhance the performance, we went on constructing 20-column HMM classifiers and 20-row HMM classifiers with optima in different neighbourhoods (see below). These HMM classifiers will later be combined and the improvement be measured.

4.3.3 Optimum Codebooks Selected by XB Index

For all clusterings from 3 clusters to 2048 clusters on the first 10000 images in the training set, the 20 smallest minima with XB index values smaller than those of their neighbours were selected as the adequate numbers of clusters, i.e. the 20 most pertinent sizes of codebooks. Once the optimum codebook sizes were selected, we performed clusterings on the whole training data (including 150000 images) with the selected numbers of clusters to generate HMM codebooks. These codebooks were then used for HMM sequence observations and HMM classifier training. This process was carried out for the column features as well as for the row features, all HMM classifiers being trained with the whole feature set and all the training samples. Thus, 20-column HMM classifiers and 20-row HMM classifiers were generated, for a total of 40 HMM classifiers (Table XII).

The best single column HMM achieved a classification accuracy of 98.42% with a codebook size of 1965, which is 0.82% better than the benchmark column HMM classifier; and the best row HMM classifier had a recognition rate of 97.97%, with a codebook size of 1786, which is 1.21% better than the benchmark row HMM. Compared with the benchmark column HMM classifier (97.60%) and with the benchmark row HMM classifier

Table XII

Classification accuracies of 20 column HMM classifiers and 20 row HMM classifiers generated by different codebook sizes on test data set. CCS: Column Codebook Size; RCS: Row Codebook Size; CA: Classification Accuracy. The codebook sizes are ranked by their XB index from left to right

CCS	1893	1991	1986	1965	2012	1934	1796	1998	1627	269
CA	98.32 %	98.33 %	98.35 %	98.40 %	98.30 %	98.39 %	98.34 %	98.33 %	98.33 %	97.56 %
CCS	2040	264	2048	1625	1715	1665	1667	1491	1488	1456
CA	**98.42 %**	97.55 %	98.35 %	98.37 %	98.37 %	98.34 %	98.32 %	98.29 %	98.29 %	98.30 %
RCS	1771	1809	2022	1975	1978	1786	1657	1897	1851	1694
CA	97.84 %	97.88 %	97.93 %	97.73 %	97.95 %	**97.97 %**	97.83 %	97.86 %	97.93 %	97.89 %
RCS	1904	1505	1503	1920	1616	1520	1517	1835	1421	1490
CA	97.83 %	97.84 %	97.80 %	97.83 %	97.89 %	97.84 %	97.75 %	97.90 %	97.70 %	97.73 %

(96.76%), codebooks selected by the XB index gave some improvement to performance. Note that performance is not necessarily proportional to the size of the codebooks. Based on these HMM classifiers, we then construct the EoHMMs.

4.3.4 Column-EoHMM and Row-EoHMM

Without carrying out any ensemble selection process, we simply constructed three ensembles composed entirely of column HMM classifiers (COL-HMM), entirely of row HMM classifiers (ROW HMM) and of all HMM classifiers (ALL-HMM) (Table XIII). The ensembles were then combined by the SUM rule (56; 109; 111) and PCM-MAJ rule (59), since these two fusion functions have been shown to be very effective (56; 59). We note that the ensemble of column HMM classifiers improved by 0.14% over the single best column HMM classifier using the PCM-MAJ fusion function, while the ensemble of row HMM classifiers improved by 0.29% over the single best row HMM classifier using the SUM fusion function. This means that by using different codebook sizes to construct an EoHMM, we explored the diversity of different codebooks of HMM and achieve a better result. Moreover, the ensemble of all HMM classifiers gave the best performance, given

that the obvious diversity between the column HMM classifiers and the row HMM classifiers. With the PCM-MAJ rule, ALL-HMM performed 0.42% better than the single best HMM classifier, and achieved the best classification accuracy.

Table XIII

Comparison of classification accuracies on test data set with two different fusion functions and on different types of EoHMMs. The number of classifiers is shown in parenthesis

Fusion Functions → / EoHMM ↓	PCM-MAJ	SUM
COL-HMM (20)	**98.56 %**	98.55 %
ROW-HMM (20)	98.20 %	**98.26 %**
ALL-HMM (40)	**98.84 %**	98.78 %

4.3.5 Ensemble Selection

For evaluating classifier combinations, another approach is to go through the process of ensemble selection, because one of the most important requirements of EoCs is the presence of diverse classifiers in an ensemble. We tested the simple majority voting error (MVE) and the SUM rule, because of their reputation for being two of the best objective functions for selecting classifiers for ensembles (89). We also tested 10 different compound diversity functions (CDFs) (58), which combine the pairwise diversity measures with individual classifier performance to estimate ensemble accuracy, but do not use the global performance of the EoC. CDFs have been shown to be better than traditional diversity functions for ensemble selection (58).

These objective functions were evaluated by genetic algorithm (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 with the best combina-

Table XIV

Best Performances from 30 GA replications on the test data set. The numbers of classifiers are noted in parenthesis. The SUM was used as the fusion function in EoC

Recognizers	Column HMM classifiers	Row HMM classifiers	Column & Row HMM classifiers
Benchmark	97.60 % (1 / -)	96.76 % (1 / -)	98.00 % (2 / SUM)
XB Selection	98.40 % (1 / -)	97.97 % (1 / -)	98.70 % (2 / SUM)
Classifier Pool	98.55 % (20 / SUM)	98.26 % (20 / SUM)	98.78 % (40 / SUM)
EoHMM Selection	-	-	**98.80 %** (16 / SUM)

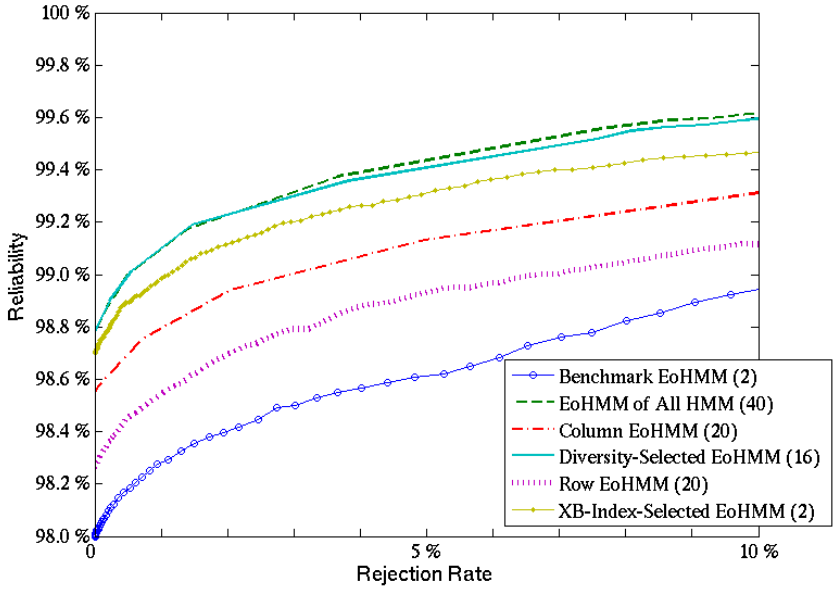

Figure 18 The Rejection mechanism with the SUM rule

tion, it can potentially be exploited to prevent generalization problems (89). GA was set up with 128 individuals in the population and with 500 generations, which means $64,000$ ensembles were evaluated in each experiment. The mutation probability was set to 0.01, and the crossover probability to 50%. With 12 different objective functions (MVE, SUM, 10 compound diversity functions, including the disagreement measure (CDF-DM), the double-fault (CDF-DF), Kohavi-Wolpert variance (CDF-KW), the interrater agreement

(CDF-INT), the entropy measure (CDF-EN), the difficulty measure (CDF-DIFF), generalized diversity (CDF-GD), coincident failure diversity (CDF-CFD), Q-statistics (CDF-Q), and the correlation coefficient (CDF-COR) (58)), and with 30 replications, 23.04 million ensembles were searched and evaluated. A threshold of 3 classifiers was applied as the minimum number of classifiers for an EoC during the whole searching process.

The selected ensembles were then combined by two types of fusion functions: The SUM rule (56; 109; 111) and the PCM-MAJ rule (58). Among all objective functions, the best ensemble was selected by the CDF-CFD and composed of 16 HMM classifiers. The recognition rate achieved by the selected ensemble is 98.80% with the SUM rule, and 98.84% with the PCM-MAJ rule. For all replications of GA searching, the variances are smaller than 0.01%, which indicates that the GA searching gives quite stable results.

We showed the results in Table XIV and Table XV. We note that the selected ensemble did perform better than column-HMM classifiers and row-HMM classifiers, but showed limited improvement compared with the ensemble of all the HMM classifiers. The PCM-MAJ rule performed better than the SUM rule on the selected ensemble. The PCM-MAJ has an improvement of 0.86% compared with the Benchmark EoHMM, and of 0.16% compared with XB-Selection EoHMM.

Table XV

Best Performances from 30 GA replications on the test data set. The numbers of classifiers are noted in parenthesis. The PCM-MAJ was used as the fusion function in EoC

Recognizers	Column HMM classifiers	Row HMM classifiers	Column & Row HMM classifiers
Classifier Pool	98.56 % (20 / PCM-MAJ)	98.20 % (20 / PCM-MAJ)	98.84 % (40 / PCM-MAJ)
EoHMM Selection	-	-	**98.86 %** (16 / PCM-MAJ)

Fig.- 18 and Fig.- 19 showed the rejection curves of the SUM rule and of the PCM-MAJ rule respectively. For the Sum rule, it is apparent that the selected ensemble performed

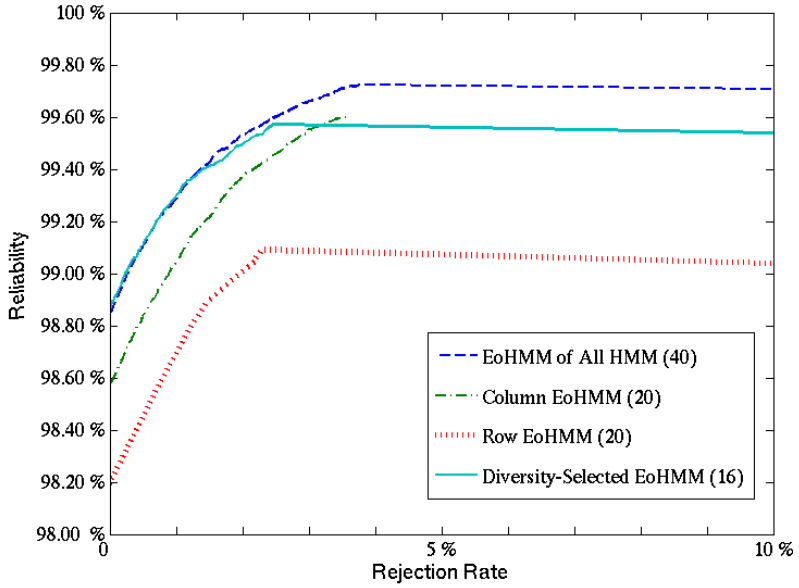

Figure 19 The Rejection mechanism with the PCM-MAJ rule

better than the column-HMM ensemble and the row-HMM ensemble, and had the comparable performance with the ensemble of all HMM classifiers (Fig.18).

If the PCM-MAJ rule was applied, we see that it offered a better improvement than the SUM rule for the rejection rate smaller than 2%. But unlike the SUM rule, it is hard for the PCM-MAJ rule to do more rejection when the majority of classifier-pairs agrees on the most of samples (59). After achieving a certain threshold, the system needs a much larger rejection rate to do further rejection. What is more, if all classifier-pairs agree on the most of samples, it is impossible to have more rejection, as in the case of the column-HMM ensemble (Fig.19). Note that to apply PCM-MAJ, the ensembles must have more than 2 classifiers, and thus we cannot use PCM-MAJ as a fusion function on the Benchmark EoHMM and on the XB-Selection EoHMM.

4.4 Discussion

In this work, we carried out a general analysis of the clustering validity indices in the literature. Of all of them, the XB index, Dunn's index and the DB index were regarded as the most reasonable. Dunn's index has a much higher calculation complexity, and thus is not applicable in large data sets. The DB index is less time-consuming and detects the best clustering for each cluster i by its statistic component $R_{i,qt}$, but the DB index has a drawback, which is its global evaluation with $\sum_{i=1}^{K} R_{i,qt}$. In contrast, the XB index targets the minimum distance d_{min} between the centroids of the two nearest clusters, and thus evaluates clustering by its worst local case. What is more, the XB index demonstrates the stronger inclination to show multiple levels of granularity of data set. The XB index is thus considered more adequate for the selection of codebooks.

HMM classifiers constructed with codebook sizes selected by the XB index show a clear improvement compared with benchmark HMM classifiers, in both column HMM classifiers and row HMM classifiers (8; 9). With an improvement of 0.80% over the benchmark column HMM classifier and 1.21% over the benchmark row HMM classifier, the usefulness of the XB index in optimizing HMM is undeniable.

As a by-product, we can also use these HMM classifiers trained with different codebook sizes to construct an EoHMM. With the SUM fusion function, the improvement in the classification accuracy of the ensemble of column HMM classifiers is 0.14% over that of the single best column HMM classifier, while the improvement in the accuracy of the ensemble of row HMM classifiers is 0.29% over that of the single best row HMM classifier. Considering that the best column HMM classifier already has a classification accuracy of 98.40% and the best row HMM classifier has a recognition rate of 97.97%, this improvement is significant. Such an improvement also indicates that the disadvantage of discrete HMM can be compensated for by EoHMM based on various codebook sizes.

Considering the objective function for EoHMM ensemble selection, the SUM rule and all the CDF rules give similar and comparable results. We also note that, by combining column HMM classifiers and row HMM classifiers, the single best EoHMM of all the replications can have a classification accuracy of 98.86%. This is about 0.30% better than COL-HMM, thanks to the further diversity contributed by row features and column features (Table XIV & Table XV).

We note that the proposed method has a speed-up advantage over other EoHMM creation schemes. Suppose we need to construct M HMM classifiers for EoHMM, given S possible codebook sizes, the proposed scheme evaluates S clusterings using the XB index and then trains M HMM classifiers. For other ensemble creation methods, such as Bagging, Boosting, and Random Subspaces, we need to train $M * S$ HMM classifiers and then select among them for the best codebook size. This offers a significant speed-up in the optimization of the codebook sizes and a new ensemble creation method.

Considering other classification methods applied in the same data set, KNN with 150000 samples can achieve 98.57% accuracy, MLP can achieve 99.16% accuracy (75), and the use of SVM can achieve a 99.30% recognition rate with a pairwise coupling strategy and a 99.37% with the one-against-all strategy (74). EoHMM performance very close to that, and its further optimization might achieve better results.

4.5 Conclusion

A fast codebook size optimization method for HMM and a new scheme of ensemble of discrete HMM were proposed in this chapter. The codebook size was selected by evaluating the quality of clustering during the construction of codewords. Because the method does not require any HMM classifiers training, the proposed scheme offers a significant speed-up for codebook size optimization. In order to fairly evaluate clustering quality, we used a clustering validity index based on different predefined numbers of clusters.

Though a number of clustering validity indices were available, we used the XB index because it has the strong theoretical support (108) and has been shown effective in clustering (4; 78). Moreover, the XB index demonstrated the property of discovering multiple levels of granularity in the data set, which would allow us to select adequate codebook sizes. In general, the HMM classifiers with codebook sizes selected by the XB index demonstrated an apparently better performance than benchmark HMM classifiers. As a by-product, we can construct an EoHMM trained with the full samples and full features based on different codebook sizes. Because the XB index gives multiple fit codebook sizes, these codebook sizes could result in more accurate and diverse HMM classifiers, and thus provide us with an EoHMM. The combination of column HMM classifiers and row HMM classifiers further improve the global performance of EoHMM.

To conclude, the result suggests that the new EoHMM scheme is applicable. The degradation associated with vector quantization in discrete HMM is compensated by the use of EoHMM without the need to deal with a number of optimization of parameters found in continuous HMM. EoHMM can also explore the advantage of the number of different ensemble combination methods proposed in the literature.

Future work is planned to further improve the performance of EoHMM by exploring the issue of the number of states that need to be optimized as well. With EoHMM based on different numbers of states, it will be possible to obtain further improvement without adding any parameters optimization problems, which will be of the great interest in the application of HMM. Furthermore, the codebook pruning will be also an interesting issue for the decrease of the computation cost for the construction of HMM classifiers.

At this chapter, we have already a complete system for ensemble creation, ensemble selection and classifier combination. We see the impacts of these processes and the improvements on an EoC. However, the system is not perfect. We can improve it on a number of issues. For one, we note that the ensemble selection process is largely a static one. That

is, we select one ensemble for all test patterns. The question is: Can we select different ensembles for different test patterns? We believe that this approach is feasible, and thus propose a new dynamic ensemble selection scheme at the next chapter.

CHAPTER 5

FROM DYNAMIC CLASSIFIER SELECTION TO DYNAMIC ENSEMBLE SELECTION

Static selection schemes select an EoC for all test patterns, and dynamic selection schemes select different classifiers for different test patterns. Nevertheless, it has been shown that traditional dynamic selection performs no better than static selection. We propose four new dynamic selection schemes which explore the properties of the oracle concept. Our results suggest that the proposed schemes, using the majority voting rule for combining classifiers, perform better than the static selection method.

5.1 Introduction

The mechanism for ensemble selection is designed to select adequate classifiers from a pool of different classifiers, so that the selected group of classifiers can achieve optimum recognition rates. We can perform this task either by static selection, i.e. selecting an EoC for all test patterns, or by dynamic selection, i.e. selecting different EoCs for different test patterns.

However, since different test patterns are, in general, associated with different classification difficulties, it is reasonable to assume that they might be better if they are fit to different classifiers rather than to a single static EoC. This may give us reason to believe that dynamic classifier selection is better than static ensemble selection. The dynamic scheme explores the use of different classifiers for different test patterns (12; 15; 14; 28; 44; 65; 107). Based on the different features or different decision regions of each test pattern, a classifier is selected and assigned to the sample. Some popular methods are a priori selection, a posteriori selection, overall local accuracy and local class accuracy (15; 14; 28; 107), hereafter referred to as the A Priori, A Posteriori, OLA and LCA methods respectively. In general, their performances are compared with that of the oracle, which assigns the

correct class label to a pattern if at least one individual classifier from an ensemble produces the correct class label for this pattern. Against all expectations, however, it has been shown that there is a large performance gap between dynamic classifier selection and the oracle (15), and, moreover, dynamic classifier selection does not necessarily give better performance than static ensemble selection (28).

A critical point in dynamic classifier selection is that our choice of one individual classifier over the rest will depend on how much we trust the estimate of the generalization of the classifiers (65). The advantage of dynamic ensemble selection is that we distribute the risk of this over-generalization by choosing a group of classifiers instead of one individual classifier for a test pattern. So far, this scheme seems to work well.

We note that most dynamic classifier selection schemes use the concept of classifier accuracy on a defined neighborhood or region, such as the local accuracy A Priori or A Posteriori methods (15). These classifier accuracies are usually calculated with the help of KNN, and its use is aimed at making an optimal Bayesian decision. However, KNN could be still outperformed by some static ensemble selection rule, such as the MVE. This poses a dilemma in the estimation of these local accuracies, because their distribution might be too complicated for a good result. Interestingly, dynamic classifier selection is regarded as an alternative to EoC (15; 14; 107), and is supposed to select the best single classifier instead of the best EoC for a given test pattern. The question of whether or not to combine dynamic schemes and EoC in the selection process is a debate being carried out (65). But, in fact, the two are not mutually exclusive. Hybrid methods have been shown to be useful, in that they apply the methods for different patterns (44; 65). However, we are interested in exploring another type of approach here, because we believe that ensemble selection can be dynamic as well. This means that, instead of performing dynamic classifier selection, we will perform dynamic ensemble selection (Fig. 20).

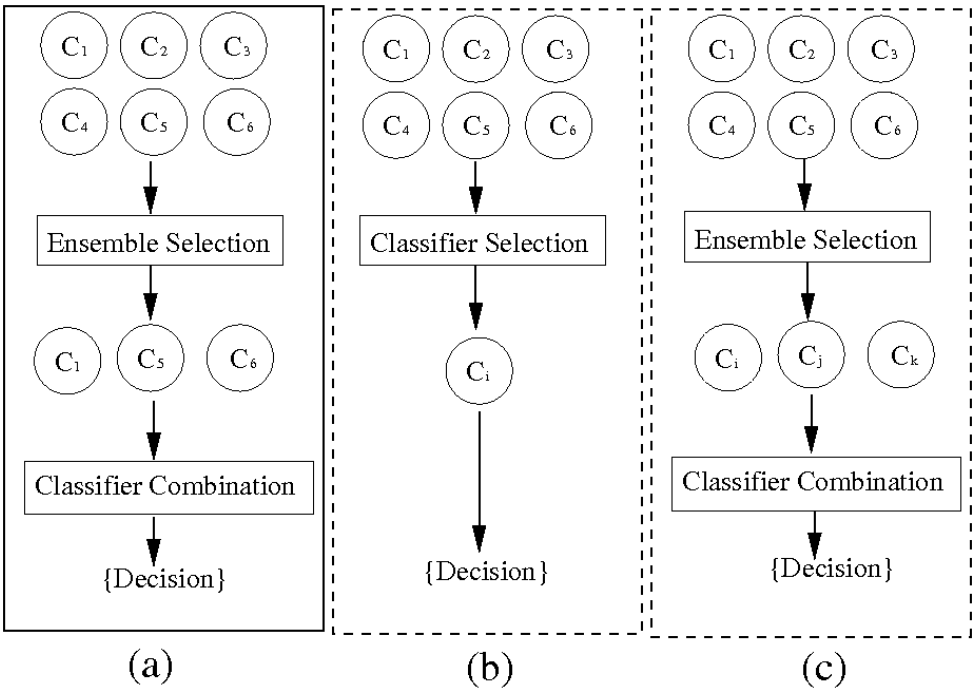

Figure 20 Three different schemes for selection and combining classifiers: (a) static ensemble selection; (b) dynamic classifier selection; (c) proposed dynamic ensemble selection. The solid line indicates a static process carried out only once for all patterns, and the dash lines indicate dynamic process repeated each time for a different test pattern

We also note that the oracle is usually regarded as a possible upper bound for EoC performances. As far as we know, no effort has been made to explore the appropriateness of the properties of the oracle for dynamic selection. We believe that the complicated process of local classifier accuracy estimation can actually be carried out by the oracle on a validation data set, and a simple KNN method can allow the test data set to obtain the approximate local classifier accuracy from the validation data set. Here are the key questions that need to be addressed:

a. Can the concept of the oracle be useful in dynamic ensemble selection?

b. Can dynamic ensemble selection outperform dynamic classifier selection?

c. Can dynamic ensemble selection outperform static ensemble selection?

To answer these questions, we propose a dynamic ensemble selection scheme which explores the properties of the oracle concept, and compare the scheme with static ensemble selection guided by different objective functions. All the approaches are evaluated on small-scale pattern recognition problems taken from the UCI machine learning repository, and on a large-scale pattern recognition problem related to the recognition of handwritten numerals from NIST SD19. It is important to state that the purpose of this work is not to achieve the best handwritten pattern recognition rate using dynamic selection, but to explore a potential advantage of dynamic selection which might suit the nature of the dynamic environment in machine learning, such as incremental learning. In order to gain a better understanding of the impact of dynamic selection, we use weak classifiers in our experiment.

5.2 Dynamic Classifier Selection Methods

5.2.1 Overall Local Accuracy (OLA)

The basic idea of this scheme is to estimate each individual classifier's accuracy in local regions of the feature space surrounding a test sample, and then use the decision of the most locally accurate classifier (107). Local accuracy is estimated as the percentage of training samples in the region that are correctly classified.

5.2.2 Local Class Accuracy (LCA)

This method is similar to the OLA method, the only difference being that the local accuracy is estimated as the percentage of training samples relative to output classes (107). In other words, we consider the percentage of the local training samples assigned to a class cl_i by this classifier that have been correctly labeled.

5.2.3 A Priori Selection Method (a priori)

The classifier accuracycan be weighted by the distances between the training samples in the local region and the test sample. Consider the sample $x_j \in \omega_k$ as one of the k-nearest neighbors of the test pattern X. The $\hat{p}(\omega_k|x_j, c_i)$ provided by classifier c_i can be regarded as a measure of the classifier accuracy for the test pattern X based on its neighbor x_j. If we suppose that we have N training samples in the neighborhood, then the best classifier C_* for classifying the sample X can be selected by (15; 28):

$$C_* = \arg_i \max \frac{\sum_{j=1}^{N} \hat{p}(\omega_k|x_j \in \omega_k, c_i) W_j}{\sum_{j=1}^{N} W_j} \tag{5.1}$$

where $W_j = \frac{1}{d_j}$ is the distance between the test pattern X and the its neighbor sample x_j.

5.2.4 A Posteriori Selection Method (a posteriori)

If the class assigned by the classifier c_i is known, then we can use the classifier accuracy in the aspect of the known class. Suppose that we have N training samples in the neighborhood and let us consider the sample $x_j \in \omega_k$ as one of the k-nearest neighbors of the test pattern X. Then, the best classifier $C_*(\omega_k)$ with the output class ω_k for classifying the sample X can be selected by (15; 28):

$$C_*(\omega_k) = \arg_i \max \frac{\sum_{x_j \in \omega_k} \hat{p}(\omega_k|x_j, c_i) W_j}{\sum_{j=1}^{N} \hat{p}(\omega_k|x_j, c_i) W_j} \tag{5.2}$$

where $W_j = \frac{1}{d_j}$ is the distance between the test sample and the training sample.

5.3 K-Nearest-Oracles (KNORA) Dynamic Ensemble Selection

All the above dynamic selection methods are designed to find the classifier with the greatest possibility of being correct for a sample in a pre-defined neighborhood. We, however,

are proposing another approach: Instead of finding the most suitable classifier, we select the most suitable ensemble for each sample.

The concept of the K-Nearest-Oracles (KNORA) is similar to the concepts of OLA, LCA, and the A Priori and A Posteriori methods in their consideration of the neighborhood of test patterns, but it can be distinguished from the others by the direct use of its property of having training samples in the region with which to find the best ensemble for a given sample. For any test data point, KNORA simply finds its nearest K neighbors in the validation set, figures out which classifiers correctly classify those neighbors in the validation set and uses them as the ensemble for classifying the given pattern in that test set.

We propose four different schemes using KNORA:

a. KNORA-ELIMINATE

Given K neighbors $x_j, 1 \leq j \leq K$ of a test pattern X, and supposing that a set of classifiers $C(j), 1 \leq j \leq K$ correctly classifies all its K nearest neighbors, then every classifier $c_i \in C(j)$ belonging to this correct classifier set $C(j)$ should submit a vote on the sample X. In the case where no classifier can correctly classify all the K nearest neighbors of the test pattern, then we simply decrease the value of K until at least one classifier correctly classifies its neighbors (Fig. 21).

b. KNORA-UNION

Given K neighbors $x_j, 1 \leq j \leq K$ of a test pattern X, and supposing that the j nearest neighbor has been correctly classified by a set of classifiers $C(j), 1 \leq j \leq K$, then every classifier $c_i \in C(j)$ belonging to this correct classifier set $C(j)$ should submit a vote on the sample X. Note that, since all the K nearest neighbors are considered, a classifier can have more than one vote if it correctly classifies more than one neighbor. The more neighbors a classifier classifies correctly, the more votes this classifier will have for a test pattern (Fig. 22).

c. KNORA-ELIMINATE-W

This scheme is the same as KNORA-ELIMINATE, but each vote is weighted by the distance between the neighbor pattern x_j and the test pattern X.

d. KNORA-UNION-W

This scheme is the same as KNORA-UNION, but each vote is weighted by the distance between the neighbor pattern x_j and the test pattern X.

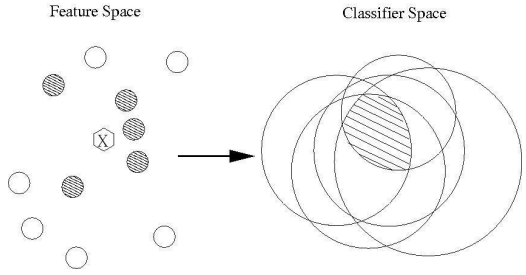

Figure 21 The KNORA-ELIMINATE only uses classifiers that correctly classify all the K-nearest patterns. On the left side, test pattern is shown as a hexagon, validation data points are shown as circles and the 5 nearest validation points are darkened. On the right side, the used classifiers -the intersection of correct classifiers- are darkened

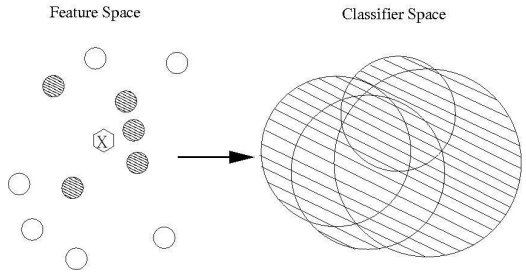

Figure 22 The KNORA-UNION uses classifiers that correctly classify any of the K-nearest patterns. On the left side, test pattern is shown as a hexagon, validation data points are shown as circles, and the 5 nearest validation points are darkened. On the right side, the used classifiers -the union of correct classifiers- are darkened

5.3.1 Comparison of Dynamic Selection Schemes on UCI Repository

To ensure that KNORA is useful for dynamic ensemble selection, 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 6 databases selected from a UCI data repository (see Table XVI). In general, among the available samples, 50% are used as a training data set and 50% are used as a test data set, except for the Image Segmentation data set, the training data set and test data set of which have been defined on the UCI data repository. Of the training data set, 70% of the samples are used for classifier training and 30% are used for validation.

Three ensemble creation methods have been used in our study: Random Subspaces, Bagging and Boosting. 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. The cardinality of Random Subspaces 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 Neighbor Classifiers (KNN), Parzen Windows Classifiers (PWC) and Quadratic Discriminant Classifiers (QDC) (19). For each of 6 databases and for each of 3 classification algorithms, 10 classifiers were generated to constitute the pool of classifiers. We used different dynamic selection schemes to select ensembles from the pools of 10 classifiers, and then combined these ensembles with the simple Majority Voting Rule (MAJ).

Table XVI

UCI data for ensembles of classifiers. 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

Database	Classes	Tr	Ts	Features	RS-Card.	Bagging	Boosting
Liver-Disorders (LD)	2	172	172	6	4	66 %	66 %
Pima-Diabetes (PD)	2	384	384	8	4	66 %	66 %
Wisconsin Breast-Cancer (WC)	2	284	284	30	5	66 %	66 %
Wine (W)	3	88	88	13	6	66 %	66 %
Image Segmentation (IS)	7	210	2100	19	4	66 %	66 %
Letter Recognition (LR)	26	10000	10000	16	12	66 %	66 %

5.3.2 Random Subspace

The Random Subspace method creates diverse classifiers by using different subsets of features to train classifiers. Due to the fact that problems are represented in different subspaces, different classifiers develop different borders for the classification.

For Random Subspace, we observe that KNORA-UNION and LCA have more stable performances than other methods. We also observe that the A Priori and A Posteriori methods are not necessarily better than OLA or LCA. This means that the probabilities weighted by the Euclidean distances between the test pattern and validation patterns are not always useful for dynamic classifier selection.

Similarly, we note that KNORA-UNION-W is not always better than KNORA-UNION. More interestingly, KNORA-ELIMINATE-W and KNORA-ELIMINATE have the same performances on Random Subspaces. This indicates that the probabilities weighted by the Euclidean distances between the test pattern and validation patterns do not affect the decisions of KNORA-ELIMINATE on Random Subspaces.

Table XVII

Dynamic Selection results for Random Subspace using KNN classifiers. KN-E = KNORA-ELIMINATE; KN-E-W = KNORA-ELIMINATE-W;. KN-U= KNORA-UNION; KN-U-W= KNORA-UNION-W; a Pr = a Priori; a Post = a Posteriori; SB = Single Best

	KN-E	KN-E-W	KN-U	KN-U-W	a Pr.	a Post.	OLA	LCA	Oracle	All	SB
LD	78.47 %	78.47 %	80.56 %	**84.03** %	77.78 %	70.14 %	79.17 %	70.83 %	100.00 %	76.39 %	74.31 %
PD	**97.54** %	**97.54** %	96.83 %	96.48 %	94.37 %	93.66 %	96.83 %	93.66 %	98.25 %	96.13 %	96.83 %
WC	93.66 %	93.66 %	**94.37** %	93.66 %	90.85 %	80.99 %	93.31 %	88.38 %	99.65 %	92.61 %	95.07 %
W	**97.73** %	**97.73** %	**97.73** %	**97.73** %	**97.73** %	37.50 %	**97.73** %	**97.73** %	97.73 %	76.14 %	90.91 %
IS	78.29%	78.29%	**78.67**%	78.62%	75.81%	60.90%	75.43%	59.62 %	97.29 %	78.19 %	84.14 %
LR	83.33%	83.33%	83.85%	84.20%	84.84%	87.02%	84.84%	**87.24**%	94.78 %	83.08 %	85.32 %

Table XVIII

Dynamic Selection results for Random Subspace using Parzen classifiers. KN-E = KNORA-ELIMINATE; KN-E-W = KNORA-ELIMINATE-W;. KN-U= KNORA-UNION; KN-U-W= KNORA-UNION-W; a Pr = a Priori; a Post = a Posteriori; SB = Single Best

	KN-E	KN-E-W	KN-U	KN-U-W	a Pr.	a Post.	OLA	LCA	Oracle	All	SB
LD	71.53 %	71.53 %	72.22 %	**75.00** %	**75.00** %	65.28 %	71.53 %	67.36 %	89.58 %	70.83 %	75.00%
PD	**82.82** %	**82.82** %	82.03 %	82.29 %	81.51 %	65.63 %	80.99 %	77.08 %	92.19 %	78.12 %	79.69 %
WC	92.96 %	92.96 %	92.96 %	92.96 %	91.20 %	83.10 %	**93.31** %	87.68 %	98.94 %	91.55 %	92.96 %
W	88.64%	88.64%	81.82%	89.77%	87.50%	84.09%	89.77%	**90.91** %	100.00 %	76.14 %	88.71%
IS	79.90%	79.90%	80.05%	**80.19**%	78.10%	64.90%	77.76%	64.76 %	98.48 %	79.62 %	85.38%
LR	89.07%	89.07%	89.68%	89.81 %	**90.51**%	88.43%	**90.51**%	88.49 %	96.70 %	89.52 %	90.61%

5.3.3 Bagging

Bagging generates diverse classifiers by randomly selecting subsets of samples to train classifiers. Intuitively, we can see that classifiers will have different behaviors based on different sample subsets.

For Bagging, we note that KNORA-ELIMINATE, KNORA-UNION and LCA have good performances. As with Random Subspaces, A Priori and A Posteriori are not necessar-

Table XIX

Dynamic Selection results for Random Subspace using QDC classifiers. KN-E = KNORA-ELIMINATE; KN-E-W = KNORA-ELIMINATE-W;. KN-U= KNORA-UNION; KN-U-W= KNORA-UNION-W; a Pr = a Priori; a Post = a Posteriori; SB = Single Best

	KN-E	KN-E-W	KN-U	KN-U-W	a Pr.	a Post.	OLA	LCA	Oracle	All	SB
LD	63.89 %	63.89 %	61.11 %	**70.19** %	61.81 %	70.14 %	65.28 %	68.06 %	88.19 %	57.64 %	64.58%
PD	**80.21** %	**80.21** %	**80.21** %	**80.21** %	79.69 %	63.28 %	**80.21** %	75.26 %	93.23 %	77.86 %	79.43 %
WC	95.42 %	95.42 %	**95.07** %	**95.07** %	92.25 %	88.03 %	95.42 %	90.85 %	99.65 %	93.66 %	96.48 %
W	**98.86**%	**98.86**%	97.73%	**98.86**%	97.73%	96.59%	97.73%	95.45 %	100.00 %	96.59 %	96.77 %
IS	83.29%	83.29%	81.76%	82.19%	83.14%	39.52%	**84.19**%	37.76 %	95.29 %	78.24 %	83.24 %
LR	83.97%	83.97%	84.62%	85.00%	81.96%	85.99%	81.96%	**86.73** %	93.40 %	84.36 %	82.44 %

Table XX

Dynamic Selection results for Bagging using KNN classifiers. KN-E = KNORA-ELIMINATE; KN-E-W = KNORA-ELIMINATE-W;. KN-U= KNORA-UNION; KN-U-W= KNORA-UNION-W; a Pr = a Priori; a Post = a Posteriori; SB = Single Best

	KN-E	KN-E-W	KN-U	KN-U-W	a Pr.	a Post.	OLA	LCA	Oracle	All	SB
LD	59.03%	59.03%	**60.42**%	**60.42**%	58.33%	**60.42**%	59.03%	59.72 %	79.17 %	60.42 %	63.19 %
PD	74.22%	74.22%	**74.74**%	**74.74**%	73.70%	72.92%	74.22%	72.92 %	90.10 %	75.00 %	75.26 %
WC	**94.72**%	**94.72**%	93.66%	94.01%	93.31%	92.96%	**94.72**%	93.31 %	96.83 %	93.66 %	94.72 %
W	73.86%	73.86%	73.86%	73.86%	**75.00**%	73.86%	73.86%	73.86 %	81.82 %	72.73 %	73.86 %
IS	**87.67**%	**87.67**%	**87.67**%	**87.67**%	86.67%	85.24%	86.52%	**87.67** %	93.19 %	86.24 %	84.57 %
LR	93.89%	93.89%	93.94%	93.94%	93.07%	93.97%	93.07%	**94.05** %	97.64 %	93.76 %	92.33 %

ily better than OLA or LCA on Bagging. Again, KNORA-UNION-W is not always better than KNORA-UNION. This indicates that the probabilities weighted by the Euclidean distances between the test pattern and validation patterns do not always contribute to higher classification rates for either dynamic classifier selection or dynamic ensemble selection.

Still, KNORA-ELIMINATE-W and KNORA-ELIMINATE have the same performances on Bagging.

Table XXI

Dynamic Selection results for Bagging using Parzen classifiers. KN-E = KNORA-ELIMINATE; KN-E-W = KNORA-ELIMINATE-W;. KN-U= KNORA-UNION; KN-U-W= KNORA-UNION-W; a Pr = a Priori; a Post = a Posteriori; SB = Single Best

	KN-E	KN-E-W	KN-U	KN-U-W	a Pr.	a Post.	OLA	LCA	Oracle	All	SB
LD	67.36%	67.36%	66.67%	**68.75**%	68.06%	61.81%	67.36%	62.50 %	94.44 %	65.28 %	68.06%
PD	**74.74**%	**74.74**%	72.40%	71.88%	73.70%	74.22%	74.22%	74.48 %	84.64 %	71.88 %	72.40%
WC	94.72%	94.72%	93.31%	93.31%	93.31%	92.61%	**95.07**%	92.61 %	97.18 %	91.90 %	94.01%
W	73.86%	73.86%	73.86%	73.86%	**76.14**%	73.86%	**76.14**%	73.86 %	85.23 %	71.59 %	73.86%
IS	**84.62**%	**84.62**%	82.90%	82.95%	84.43%	82.14%	83.76%	84.43 %	89.90 %	80.00 %	81.76%
LR	94.51%	94.51%	94.56%	**94.58**%	93.72%	94.17%	93.72%	94.22 %	97.63 %	94.33 %	92.99%

Table XXII

Dynamic Selection results for Bagging using QDC classifiers. KN-E = KNORA-ELIMINATE; KN-E-W = KNORA-ELIMINATE-W;. KN-U= KNORA-UNION; KN-U-W= KNORA-UNION-W; a Pr = a Priori; a Post = a Posteriori; SB = Single Best

	KN-E	KN-E-W	KN-U	KN-U-W	a Pr.	a Post.	OLA	LCA	Oracle	All	SB
LD	**70.83**%	**70.83**%	63.89%	66.67%	68.75%	61.11%	70.14%	62.50 %	91.67 %	56.94 %	68.75
PD	74.22%	74.22%	**74.48**%	73.96%	73.70%	72.66%	**74.48**%	72.92 %	83.85 %	73.96 %	74.22 %
WC	97.89%	97.89%	96.83%	96.83%	97.54%	98.94%	97.54%	**99.30** %	100.00 %	96.83 %	98.24%
W	**100.00** %	**100.00**%	98.86%	98.86%	94.32%	94.32%	94.32%	95.45 %	100.00 %	97.73 %	96.59 %
IS	**100.00** %	**100.00**%	99.14%	97.33 %	**100.00**%	91.29 %	**100.00** %	**100.00** %	100.00 %	100.00 %	100.00%
LR	89.70%	89.70%	89.01%	88.99%	89.64%	91.04%	89.61%	**91.29** %	92.81 %	88.47 %	88.21%

5.3.4 Boosting

Boosting uses a part of the samples to train classifiers, but not randomly. As stated above, difficult samples have higher probability of being selected, and easier samples have less chance of being used for training. With this mechanism, most of the classifiers created will focus on hard samples and can be more effective.

For Boosting, KNORA-ELIMINATE, KNORA-UNION and LCA seem to be quite stable. We observe the same situations as for Random Subspaces and Bagging: the A Priori and A Posteriori methods are not necessarily better than OLA or LCA; KNORA-UNION-W is not always better than KNORA-UNION, and KNORA-ELIMINATE-W and KNORA-ELIMINATE have the same performances.

However, these results cannot discount the usefulness of the probabilities weighted by the Euclidean distances between the test pattern and validation patterns, because, in many problems, the number of samples is quite small. Moreover, since there are only 10 classifiers in a pool, there are not many choices for either dynamic classifier selection or dynamic ensemble selection. This might also be a reason why KNORA-ELIMINATE and KNORA-ELIMINATE-W have the same performances.

Although the experiments suggest that the four KNORA schemes proposed for dynamic ensemble selection might be applicable in various ensemble creation methods – such as Random Subspace, Bagging and Boosting – the problems extracted from the UCI machine learning repository usually consist of a small number of samples with few features. Furthermore, given these constraints, the classifier pool is composed of only 10 classifier in our experiment, which makes the results less convincing. As a result, we were able to justify the need to carry out a larger scale experiment on a problem with more features and larger classifier pools. This is why we conducted our next experiment on a 10-class handwritten-numeral problem with 132 features and 100 classifiers.

5.4 Experiments for Dynamic Selection on Handwritten Numerals

5.4.1 Experimental Protocol for KNN

Our experiments were carried out on a 10-class handwritten-numeral problem. The data were extracted from NIST SD19, essentially as in (99), based on the ensembles of KNNs generated by the Random Subspaces method. We used nearest-neighbor classifiers ($K =$

Table XXIII

Dynamic Selection results for Boosting using KNN classifiers. KN-E = KNORA-ELIMINATE; KN-E-W = KNORA-ELIMINATE-W;. KN-U= KNORA-UNION; KN-U-W= KNORA-UNION-W; a Pr = a Priori; a Post = a Posteriori; SB = Single Best

	KN-E	KN-E-W	KN-U	KN-U-W	a Pr.	a Post.	OLA	LCA	Oracle	All	SB
LD	**66.67**%	**66.67**%	64.58%	65.28%	65.97%	64.58%	65.28%	65.28 %	90.28 %	62.50 %	62.50 %
PD	72.14%	72.14%	71.88%	71.09%	73.44%	73.44%	**75.00**%	73.44 %	91.67 %	71.09 %	72.14%
WC	95.77%	95.77%	95.42%	**96.13**%	95.42%	94.72%	94.37%	95.42 %	96.83 %	95.42 %	95.42%
W	73.86%	73.86%	73.86%	73.86%	73.86%	73.86%	73.86%	**76.14** %	78.41 %	71.59 %	73.86 %
IS	86.57%	86.57%	86.57%	86.57%	86.86%	86.71%	86.86%	**87.67** %	90.00 %	86.43 %	87.67%
LR	93.57%	93.57%	93.79%	93.80%	92.76%	93.95%	92.75%	**94.00** %	97.20 %	93.62 %	92.57%

Table XXIV

Dynamic Selection results for Boosting using Parzen classifiers. KN-E = KNORA-ELIMINATE; KN-E-W = KNORA-ELIMINATE-W;. KN-U= KNORA-UNION; KN-U-W= KNORA-UNION-W; a Pr = a Priori; a Post = a Posteriori; SB = Single Best

	KN-E	KN-E-W	KN-U	KN-U-W	a Pr.	a Post.	OLA	LCA	Oracle	All	SB
LD	66.67%	66.67%	67.36%	**72.92**%	63.89%	63.89%	66.67%	68.06 %	100.00 %	65.97 %	63.89 %
PD	74.74%	74.74%	73.96%	73.18%	73.70%	71.61%	**75.00**%	73.18 %	99.74 %	72.40 %	73.18 %
WC	**93.31**%	**93.31**%	92.96%	92.96%	92.96%	92.96%	**93.31**%	92.96 %	94.72 %	92.96 %	92.96 %
W	80.68%	80.68%	77.27%	**81.82**%	78.41%	73.86%	79.55%	73.86 %	95.45 %	75.00 %	79.55 %
IS	84.19%	84.19%	83.33%	83.38%	**84.90**%	83.76%	**84.90**%	84.71 %	88.43 %	80.48 %	82.81 %
LR	94.03%	94.03%	94.07%	94.10%	93.02%	94.17%	92.95%	**94.19** %	97.29 %	94.13 %	93.18 %

1) for KNN, each KNN classifier having a different feature subset of 32 features extracted from the total of 132 features.

To evaluate the static ensemble selection and the dynamic ensemble selection schemes, four databases were used: the training set with 5,000 samples ($hsf\_\{0-3\}$) to create 100 KNN in Random Subspaces. The optimization set containing 10,000 samples ($hsf\_\{0-3\}$) was used for GA searching for static ensemble selection. To avoid overfitting during

Table XXV

Dynamic Selection results for Boosting using QDC classifiers. KN-E = KNORA-ELIMINATE; KN-E-W = KNORA-ELIMINATE-W;. KN-U= KNORA-UNION; KN-U-W= KNORA-UNION-W; a Pr = a Priori; a Post = a Posteriori; SB = Single Best

	KN-E	KN-E-W	KN-U	KN-U-W	a Pr.	a Post.	OLA	LCA	Oracle	All	SB
LD	73.61%	73.61%	**77.08**%	**77.08**%	70.14%	61.81%	73.61%	64.58 %	96.53 %	70.83 %	75.00%
PD	**75.26**%	**75.26**%	73.96%	74.48%	73.70%	73.18%	74.22%	73.96%	86.98 %	74.22 %	74.74%
WC	97.18%	97.18%	96.83%	97.18%	95.77%	**97.89**%	95.77%	**97.89** %	98.59 %	96.83 %	97.89 %
W	96.59%	96.59%	96.59%	96.59%	96.59%	**97.73**%	96.59%	96.59 %	97.73 %	96.59 %	97.73%
IS	86.38%	86.38%	86.52%	86.48%	86.24%	86.43%	86.05%	**86.57** %	90.00 %	86.43 %	87.67%
LR	93.54%	93.54%	93.69%	93.73%	92.63%	93.95%	92.61%	**94.00** %	97.20 %	93.62 %	92.57%

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. Using the best solution from this archive (86), 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 the available features and all the training samples, a much better performance can be observed (15; 14). But, since this is not the objective of this chapter, we are focusing on the improvement of EoCs through the optimization of performances by 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. The possible upper limit of classification accuracy (the oracle) is defined as the ratio of samples classified correctly by at least one classifier in a pool to all samples. The oracle is 99.95% accurate for KNN.

5.4.2 Static Ensemble Selection with Classifier Performance

The MVE was tested because of its reputation as one of the best objective functions in selecting classifiers for ensembles (89). It directly evaluates the global EoC performance by the MAJ. For this reason, we tested the MAJ as the objective function for static and dynamic ensemble selection, as well as using it as the fusion function. We also tested the mean classifier error (ME).

In Table XXVI, we observe that the MVE performs better than the ME as an objective function for static ensemble selection. The ensemble selected by the MVE also outperforms that of all 100 KNNs.

Table XXVI

The recognition rates on test data of ensembles searched by GA with the Mean Classifier Error, Majority Voting Error. ME = Mean Classifier Error; MVE = Majority Voting Error; OF = Objective Functions

OF	Min	Q_L	Median	Q_U	Max
ME	94.18 %	94.18 %	94.18 %	94.18 %	94.18 %
MVE	96.32 %	96.41 %	96.45 %	96.49 %	96.57 %

5.4.3 Dynamic Ensemble Selection

Even though the MVE has thus far been able to find the best ensemble for all the samples, this does not mean that a single ensemble is the best solution for combining classifiers. In other words, each sample may have a most suitable ensemble that is different from that of the others. We intend to determine whether or not the use of different ensembles on different samples can further increase the accuracy of the system.

Table XXVII

The best recognition rates of proposed dynamic ensemble selection methods. RR= Recognition Rates

	KN-E	KN-E-W	KN-U	KN-U-W
RR	97.52 %	97.52 %	97.25 %	97.25 %
K-value	7,8	7,8	1	1

Note that dynamic ensemble selection does not use any search algorithm for selecting the ensemble, because each sample has its own ensemble for the classifier combination. As a result, it was not necessary to repeat the search.

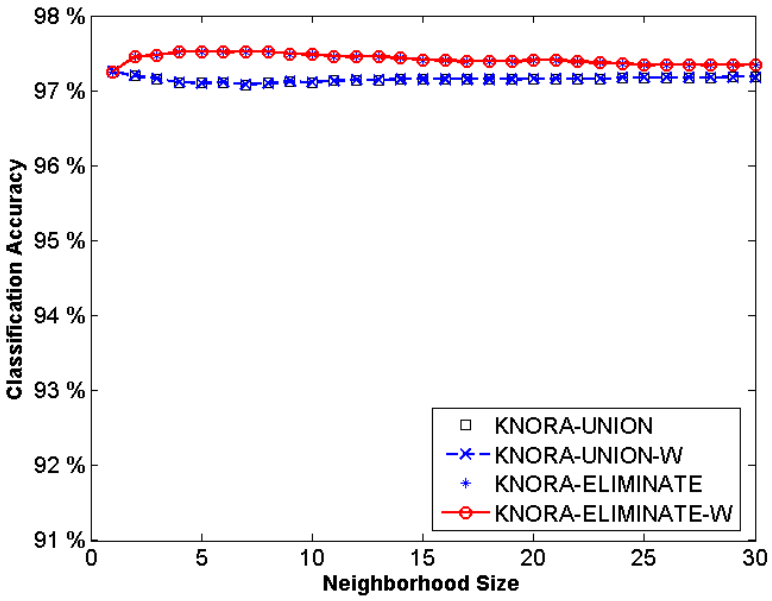

Figure 23 The performances of proposed dynamic ensemble selection schemes based on different neighborhood sizes $1 \leq k \leq 30$ on NIST SD19 database. In the figure KNORA-ELIMINATE overlaps with KNORA-ELIMINATE-W, and KNORA-UNION overlaps with KNORA-UNION-W

For dynamic ensemble selection, only three databases were used: the training set with 5,000 samples ($hsf\_\{0-3\}$) to create 100 KNN in Random Subspaces, the optimization set containing 10,000 samples ($hsf\_\{0-3\}$) and the test set containing 60,089 samples ($hsf\_\{7\}$) to evaluate the EoC accuracies. We tested the four KNORA algorithms and compared them with the other proposed schemes: OLA, LCA, and the A Priori and A Posteriori local class accuracy methods.

Table XXVIII

The best recognition rates of each dynamic ensemble selection methods. RR= Recognition Rates

	KNORA-E	OLA	LCA	a priori	a posteriori
RR	97.52 %	94.11 %	97.40 %	94.12 %	97.40 %
K-value	7,8	30	1	30	1

We note that most of the dynamic schemes have so far proved better than all the tested objective functions for static ensemble selection. The exceptions are OLA and the A Priori method. Both LCA and the A Posteriori method achieved very good performances, with 97.40% recognition rates. But KNORA-ELIMINATE and KNORA-ELIMINATE-W have good performances as well, and, with recognition rates of 97.52%, KNORA-ELIMINATE and KNORA-ELIMINATE-W turned out to constitute the best dynamic selection scheme for our handwritten-numeral problems (Table XXII).

However, KNORA-UNION and KNORA-UNION-W do not perform as well as KNORA-ELIMINATE. They are still better than OLA and the A Priori method, but not as good as LCA and the A Posteriori method (Fig. 23).

If we compare their performances in different neighborhood sizes, we note that, while the LCA and A Posteriori dynamic selection schemes outperform static GA selection with the MVE as the objective function in a small neighborhood, their performances declined

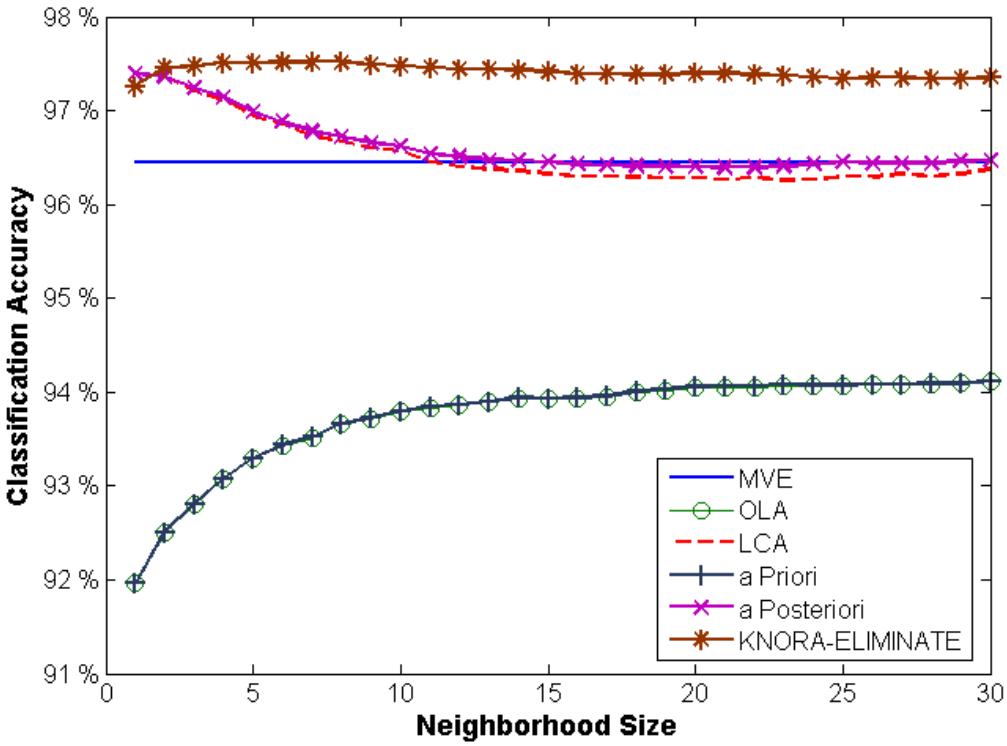

Figure 24 The performances of various ensemble selection schemes based on different neighborhood sizes $1 \leq k \leq 30$ on NIST SD19 database. In the figure OLA overlaps with a priori selection

with an increase in the value of k (Fig. 24). In this case, static GA selection with the MVE may still be better than the LCA or A Posteriori dynamic selection schemes. By contrast, KNORA-ELIMINATE has a more stable performance, even when the value of k increases. It gives a better recognition rates than all the other schemes in our experimental study, except when $k = 1$. But still, the stable performance of KNORA-ELIMINATE suggests that the dynamic selection schemes are worthy of more attention.

5.4.4 Effect of Validation Sample Size

Since all the traditional dynamic selection schemes and KNORA take into account the neighborhood of the test pattern for classifier and ensemble selection, the size of the validation samples will have somewhat of an effect on these methods.

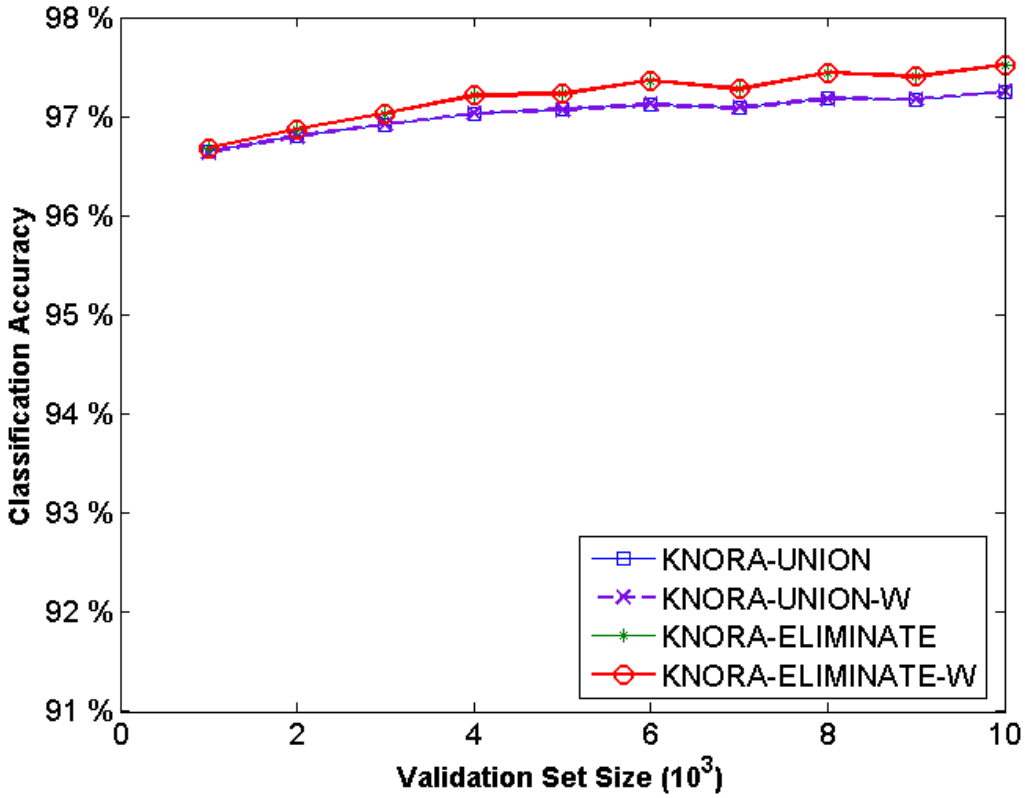

Figure 25 The performances of proposed dynamic ensemble selection schemes based on different validation sample sizes from 1000 to 10000 on NIST SD19 database. The best performances from neighborhood sizes $1 \leq k \leq 30$ are shown. The classifier pool size is 100. In the figure KNORA-ELIMINATE overlaps with KNORA-ELIMINATE-W, and KNORA-UNION overlaps with KNORA-UNION-W

We thus varied the size of the validation samples from 1000 to 10000 samples, and measured the impact of the variation on these dynamic selection schemes. As the number of

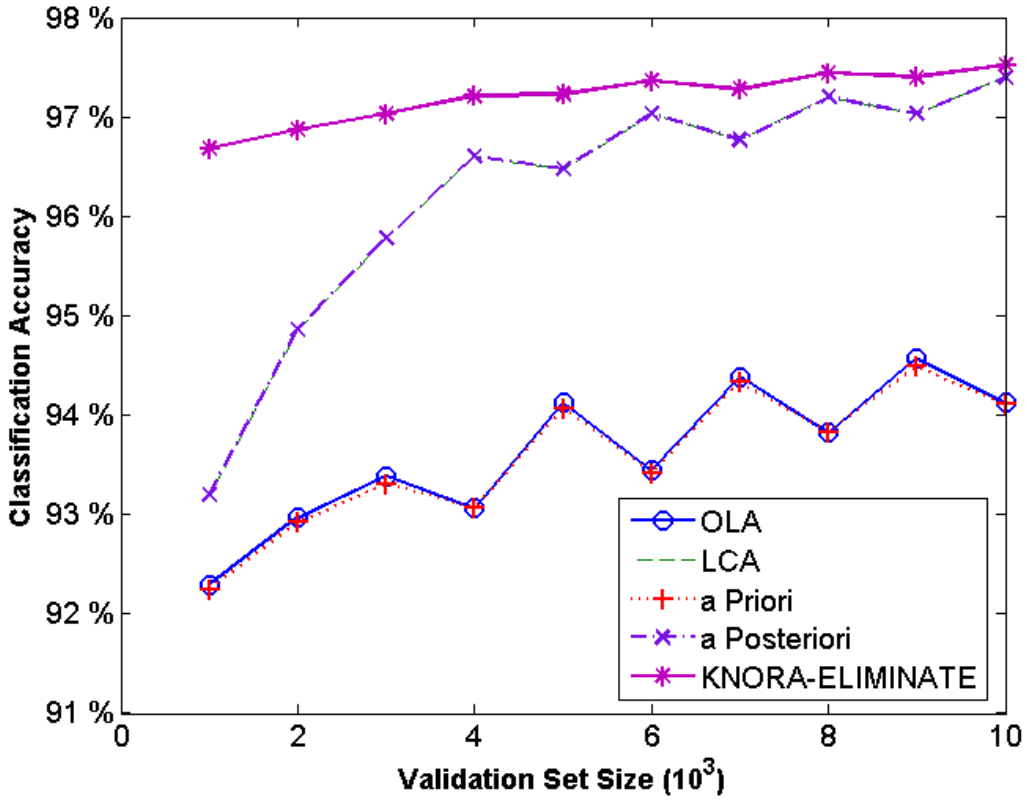

Figure 26 The performances of various ensemble selection schemes based on different validation sample sizes from 1000 to 10000 on NIST SD19 database. The best performances from neighborhood sizes $1 \leq k \leq 30$ are shown. The classifier pool size is 100. In the figure OLA overlaps with a priori selection, and LCA overlaps with a posteriori selection

validation samples increases, a test pattern is more likely to have better nearest neighbors. These nearest neighbors might also better distinguish truly useful classifiers from the pool.

Our results seem to confirm this supposition. When the validation sample size increases, all four proposed KNORA methods show slight improvement (Fig. 25). However, for the traditional dynamic selection schemes, the benefit to be derived from the increase in validation samples seems to be less stable. We observe some fluctuations in classification

accuracy on the four traditional dynamic selection schemes when the validation sample size increases (Fig. 26).

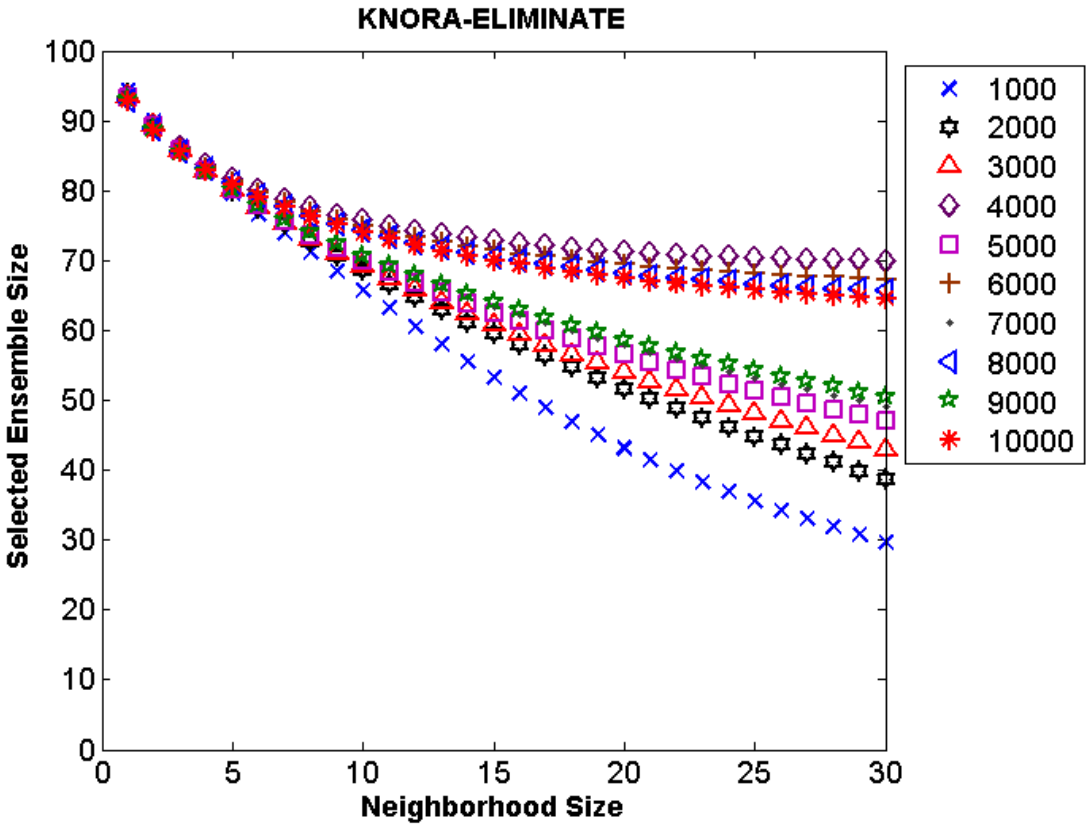

Figure 27 The relationship between selected ensemble size and neighborhood size on different validation sample sizes from 1000 to 10000 on NIST SD19 database for KNORA-ELIMINATE. The classifier pool size is 100

The interesting point is that all four KNORA methods demonstrate better performances than other traditional dynamic selection schemes when the validation sample size is small. Also note that the increase in sample size does not necessarily increase the selected ensemble sizes (Fig. 27 & Fig. 28).

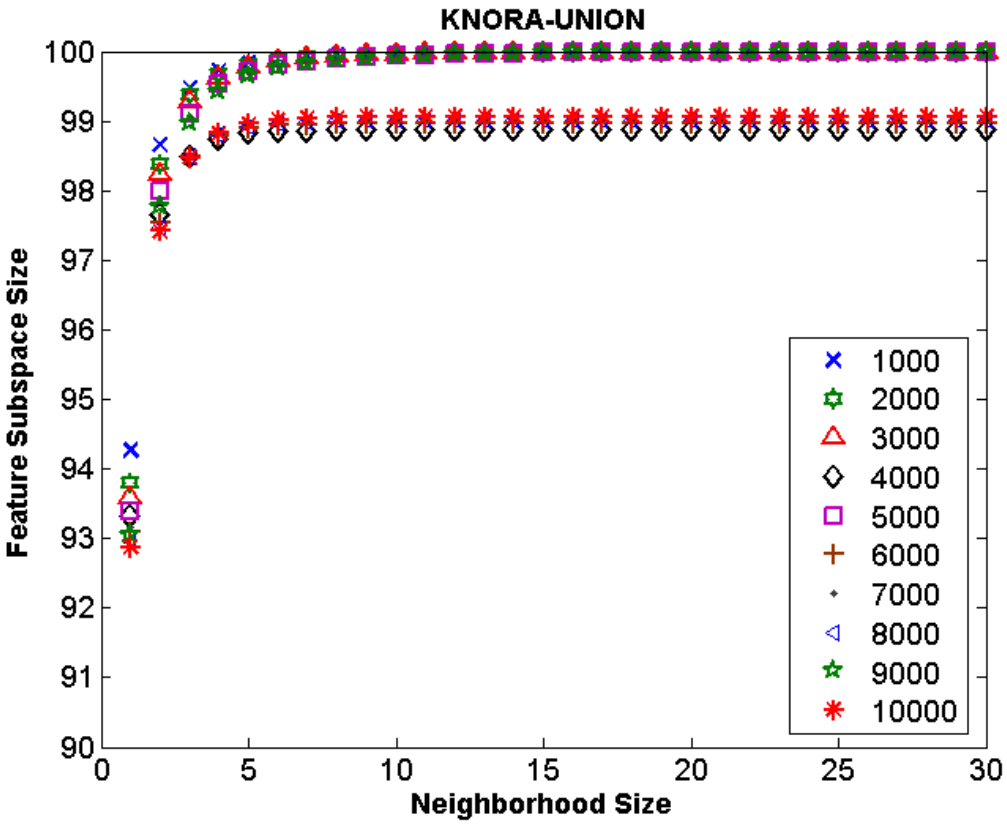

Figure 28 The relationship between selected ensemble size and neighborhood size on different validation sample sizes from 1000 to 10000 on NIST SD19 database for KNORA-UNION. The classifier pool size is 100

5.4.5 Effect of Classifier Pool Size

The classifier pool size has a clear effect on the performances of the proposed KNORA methods. While all four of these methods show improvement as the classifier pool size increases, KNORA-ELIMINATE and KNORA-ELIMINATE-W show a better improvement than KNORA-UNION and KNORA-UNION-W (Fig. 29). Compared with the traditional dynamic selection schemes, we note that KNORA-ELIMINATE is apparently superior to OLA and to the A Priori method, but it is not necessarily better than LCA or the A Posteriori method (Fig. 30).

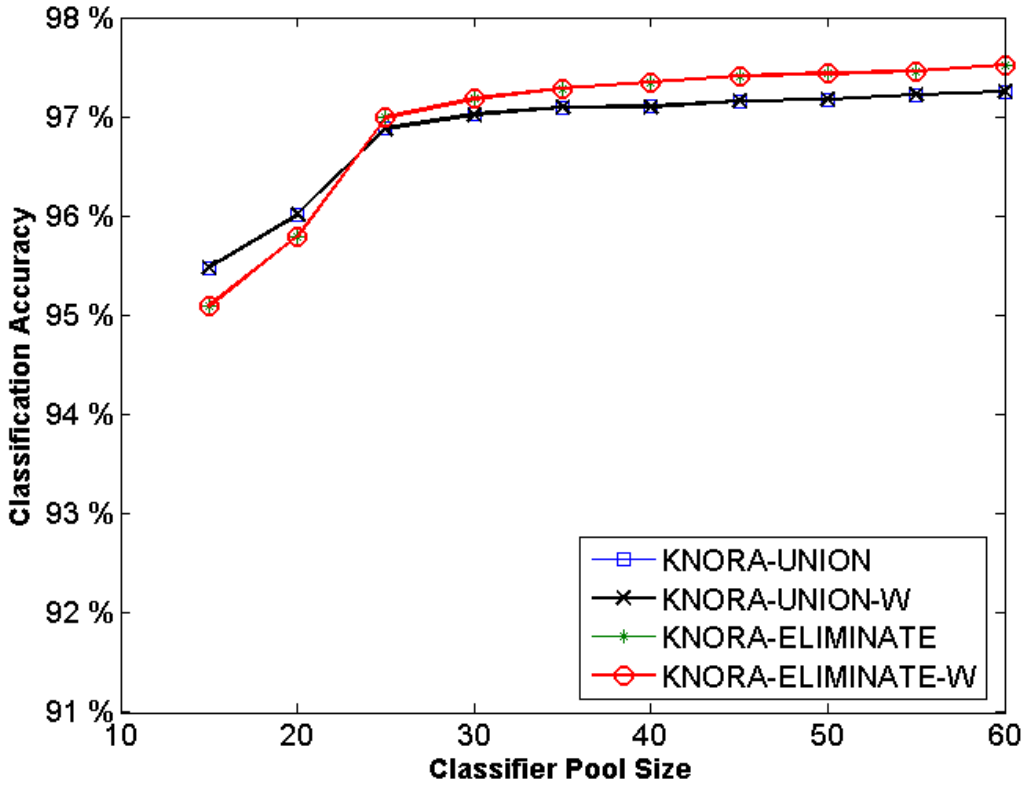

Figure 29 The performances of proposed dynamic ensemble selection schemes based on different classifier pool sizes from 10 to 100 on NIST SD19 database. The best performances from neighborhood sizes $1 \leq k \leq 30$ are shown. The validation sample size is 10000. In the figure KNORA-ELIMINATE overlaps with KNORA-ELIMINATE-W, and KNORA-UNION overlaps with KNORA-UNION-W

It is clear that the increase in classifier pool size benefits all kinds of dynamic selection methods, because more classifiers are available. Nevertheless, KNORA-ELIMINATE has shown more improvement than other dynamic selection schemes. We note that, when there are fewer than 70 classifiers in the pool, LCA and the A Posteriori method outperform KNORA-ELIMINATE. By contrast, when there are more than 70 classifiers in the pool, KNORA-ELIMINATE has a slightly better classification accuracy than LCA and the A Posteriori method.

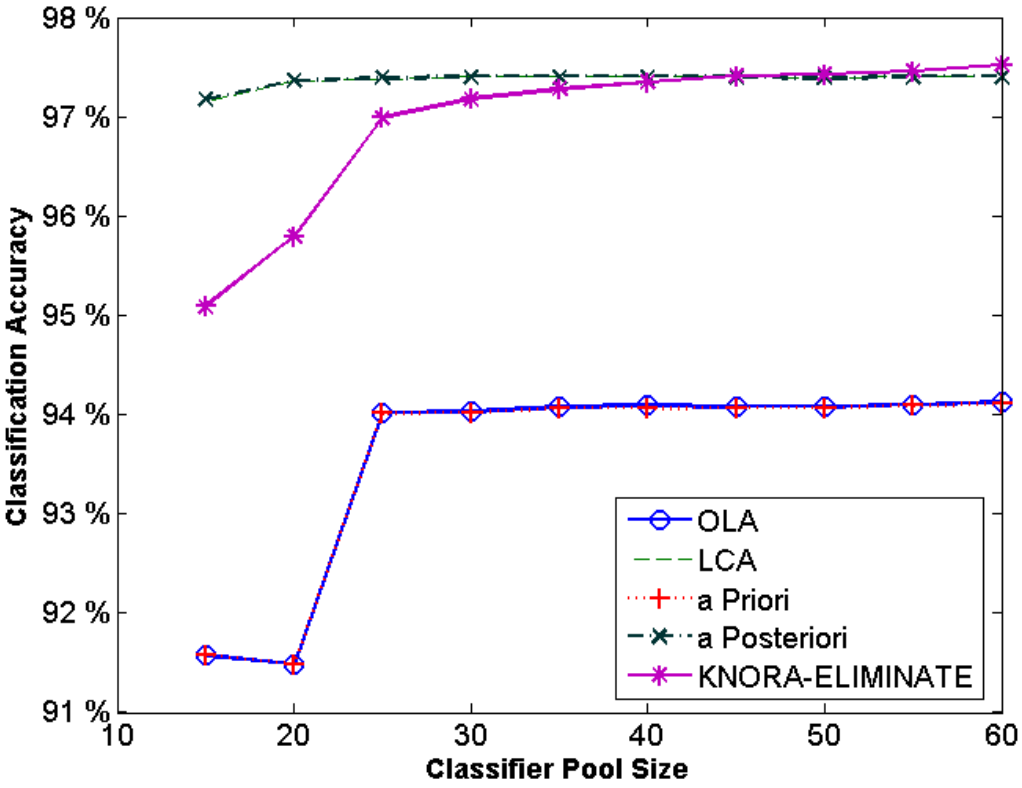

Figure 30 The performances of various ensemble selection schemes based on different classifier pool sizes from 10 to 100 on NIST SD19 database. The best performances from neighborhood sizes $1 \leq k \leq 30$ are shown. The validation sample size is 10000. In the figure OLA overlaps with a priori selection, and LCA overlaps with a posteriori selection

This is an interesting finding, since it indicates that the KNORA methods are better suited to large classifier pools. Since problems extracted from the UCI machine learning repository use only relatively small classifier pools, this might be why KNORA is not always better than the traditional dynamic selection schemes. Moreover, we also note that the increase in sample size does lead to the increase in the selected ensemble sizes (Fig. 31 & Fig. 32).

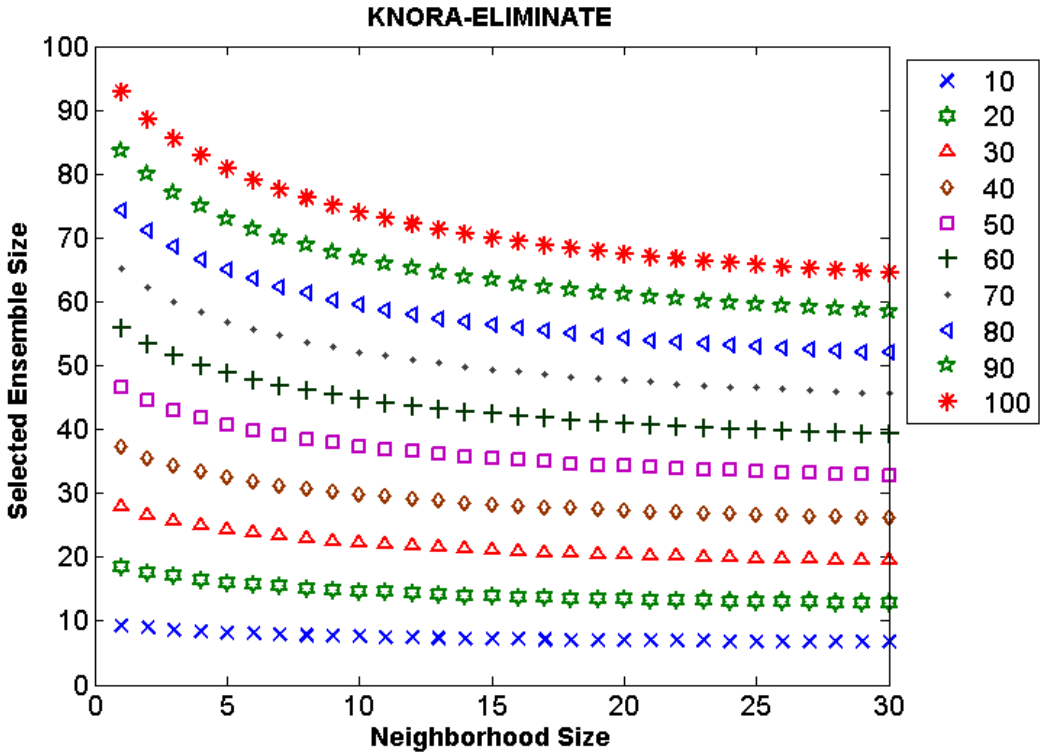

Figure 31 The relationship between selected ensemble size and neighborhood size on different classifier pool sizes from 10 to 100 on NIST SD19 database for KNORA-ELIMINATE. The validation sample size is 10000

5.5 Discussion

In this chapter, we propose a new dynamic ensemble selection scheme which directly applies the concept of the oracle on the validation set. Unlike other dynamic selection methods which use the estimated best classifier for a certain data point, the K-nearest oracle uses the EoCs that are estimated to be the best for dynamic ensemble selection.

In our study of handwritten numerals, the proposed method apparently outperforms the static ensemble selection schemes such as the use of the MVE or the ME as the objective function in a GA search. Using the GA search, the MVE can achieve 96.45% recog-

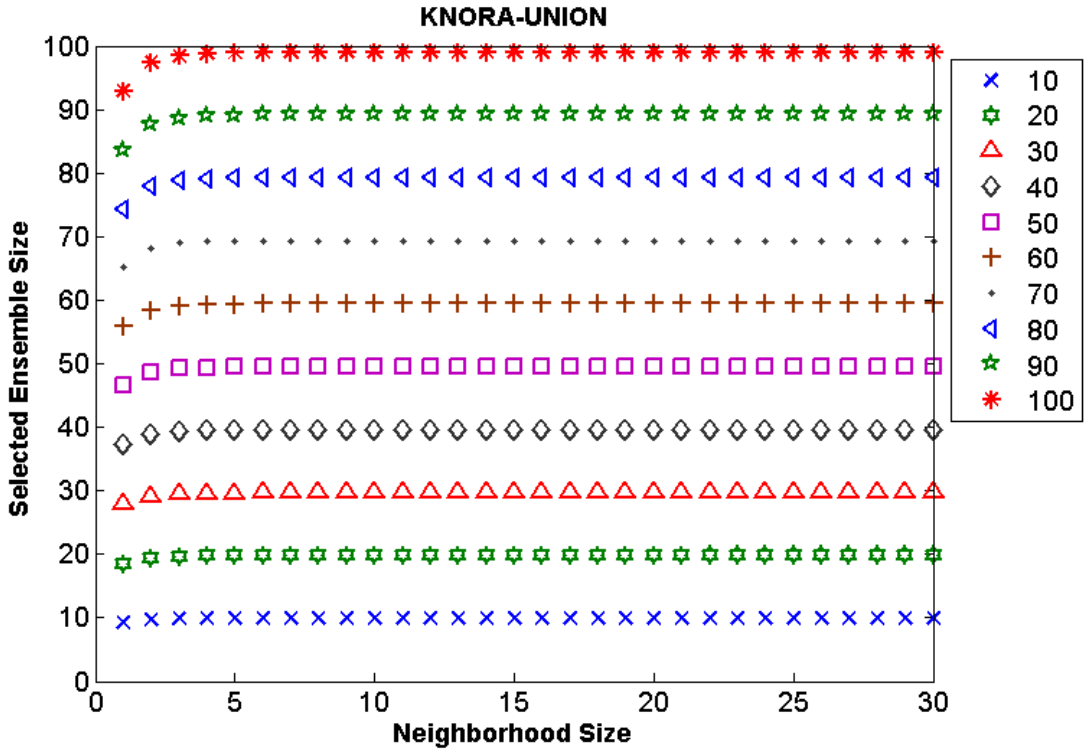

Figure 32 The relationship between selected ensemble size and neighborhood size on different classifier pool sizes from 10 to 100 on NIST SD19 database for KNORA-UNION The validation sample size is 10000

nition rates, and ME 94.18%. Nevertheless, with 97.52% recognition rates, KNORA-ELIMINATE is significantly better than the static ensemble selection methods evaluated.

We note that the OLA and A Priori dynamic selection schemes were not as good as the static GA selection scheme with the MVE. The OLA takes into the account neither class dependence nor the weighting of each classifier, while the A Priori method ignores class dependence. Since our experiment has a high class dimension (10) and the ensemble pool size is quite large (100), it is not surprising that they do not perform well.

We also observe that KNORA-UNION and KNORA-UNION-W perform less well than KNORA-ELIMINATE or KNORA-ELIMINATE-W. This might be due to the extreme

elitism in the behavior of the oracle. Since only very few classifiers can correctly classify some difficult patterns, the increase in ensemble size does not lead to a better recognition rate. So, when the value of K increases, the performances of KNORA-UNION and KNORA-UNION-W decline.

KNORA-ELIMINATE also performs slightly better than the other dynamic selection schemes. The LCA and A Posteriori schemes can achieve recognition rates of 97.40%, which is better than the other static methods, but not as good as KNORA-ELIMINATE. However, the performance of KNORA-ELIMINATE is still far from the oracle, which can achieve rates of 99.95%.

This might indicate that addressing the behavior of the oracle is much more complex than applying a simple neighborhood approach, and that the task of figuring out its behavior merely based on the pattern feature space is not an easy one.

Considering the effect of validation sample size, we note that all four KNORA methods demonstrate much better performances than other traditional dynamic selection schemes when the validation sample size is small. On the contrary, classifier pool size has an even more dramatic effect on KNORA performances. In general, when there are few classifiers in the pool, LCA and the A Posteriori method outperform the KNORA methods. However, when the classifier pool size increases, KNORA seems to improve more than LCA and the A Posteriori method. When a number of classifiers is given, KNORA seems to perform better than either LCA or the A Posteriori method (Fig. 30).

Note that, for an ensemble of M KNN classifiers with N training samples and with total features d and a cardinality of features c (size of fixed feature subspaces), we can first pre-calculate the distance on each feature. This pre-calculation has the complexity $O(d \cdot N)$. After the pre-calculation, we only need to carry out the summation and the sorting calculation, which have the complexity $O(M \cdot (c \cdot N + N \log N)$ of the ensemble, rather than the complexity $O(d \cdot N + N \log N)$ of a single KNN classifier. In our study, the

best dynamic selection scheme is KNORA-ELIMINATE with the neighborhood size 7, which used 76 classifiers on average, which means that its ensemble is 11.78 times more complex than a single KNN classifier, including the pre-calculation cost. However, the best performance of KNORA-ELIMINATE is 4.18% better than that of a single KNN classifier.

Finally, we must emphasize that the purpose of this work is not to achieve the best handwritten pattern recognition rate using dynamic selection, but to explore the potential advantages of dynamic selection that might suit the nature of the dynamic environment in machine learning, such as incremental learning. In order to gain a better understanding of the impact of dynamic selection, we use 100 KNN classifiers trained with only 5000 samples in our experimental study. The combination of these 100 KNN by simple MAJ gives only a 96.28% recognition rate. Considering other classification methods applied in the same data set, KNN trained with 150000 samples can achieve 98.57% accuracy, MLP can achieve 99.16% accuracy (75), and the use of SVM can achieve a 99.30% recognition rate with a pairwise coupling strategy and a 99.37% rate with the one-against-all strategy (74). However, the use of weak classifiers can demonstrate more differences between various ensemble selection schemes, which makes this a better option for comparing different ensemble selection schemes.

5.6 Conclusion

We describe a methodology to dynamically select an ensemble for every test data point. We find that by the direct use of the concept of the oracle, the proposed scheme apparently gives better performances than static ensemble selection schemes such as GA with the MVE as the objective function. Moreover, the proposed scheme also perform slightly better than other dynamic selection methods in our study.

We show that a dynamic ensemble selection scheme can, in some cases, perform better than some static ensemble selection methods. Furthermore, our study suggests that an en-

semble of classifiers might be more stable than a single classifier in the case of dynamic selection. Yet our method is limited by the uncertainty of the behavior of the oracle, since the recognition rates achieved are still not close to those of the oracle. We believe that this methodology can be greatly enhanced with theoretical studies on the connection between the feature subspaces and the classifier accuracies, the influence of geometrical and topological constraints on the oracle, better statistical studies to quantify the uncertainty of the oracle's behavior and empirical studies in more real-world problems with various ensemble generation methods.

Although we believe that this dynamic ensemble selection scheme is promising, like static ensemble selection, it has some drawbacks. One of these disadvantages is that we need to train some classifiers that might not be used. Since all classifiers are created based on data subsets, we wonder whether we can just only do a data subset selection instead of classifier selection. We thus propose a classifier-free ensemble selection at the next chapter.

CHAPTER 6

THE IMPLICATION OF DATA DIVERSITY FOR A CLASSIFIER-FREE ENSEMBLE SELECTION IN RANDOM SUBSPACES

To select the best EoC from a pool of classifiers, the classifier diversity is considered one of the most important properties. In general, the classifier diversity does not occur randomly, but is generated systematically by different ensemble creation methods. By using diverse data subsets to train classifiers, the ensemble creation methods can create diverse classifiers for the EoC. In this work, we propose a scheme to measure the data diversity directly from random subspaces and we explore the possibility of using the data diversity directly to select the best data subsets for the construction of the EoC. The applicability is tested on UCI machine learning problems and NIST SD19 handwritten numerals.

6.1 Introduction

In general, the classifiers created are stored in a pool of classifiers, however not all the classifiers in this pool will be useful. To select the most pertinent classifiers from the pool (5; 11; 61; 66; 80; 89; 101), we need to define an adequate objective function. This objective function can be a fusion function, like the majority voting error (11; 66; 80; 89), or simply the diversity among classifiers (30; 73).

The two key issues that are crucial to the success of an EoC routine are the following: first, we need diversity for ensemble creation, because an EoC will not perform well without it (56; 63; 66; 88; 89); and second, we need to select classifiers once they have been created (11; 63; 66; 89), because not all the classifiers created are useful. However, the routine: ensemble creation first, then ensemble selection, has some disadvantages, one of them being additional classifier training. Since not all the classifiers created will be used, time is spent in training classifiers that will not ultimately be used. Another is the evaluation of high dimensional classifier combinations, since we need to evaluate different

combinations of classifiers for ensemble selection after classifier training, and this evaluation will be very time-consuming in a large classifier pool. Hence, our question: Can we select data subsets for ensemble creation directly, instead of performing the ensemble creation/ensemble selection routine?

We assume that data subset selection might be feasible through the evaluation of the data diversity of data subsets. We thus propose a data subset selection for the Random Subspaces ensemble generation method (See appendix 1). Note that with this method data points might have relatively different distributions in the feature subspaces. This means that, by clustering these data points in different feature subspaces, we might have quite diverse clustering partitions. Since clustering diversities measure the diversity of these partitions, they give an indirect indication of the data diversity of the feature subspaces.

Here, we need to clarify the concept of clustering diversity. In general, it is meant to help in the construction of a cluster ensemble, and has nothing to do with classifiers. A cluster ensemble combines the results of several partitions and thus improves the quality and robustness of partitions of data (17; 23; 24; 26; 67; 79; 82; 95; 97; 98). It has been shown that more diverse cluster ensembles offer the potential for greater improvement than do less diverse cluster ensembles (23), and that is why we use clustering diversity in our study.

Given a pool of feature subsets, we use a clustering algorithm with fixed parameters to form clusterings in feature subsets (Fig. 33). It is reasonable to assume that clustering diversity between different feature subsets also indicates their data diversity (See appendix 5 and 6). This scheme will provide us with the following advantages:

a. By selecting the useful feature subsets, we can reduce the time needed for classifier training for ensemble creation.

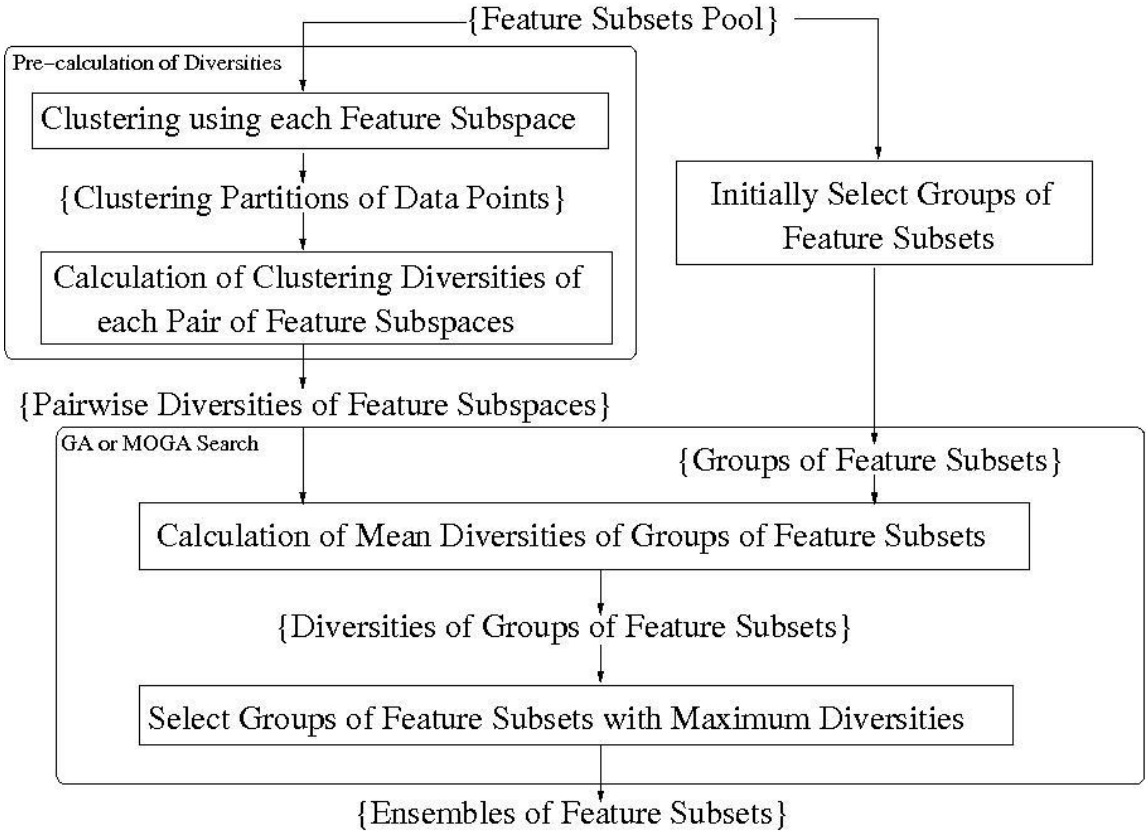

Figure 33 The proposed classifier-free ensemble selection scheme is, in fact, a feature subset selection in Random Subspaces. We carried out this feature subset selection using clustering diversity as objective function. Note that the pre-calculation of diversities is carried out once for all, while GA or MOGA search are repeated from generation to generation

b. By evaluating the pertinent feature subsets, we can significantly reduce the search space for ensemble selection.

c. Feature subset selection might be able to replace ensemble selection completely for Random Subspaces in some circumstances, and offers de facto classifier-free ensemble selection.

Our experimental results suggest that there is a strong correlation between classifier diversity and clustering diversity in Random Subspaces, and that clustering diversity does

work for a classifier-free ensemble selection scheme. Here, we need to mention that the proposed strategy would not work for the Bagging and Boosting ensemble generation methods. Since Bagging and Boosting draw a certain proportion of the data points to train classifiers, it is quite possible that the distributions of data points are rather similar. Consequently, clustering these data points might not generate significantly different clustering partitions. More importantly, since Bagging uses various data points for each classifier, it is impossible for us to measure data diversity by clustering different parts of data points.

In the next section, we introduce general clustering diversity measures. In section 3, we investigate the possibility of ensemble selection using clustering diversity measures on the UCI machine learning repository. In section 4, we report the experiments we performed on NIST SD19 handwritten numeral digits. Discussion is provided in section 5 and our conclusion comprises the last section.

6.2 Clustering Diversity Measures

In general, given two clustering partitions, we can apply clustering diversity to measure the diversity between the partitions. Since there is no class label available in clustering, the concept of diversity based on correct/incorrect classification cannot be applicable for clustering diversity, and another kind of approach will be needed. First, we introduce the concept of clustering diversity from the framework defined in (72). For C data points, suppose one clustering C_i groups these data points into I clusters, and another clustering C_k groups them into K clusters, then the diversity between these two clusterings can be deduced as follows:

6.2.1 Basic Concept of Clustering Diversity

For two clusterings, consider a contingency table (or confusion matrix) M as a $I \times K$ matrix which describes the partitions of data points in these two clusterings. Consider the $ik_{th}, 1 \leq i \leq I, 1 \leq k \leq K$ element in the contingency table M - let us call it block M_{ik}-

which represents data points grouped as a cluster by clustering C_i and also groups as a cluster by clustering C_k. In other words, all the data points that are grouped into cluster c_i by clustering C_i and grouped into cluster c_k by clustering C_k are located in the block M_{ik}. So, in this contingency table M, we can denote the number of data points in block M_{ik} as m_{ik}:

$$m_{ik} = |c_i \bigcap c_k| \tag{6.1}$$

$$\sum_{1 \le i \le I} \sum_{1 \le k \le K} m_{ik} = C \tag{6.2}$$

We note that, given two clusterings, the complexity of the calculation of all m_{ik} is $O(C \cdot (I + K))$. Once we have every element m_{ik} for contingency table M, we can use m_{ik} to calculate the clustering diversity between clustering C_i and clustering C_k. Given that we have C data points, we want to determine the relationship between these $\frac{C \cdot (C-1)}{2}$ data point pairs. We then classify the relationship of these $\frac{C \cdot (C-1)}{2}$ data point pairs into four different cases and count the numbers of occurrences of these cases:

a. C_{11}: the number of data point pairs that are in the same cluster under both C_i and C_k

b. C_{00}: the number of data point pairs that are in different clusters under both C_i and C_k

c. C_{10}: the number of data point pairs that are in the same cluster under C_i, but not under C_k

d. C_{01}: the number of data point pairs that are in the same cluster under C_k, but not under C_i

Suppose that we have C points in total, then the following condition must be satisfied :

$$C_{11} + C_{00} + C_{10} + C_{01} = \frac{C(C-1)}{2} \tag{6.3}$$

To illustrate the meanings of C_{ij} in Fig. 34 and Fig. 35, we carried out 2 clusterings on 4 data points. Note that these 4 data points mean 6 data point pairs. In Fig. 36, $C_{11} = 1$, because the triangle and the rectangle are grouped together in the same clusters by both clusterings. $C_{10} = 2$, because the star is grouped in the same cluster as the triangle and the rectangle by one clustering, but into different clusters by another clustering. By a similar analysis, we can observe that $C_{10} = 0$. $C_{00} = 3$, because the ellipse is considered to be in a different cluster from the star, the triangle and the rectangle by both clusterings.

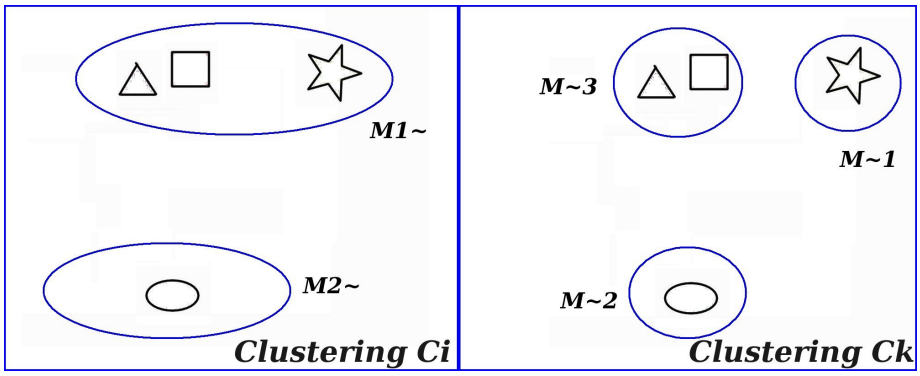

Figure 34 Illustration of 2 clustering partitions. The first clustering generates 2 partitions and the second clustering generates 3 partitions

While the direct calculation of $C_{11}, C_{00}, C_{10}, C_{01}$ could be very time-consuming - the complexity is $O(\frac{C(C-1)}{2})$ - this calculation can be greatly accelerated. In fact, all the values $C_{11}, C_{00}, C_{10}, C_{01}$ can be quickly derived from the contingency table M using its element m_{ik}.

Suppose there are m_{ik} data points in block M_{ik}, then we can calculate the C_{11} value as the data point pairs in this block, i.e. $C_{11}(M_{ik}) = \frac{m_{ik}(m_{ik}-1)}{2}$. Consequently, the total

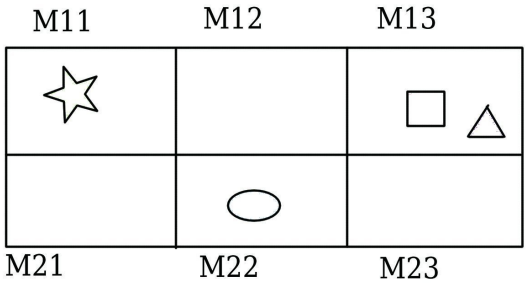

Figure 35 The 2 partitions of the first clustering can be denoted as (M_{1k} and M_{2k}), and those of the second clustering can be denoted as (M_{i1}, M_{i2} and M_{i3}). All data points are classified into M_{ik} based on these partitions

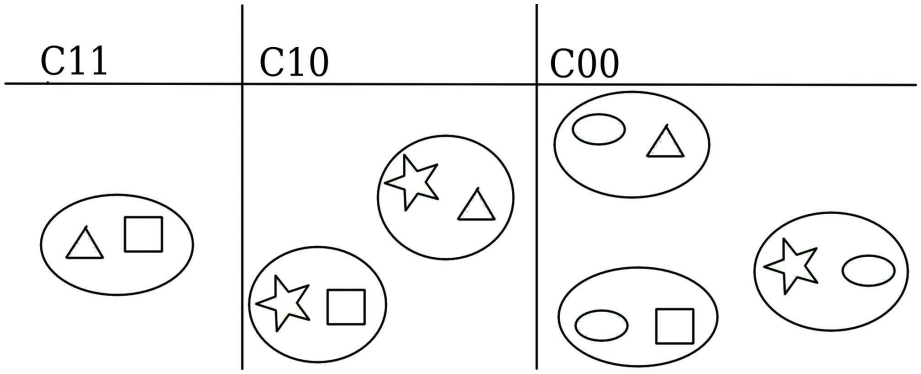

Figure 36 Examples of the calculation of $C_{11}, C_{00}, C_{10}, C_{01}$ based on 4 data points and thus 6 data point pairs

C_{11} value can be calculated as the sum of $C_{11}(M_{ik})$ from all these blocks, i.e. $C_{11} = \sum_{1 \leq i \leq I} \sum_{1 \leq k \leq K} C_{11}(M_{ik})$:

$$C_{11} = \sum_{1 \leq i \leq I} \sum_{1 \leq k \leq K} \frac{m_{ik}(m_{ik} - 1)}{2} \tag{6.4}$$

Using the eq. 6.2, we can write :

$$C_{11} = \frac{(\sum_{1 \leq i \leq I} \sum_{1 \leq k \leq K} m_{ik}^2) - C}{2} \tag{6.5}$$

For C_{10} and C_{01}, the calculation follows the same principle. It can be deduced that there are $\frac{((\sum_i m_{ik})-m_{ik})}{2}$ data point pairs grouped in the same cluster by clustering C_i, but in different clusters by clustering C_k. Consequently, we can arrive at a value for C_{10}.

$$C_{10} = \frac{\sum_{1\leq i\leq I}\sum_{1\leq k\leq K} m_{ik}((\sum_i m_{ik}) - m_{ik})}{2} \tag{6.6}$$

For C_{01}, we can use the same method and get similar result.

$$C_{01} = \frac{\sum_{1\leq i\leq I}\sum_{1\leq k\leq K} m_{ik}((\sum_k m_{ik}) - m_{ik})}{2} \tag{6.7}$$

The more complicated case is the deduction of C_{00}, for which we should look for data point pairs that are grouped in different clusters by both C_i and C_k clustering. Since there are $(C - \sum_k m_{ik} - \sum_i m_{ik} + m_{ik})$ samples satisfying this condition, we can arrive at :

$$C_{00} = \frac{\sum_{1\leq i\leq I}\sum_{1\leq k\leq K}(m_{ik}\cdot(C - \sum_k m_{ik} - \sum_i m_{ik} + m_{ik}))}{2} \tag{6.8}$$

The result can be verified by calculating $C_{11} + C_{10} + C_{01} + C_{00} = \frac{C(C-1)}{2}$.

Remember that the complexity of the calculation of all m_{ik} is $O(C\cdot(I+K))$. Given that $I, K \ll C$, the calculation of $C_{11}, C_{00}, C_{10}, C_{01}$ deduced by m_{ik} is much faster than the direct calculation of $C_{11}, C_{00}, C_{10}, C_{01}$, which had the complexity of $O(\frac{C(C-1)}{2})$.

We need to mention that we fix all the clustering parameters, including the number of clusters. In other words, in our case, $I = K$, and the contingency table M is, in fact, a square matrix.

However, these four types of relationships of data point pairs are not themselves clustering diversity measures. In fact, several different clustering diversity measures have been proposed using the counts of these four cases. We introduce them in the next section.

6.2.2 Pairwise Clustering Diversity Measures

Based on the pairwise counts, a number of clustering diversity measures are proposed (72):

a. Wallace Indices

$$Wallace - 1 : W_i(C_i, C_k) = \frac{C_{11}}{C_{11} + C_{10}} \tag{6.9}$$

$$Wallace - 2 : W_k(C_i, C_k) = \frac{C_{11}}{C_{11} + C_{01}} \tag{6.10}$$

b. Fowlkes-Mallows Index

$$F(C_i, C_k) = \frac{C_{11}}{((C_{11} + C_{10})(C_{11} + C_{01}))^{\frac{1}{2}}} = (W_i(C_i, C_k)W_k(C_i, C_k))^{\frac{1}{2}} \tag{6.11}$$

c. Rand Index

$$R(C_i, C_k) = \frac{C_{11} + C_{00}}{\frac{C(C-1)}{2}} \tag{6.12}$$

d. Jacard Index

$$J(C_i, C_k) = \frac{C_{11}}{C_{11} + C_{01} + C_{10}} \tag{6.13}$$

e. Mirkin's Metric

$$K(C_i, C_k) = 2(C_{10} + C_{01}) = C(C-1)[1 - R(C_i, C_k)] \tag{6.14}$$

Note that all these measures calculate the clustering diversity between two clusterings. In the case where there are more than two clusterings, the global clustering diversity is simply the mean of all clustering diversities between all clustering pairs. Given L clusterings, there are $\frac{L\times(L-1)}{2}$ clustering diversities $d_{12}, d_{13}, ..., d_{(L-1)L}$ to be calculated, and the global clustering diversity $\bar{d}$ will be its average :

$$\bar{d} = 2 \times \frac{\sum_{ij} d_{ij}}{L \times (L-1)}, i \leq j \tag{6.15}$$

Now we want to check whether or not the clustering diversity of different feature subsets can be used as an objective function for classifier-free ensemble selection, and so we carried out the experiments on the UCI machine learning problems (see below).

6.3 Evaluation of Objective Functions for Ensemble Selection on the UCI Machine Learning Repository

First, we need to evaluate the hypothesis that the clustering diversity of different feature subsets can be used as an objective function for ensemble selection in Random Subspaces. For an ensemble created with the Random Subspaces method, we first evaluated its feature subspaces by carrying out simple K-Means clusterings with predefined numbers of clusters on these feature subsets. The number of clusters is preselected using the Xie-Beni index (XB index) (4; 45) as the clustering validity index. A clustering diversity was thus calculated based on the clusterings of these feature subsets, and served as an objective function for the search. Six various clustering diversities were tested in our experiment, including: Mirkin's Metric, two Wallace Indices, the Fowlkes-Mallows Index, the Rand Index and the Jacard Index. As we mentioned in the introduction, the search algorithm is also an important issue for ensemble selection. For the classifier-free ensemble selection scheme, we evaluate two types of search algorithms: the single genetic algorithm (GA) and the multi-objective genetic algorithm (MOGA). We used the GA because, as a population-based search algorithm, it is flexible and its complexity can be adjusted ac-

cording to the size of the population and the number of generations. Moreover, because the algorithm returns a population of the best combinations, it can be potentially exploited to prevent generalization problems (89). Once the feature subsets had been selected, we constructed corresponding classifiers using the selected feature subsets and evaluated the performance of the ensembles of these classifiers (see Fig. 37).

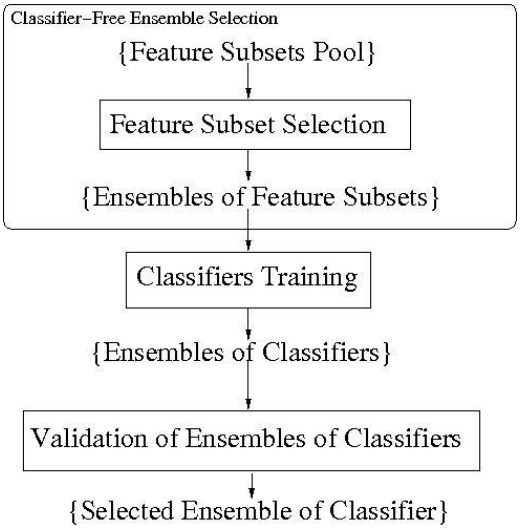

Figure 37 The processing steps of the proposed classifier-free ensemble selection method. The selected ensembles of feature subsets can be used to train ensembles of classifiers. These ensembles must be tested in a validation set in order to select the best ensemble. The detailed part of "feature subset selection" is shown on Fig. 33

At the same time, we need to compare our classifier-free ensemble selection scheme with traditional classifier-based ensemble selection methods. For traditional classifier-based ensemble selection, each feature subset was used to train a classifier, and all the trained classifiers were stored in a pool. In order to select adequate classifiers from this pool, we carried out the ensemble selection process using majority voting error (MVE) as the objective function for the GA and MOGA search algorithms.

We performed the classifier-free ensemble selection and classifier-based ensemble selection experiments on UCI machine learning problems (Table XXIX). Three classification algorithms were used: Quadratic Discriminant Classifiers QDC), K-Nearest Neighbors Classifiers (KNN) and Parzen Windows Classifiers (PWC) (19) for the classification tasks.

Table XXIX

The problems extracted from the UCI Machine Learning Data Repository

database	number of classes	number of clusters	number of train samples	number of test samples	number of features	number of cardinality
Pima-Diabetes	2	3	384	384	8	4
Liver-Disorders	2	5	144	144	6	3
Wisconsin Breast-Cancer	2	12	284	284	30	5
Wine	3	4	88	88	13	6
Image Segmentation	7	53	210	2100	19	4
Letters Recognition	26	87	10000	10000	16	12

All the problems extracted from the UCI have two datasets, a training set for classifier training for the GA or MOGA search, and a test set used only for testing. The whole training set was used to create 10 classifiers in Random Subspaces. Moreover, the training samples were divided into 3 parts for each scheme:

- Optimization set:

  70% of the training samples were used for the GA or MOGA search. These samples were clustered in feature subspaces, and the clustering diversity indices were measured by comparing clusterings in a pairwise manner. The diversity of a set of feature subspaces is calculated as the mean value of pairwise diversities of the features involved (eq. 6.15).

- Archive validation set:

  Another 15% of the training samples were used as the archive validation mechanism (86) to avoid overfitting during the GA or MOGA search. They were used to evaluate all the individuals and then to store the optimal solutions in a separate archive

after each generation (Fig. 38). The reason for using this archive validation mechanism is that solutions found in a pareto front of one dataset may be optimal only for this special search dataset. From generation to generation, the solutions found may tend to overfit the search dataset. To make sure that the solutions found were not overfitted in our case, we validated them in another archive validation set. The solutions are stored in the archive only if they dominate all solutions in the archive validation set.

- Classifier-free MOGA evaluation set:

  The last 15% of the training samples were used solely for the final classification performance validation for the classifier-free MOGA search. The reason for this was that, unlike the GA search, which gives the best individual in the population, a MOGA search gives a group of individuals, called a pareto front. As a result, we need a means to evaluate the solutions found in this pareto front. Even though a MOGA search is a purely classifier-free process, the evaluation of these potential solutions will require the construction of classifiers. So, during this process, the feature subset candidates stored in the archive are then used to construct ensembles and their performances evaluated on these samples.

- Test set:

  The best solutions found were evaluated on the test set.

The classifier-free GA search used the clustering diversities calculated from the optimization set to search for feature subspaces with the maximum clustering diversity. During the search, solutions found in each generation were evaluated with clustering diversity in the archive validation set and stored in an archive. Finally, solutions stored in the archive were used on a test set.

The classifier-free MOGA search follows the same procedure as the classifier-free GA search, except that the classifier-free MOGA search has two objective functions: max-

imization of clustering diversity and maximization of the number of feature subspaces. We will discuss in the next section the reason why the number of feature subspaces is to be maximized. Moreover, since the classifier-free MOGA search provides a group of solutions instead of one solution as in the classifier-free GA search, we needed to evaluate the solutions stored in the archive. We trained an EoC using subspaces found by the classifier-free MOGA search. These EoCs were then evaluated in a classifier-free MOGA evaluation set. The best ensemble was then used on a test set.

The classifier-based GA search first constructed all the classifiers using the training set, and then used mean ME or MVE evaluated on the optimization set to search for EoCs with the ME or MVE. Again, during the search, solutions found in each generation were evaluated in the archive validation set and stored in an archive. Finally, solutions stored in the archive were used on a test set.

The classifier-based MOGA search also constructed all the classifiers using the training set, and then used the ME or MVE evaluated on the optimization set to search for EoCs with the ME or MVE. However, in order to compare this search with the classifier-free MOGA search, it also used the maximization of the number feature subspaces as another objective function. Following the MOGA search, the best solution was selected as the individual at the pareto front with the minimum error rate. This solution was then used on a test set. Because the error rate had already been evaluated during the search, the classifier-based MOGA search did not need to use an external evaluation set for the final evaluation as was done in the classifier-free MOGA search.

We first carried out the experiments with a single GA search, and then we compared the results with those of a MOGA search.

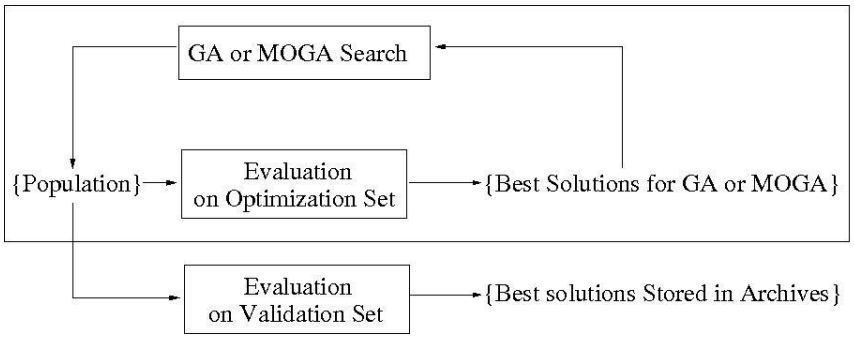

Figure 38 The archive validation set is used to validate the population found by GA or MOGA and then stores the best solutions in a separate archive

6.3.1 Search with the Single Genetic Algorithm

For classifier-free ensemble selection (or feature subset selection), we used different clustering diversity indices as objective functions to find the potentially adequate feature subsets. Among these objective functions, we minimized two Wallace indices, the Fowlkes-Mallows index, the Rand index, the Jacard index and the maximized Mirkin Metric. All the global clustering diversity measures are calculated as the mean values of clustering diversities between all clustering pairs. Note that the clustering diversity between any two clustering pairs can be calculated prior to the GA search, so that during the GA search we simply calculate the mean of the clustering diversities among selected clusterings. For each of 6 problems extracted from the UCI, 10 feature subsets with fixed cardinality are given as the pool for the search (see Table XXIX). Using the pre-calculated clustering diversities based on the clusterings with these feature subsets, the GA search evaluated the global diversity of various combinations of these feature subsets. The combination of these feature subsets with the best global diversity was regarded as the best solution, and then the selected feature subsets were used to construct the needed classifiers. These classifiers were then combined using the MAJ fusion function to give the classification results.

Table XXX

The average recognition rates of KNN classifiers selected by GA with different objective functions. The average ensemble sizes of MVE and ME are shown in the parenthesis

	Mirkin's	Wallace Index-1	Wallace Index-2	Fowlkes-Mallows	Rand
Pima-Diabetes	79.77 ±1.73 %	76.61 ± 1.74 %	77.37 ±1.85 %	78.32 ± 2.59 %	77.22 ± 2.85 %
Liver-Disorders	72.11 ± 2.45 %	70.35 ± 3.49 %	72.01 ± 3.06 %	70.39 ± 4.33 %	69.00 ± 3.68 %
Wisconsin Breast-Cancer	92.18 ± 0.70 %	89.19 ± 4.77 %	89.71 ± 4.21 %	89.67 ± 4.71 %	91.73 ± 0.84 %
Wine	75.61 ± 5.71 %	73.52 ± 1.98 %	73.60 ± 2.58 %	74.05 ± 3.70 %	71.82 ± 4.71 %
Image Segmentation	74.78 ± 2.31 %	76.87 ± 3.63 %	77.29 ± 2.96 %	78.28 ± 2.10 %	75.29±1.79 %
Letters Recognition	82.17 ±0.85 %	76.48 ± 3.36 %	78.11 ± 3.90 %	77.12 ± 4.33 %	77.85 ±3.35 %
	Jacard	M.V.E	M.E.	ALL	Oracle
Pima-Diabetes	81.35 ± 1.64 %	79.85 ± 2.36 % (3.97)	79.57 ± 2.20 % (3.83)	**82.55** ± 0.00 %	98.18 %
Liver-Disorders	72.11 ± 2.94 %	73.91 ± 2.89 % (4.07)	72.29 ± 2.73 % (3.67)	**76.39** ± 0.00 %	100.00 %
Wisconsin Breast-Cancer	91.97 ± 3.69 %	92.10 ± 1.98 % (3.73)	92.55 ± 0.85 % (4.20)	**92.61** ± 0.00 %	99.65 %
Wine	72.42 ± 2.29 %	72.50 ± 1.39 % (3.63)	75.00 ± 3.54 % (3.93)	**76.14** ± 0.00 %	97.73 %
Image Segmentation	78.47 ± 2.68 %	72.85 ± 1.42 % (4.03)	75.33 ± 4.21 % (3.97)	**78.19** ± 0.00 %	97.29 %
Letters Recognition	76.37 ± 3.80%	79.99 ± 2.27% (4.37)	79.25 ± 3.00% (3.90)	**83.08** ± 0.00 %	94.78 %

Table XXXI

The average recognition rates of QDC classifiers selected by GA with different objective functions. The average ensemble sizes of MVE and ME are shown in the parenthesis

	Mirkin's	Wallace Index-1	Wallace Index-2	Fowlkes-Mallows	Rand
Pima-Diabetes	76.05 ± 1.53 %	72.74 ± 2.56 %	74.84 ± 4.16 %	74.00 ± 2.80 %	72.86 ± 3.00 %
Liver-Disorders	59.51 ± 0.45 %	57.11 ± 2.67 %	58.12 ± 2.54 %	57.15 ± 3.34 %	59.91 ± 1.48 %
Wisconsin Breast-Cancer	95.21 ± 1.11 %	91.50 ± 2.03 %	92.50 ± 2.23 %	91.54 ± 1.15 %	93.22 ± 1.94 %
Wine	95.45 ± 1.08 %	95.76 ±1.26 %	93.98 ± 2.82 %	92.73 ± 3.55 %	92.84 ± 3.75 %
Image Segmentation	72.03 ± 15.40 %	69.85 ± 13.19 %	67.59 ± 15.43 %	74.34 ± 9.29 %	72.89 ± 12.09 %
Letters Recognition	82.53 ± 0.97 %	82.71 ± 1.03 %	82.36 ± 1.11 %	82.57 ± 1.50 %	82.71 ± 0.88%
	Jacard	M.V.E	M.E.	ALL	Oracle
Pima-Diabetes	75.92 ± 1.60 %	75.49 ± 2.46 % (4.30)	74.34 ± 2.65 % (3.83)	**77.86** ± 0.00 %	93.23 %
Liver-Disorders	58.63 ± 2.01 %	57.15 ± 2.26% (4.23)	56.99 ± 2.70 % (4.17)	**57.64** ± 0.00%	88.19 %
Wisconsin Breast-Cancer	91.55 ± 1.40 %	93.57 ± 2.06 % (3.80)	93.69 ± 1.48 % (4.07)	**93.66** ± 0.00 %	99.65 %
Wine	93.30 ± 3.71 %	92.61 ± 1.75 % (4.43)	95.00 ± 2.44 % (4.00)	**96.59** ± 0.00 %	100.00 %
Image Segmentation	73.23 ± 12.31 %	60.59 ± 12.92% (3.80)	57.27 ± 15.65 % (4.20)	**78.24** ± 0.00 %	95.29 %
Letters Recognition	82.46 ± 1.52 %	81.13 ± 2.37% (3.80)	84.10 ± 0.00 % (9.00)	**84.36** ± 0.00 %	93.40 %

In order to compare the performance of the classifier-free approach with the traditional classifier-based approach, we also evaluated the single GA search with MVE and with ME as the objective functions. For these two schemes, classifiers were constructed using given

Table XXXII

The average recognition rates of the ensembles of PARZEN WINDOWS classifiers selected by GA with different objective functions. The average ensemble sizes of MVE and ME are shown in the parenthesis

	Mirkin's	Wallace Index-1	Wallace Index-2	Fowlkes-Mallows	Rand
Pima-Diabetes	**78.28** ± 1.52 %	73.87 ± 2.94 %	77.87 ± 2.56 %	76.22 ± 3.67 %	75.44 ± 3.16 %
Liver-Disorders	70.02 ±2.06 %	61.34 ± 2.95 %	63.54 ± 4.06 %	62.85 ± 5.17 %	68.12 ± 3.30 %
Wisconsin Breast-Cancer	90.77 ± 1.14 %	90.16 ± 1.12 %	89.51 ± 1.51 %	90.18 ±1.48 %	90.96 ± 0.31 %
Wine	**81.40** ± 4.89 %	76.74 ± 2.31 %	75.80 ± 3.06 %	76.63 ± 3.79 %	75.72 ± 5.32 %
Image Segmentation	74.91 ± 4.20 %	72.68 ± 7.67 %	76.89 ± 2.68 %	76.73 ± 5.98 %	72.51± 7.72%
Letters Recognition	89.00 ± 0.52 %	88.46 ± 1.05 %	88.23 ± 1.01 %	88.37 ± 1.26 %	88.54 ±0.76 %
	Jacard	M.V.E	M.E.	ALL	Oracle
Pima-Diabetes	78.31 ± 1.75 %	77.74 ± 2.21 % (4.13)	78.19 ± 1.88 % (4.03)	78.12 ± 0.00 %	92.19 %
Liver-Disorders	63.06 ± 4.94 %	66.76 ±4.07 % (3.80)	67.87 ± 3.77% (4.07)	**70.83** ± 0.00 %	89.58 %
Wisconsin Breast-Cancer	90.85 ± 1.18 %	90.99 ± 1.39 % (4.10)	87.88 ± 1.66 % (3.87)	**91.55** ± 0.00 %	98.94 %
Wine	76.14 ± 4.29 %	79.47 ± 4.25 % (3.97)	79.36 ± 5.07 % (4.23)	76.14 ± 0.00 %	100.00 %
Image Segmentation	79.61 ± 4.43 %	75.60 ± 5.13 % (4.57)	75.31 ± 4.97 % (4.13)	**79.62** ± 0.00 %	98.48 %
Letters Recognition	88.41 ± 1.34 %	87.00 ± 1.68 % (3.80)	89.29 ± 0.00 % (9.00)	**89.52** ± 0.00 %	96.70 %

feature subset pools, and the GA search evaluated the results directly from the classifier outputs, regardless of the clustering diversities of their feature subsets. For MVE, the ensembles were selected for the minimum ensemble errors; and for ME, the ensembles were chosen for the minimum average of the individual classifier error. All classifiers were combined using MAJ as the fusion function.

For the single GA search, we set 32 individuals in the population with 500 generations. The mutation rate was set to $\frac{1}{L}$, where L is the length of the mutated binary string (21), and the crossover probability was set to 50%. A threshold of 3 classifiers was applied as the minimum number of classifiers for the EoC during the whole search. The experiments were repeated 30 times for statistical evaluation.

We note that, in general, the MVE, and even the ME, have much better performances than all the other clustering diversity indices (Table XXX $\sim$ XXXII). This is not surprising, since the clustering diversity indices do not take into account the classifier outputs. In our experiments, ME does not converge into the minimum ensemble size, but we found

that several ensembles can achieve the same ME, which explains why ME could have ensemble sizes that are larger than the minimum. This is reasonable, because the pool consists of only 10 classifiers. Moreover, given that all GA searches with the clustering diversity indices converge to the minimum number of classifiers (fixed to 3 classifiers in our experiments), it is understandable that the single GA search with the clustering diversity indices underperforms.

Given that we are not only looking for the optimum performances from these clustering diversity indices, but also a pre-selection for the more refined ensemble selection methods, this convergence of the single GA is not desirable. In order to resolve the problem of convergence into the minimum ensemble size, we carried out a MOGA search in our next experiment.

6.3.2 Search with the Multi-Objective Genetic Algorithm

As we can observe from the single GA search, the use of pairwise diversity as an objective function has a technical problem: the search algorithm will converge to the minimum number of feature subsets (and hence the minimum size of the ensemble) with the maximum clustering diversity, which means that the search algorithm systematically prefers the smaller ensembles to bigger ones (58). It turns out that, in effect, we will encounter two problems if we use pairwise diversities. So, aside from optimizing the diversity, we should, at the same time, avoid minimizing the number of feature subsets.

Given the challenges posed by ensemble selection, the prospect of satisfying multi-objective problems makes the MOGA a desirable alternative. We thus define two objectives for the search: the optimization of diversity (and hence the minimization of two Wallace indices, the Fowlkes-Mallows index, the Rand index, the Jacard index and the maximization of Mirkin's Metric) and the maximization of the number of feature subsets. Although we only care about diversity, maximizing the number of feature subsets can pre-

vent the search from converging into the minimal number of feature subsets (and hence the minimum size of the ensemble).

Table XXXIII

The average recognition rates of the ensembles of KNN classifiers selected by MOGA with different objective functions on problems extracted from the UCI machine learning repository

	Mirkin's	Wallace Index-1	Wallace Index-2	Fowlkes-Mallows	Rand
Pima-Diabetes	80.10 ± 2.03 %	77.87 ± 1.18 %	79.07 ± 2.56 %	79.96 ± 1.77 %	79.13 ± 1.90 %
Liver-Disorders	72.78 ± 2.97 %	74.08 ± 2.83 %	74.26 ± 2.53 %	71.93 ± 3.54 %	72.94 ± 3.10 %
Wisconsin Breast-Cancer	92.28 ± 1.82 %	**92.78** ± 1.96 %	92.18 ± 1.26 %	92.30 ± 2.05 %	91.99 ± 2.01 %
Wine	74.47 ± 2.40 %	74.94 ± 2.30 %	74.33 ± 1.67 %	75.58 ± 3.51 %	75.44 ± 3.63 %
Image Segmentation	74.80 ± 5.08 %	75.47 ± 4.66 %	75.04 ± 3.60 %	75.72 ± 3.03 %	74.89 ± 3.68 %
Letters Recognition	79.13 ± 2.92 %	80.10 ± 2.74 %	80.45 ± 1.29 %	80.89 ± 1.48 %	78.98 ± 3.50 %
	Jacard	M.V.E	M.E.	ALL	Oracle
Pima-Diabetes	79.91 ± 1.87 %	79.33 ± 2.12 %	79.48 ± 2.06 %	**82.55** ± 0.00 %	98.18 %
Liver-Disorders	74.01 ± 2.47 %	74.07 ± 3.56 %	73.79 ± 2.92 %	**76.39** ± 0.00 %	100.00 %
Wisconsin Breast-Cancer	88.87 ± 1.79 %	92.48 ± 0.95 %	92.46 ± 1.28 %	92.61 ± 0.00 %	99.65 %
Wine	**76.29** ± 3.04 %	75.51 ± 2.84 %	74.27 ± 2.74 %	76.14 ± 0.00 %	97.73 %
Image Segmentation	75.55 ± 4.94 %	74.16 ± 3.67 %	74.11 ± 4.00 %	**78.19** ± 0.00 %	97.29 %
Letters Recognition	80.10 ± 2.14 %	80.30 ± 2.29 %	77.59 ± 3.82 %	**83.08** ± 0.00 %	94.78 %

Table XXXIV

The average ensemble sizes of KNN classifiers selected by MOGA with different objective functions on problems extracted from the UCI machine learning repository

	Mirkin's	Wallace Index-1	Wallace Index-2	Fowlkes-Mallows	Rand
Pima-Diabetes	4.33	4.27	4.33	5.00	4.02
Liver-Disorders	3.69	4.29	4.16	4.06	4.27
Wisconsin Breast-Cancer	3.92	4.12	3.70	4.19	4.24
Wine	4.47	4.28	3.66	4.47	3.93
Image Segmentation	3.67	4.31	4.50	4.47	4.33
Letters Recognition	4.00	4.00	4.31	4.47	3.67
	Jacard	M.V.E	M.E.	ALL	
Pima-Diabetes	4.43	4.16	4.29	10.00	
Liver-Disorders	3.99	4.02	3.95	10.00	
Wisconsin Breast-Cancer	4.23	4.26	3.87	10.00	
Wine	4.83	4.24	3.60	10.00	
Image Segmentation	4.83	4.24	3.60	10.00	
Letters Recognition	4.39	4.21	3.38	10.00	

Table XXXV

The average recognition rates of the ensembles of QDC classifiers selected by MOGA with different objective functions on problems extracted from the UCI machine learning repository

	Mirkin's	Wallace Index-1	Wallace Index-2	Fowlkes-Mallows	Rand
Pima-Diabetes	75.89 ± 2.62 %	75.08 ± 3.48 %	76.03 ± 2.20 %	74.97 ± 2.65 %	74.69 ± 2.68 %
Liver-Disorders	56.88 ± 2.50 %	57.41 ± 2.31 %	56.93 ± 2.24 %	57.17 ± 3.13 %	57.56 ± 3.06 %
Wisconsin Breast-Cancer	93.62 ± 2.01 %	93.93 ± 1.65 %	**94.36** ± 1.43 %	93.60 ± 2.01 %	93.48 ± 1.69 %
Wine	95.81 ± 2.59 %	96.20 ± 0.97 %	92.74 ± 1.63 %	95.27 ± 2.44 %	95.61 ± 1.93 %
Image Segmentation	50.67 ± 23.37 %	57.84 ± 15.54 %	63.78 ± 13.54 %	61.60 ± 13.05 %	64.78 ± 15.23 %
Letters Recognition	80.79 ± 2.41 %	81.85 ± 2.10 %	82.10 ± 1.78 %	81.98 ± 1.19 %	81.16 ± 1.60 %
	Jacard	M.V.E	M.E.	ALL	Oracle
Pima-Diabetes	75.68 ± 2.07 %	75.62 ± 2.68 %	74.58 ± 2.56 %	**77.86** ± % 0.00	93.23 %
Liver-Disorders	56.77 ± 2.38 %	56.53 ± 2.32 %	57.46 ± 2.33 %	**57.64** ± % 0.00	88.19 %
Wisconsin Breast-Cancer	91.46 ± 1.41 %	94.02 ± 1.70 %	93.67± 1.81 %	93.66 ± 0.00 %	99.65 %
Wine	95.48 ± 1.11 %	95.14 ± 2.86 %	95.11± 2.10 %	**96.59** ± % 0.00	100.00 %
Image Segmentation	52.20 ± 18.43 %	59.11 ± 12.58 %	57.20 ± 11.25 %	**78.24** ± % 0.00	95.29 %
Letters Recognition	81.76 ± 2.06 %	81.50 ± 1.67 %	81.27± 1.80 %	**84.36** ± % 0.00	93.40 %

Table XXXVI

The average ensemble sizes of QDC classifiers selected by MOGA with different objective functions on problems extracted from the UCI machine learning repository

	Mirkin's	Wallace Index-1	Wallace Index-2	Fowlkes-Mallows	Rand
Pima-Diabetes	4.31	4.12	4.49	4.30	3.94
Liver-Disorders	3.86	4.13	4.02	4.62	3.90
Wisconsin Breast-Cancer	3.92	4.15	3.57	3.94	4.10
Wine	4.35	4.22	3.85	4.29	3.78
Image Segmentation	3.16	4.41	4.50	4.25	4.55
Letters Recognition	3.79	4.08	4.61	4.62	3.84
	Jacard	M.V.E	M.E.	ALL	
Pima-Diabetes	4.42	4.16	4.56	10.00	
Liver-Disorders	4.19	4.38	3.93	10.00	
Wisconsin Breast-Cancer	4.20	3.81	4.11	10.00	
Wine	4.53	4.35	3.95	10.00	
Image Segmentation	3.48	3.81	3.72	10.00	
Letters Recognition	4.43	4.14	3.81	10.00	

We used the MOGA as the search algorithm, with 32 individuals in the population and 500 generations. The mutation rate was set to $\frac{1}{L}$, where L is the length of the mutated binary string (21), and the crossover probability was set to 50%. For both classifier-free ensemble

Table XXXVII

The average recognition rates of the ensembles of PARZEN WINDOWS classifiers selected by MOGA with different objective functions on problems extracted from the UCI machine learning repository

	Mirkin's	Wallace Index-1	Wallace Index-2	Fowlkes-Mallows	Rand
Pima-Diabetes	**78.49** $\pm$ 1.56 %	75.00 $\pm$ 1.14 %	77.12 $\pm$ 2.58 %	78.18 $\pm$ 1.13 %	77.73 $\pm$ 2.02 %
Liver-Disorders	68.66 $\pm$ 3.15 %	68.18 $\pm$ 3.52 %	68.29 $\pm$ 4.39 %	67.77 $\pm$ 3.90 %	67.55 $\pm$ 4.23 %
Wisconsin Breast-Cancer	90.83 $\pm$ 1.22 %	90.98 $\pm$ 1.08 %	90.86 $\pm$ 1.03 %	91.16 $\pm$ 1.22 %	90.25 $\pm$ 1.48 %
Wine	76.52 $\pm$ 1.61 %	79.06 $\pm$ 4.43 %	79.96 $\pm$ 1.35 %	78.60 $\pm$ 4.51 %	79.62 $\pm$ 5.08 %
Image Segmentation	75.53 $\pm$ 5.62 %	75.74 $\pm$ 5.42 %	76.33 $\pm$ 5.24 %	76.61 $\pm$3.28 %	75.79 $\pm$ 5.10 %
Letters Recognition	86.88 $\pm$ 2.13 %	87.39 $\pm$ 1.96 %	87.70 $\pm$ 1.03 %	87.74 $\pm$ 1.14 %	86.83 $\pm$ 2.06%
	Jacard	M.V.E	M.E.	ALL	Oracle
Pima-Diabetes	77.57 $\pm$ 2.33 %	76.45 $\pm$ 2.78 %	77.62 $\pm$ 1.92 %	78.12 $\pm$ 0.00 %	92.19 %
Liver-Disorders	68.11 $\pm$ 3.55 %	68.23 $\pm$ 2.96 %	68.39$\pm$ 3.50 %	**70.83** $\pm$ 0.00 %	89.58 %
Wisconsin Breast-Cancer	88.23 $\pm$ 1.47 %	91.27 $\pm$ 1.30 %	90.89 $\pm$ 1.34 %	**91.55** $\pm$ 0.00 %	98.94 %
Wine	78.66 $\pm$ 4.32 %	78.45 $\pm$ 4.10 %	**80.02**$\pm$ 4.29 %	76.14 $\pm$ 0.00 %	100.00 %
Image Segmentation	77.63 $\pm$ 5.86 %	75.94 $\pm$ 4.13 %	76.83$\pm$ 4.71 %	**79.62** $\pm$ 0.00 %	98.48 %
Letters Recognition	87.46 $\pm$ 1.49%	87.26 $\pm$ 1.61 %	87.45 $\pm$ 1.01 %	**89.52** $\pm$ 0.00 %	96.70 %

Table XXXVIII

The average ensemble sizes of PARZEN WINDOWS classifiers selected by MOGA with different objective functions on problems extracted from the UCI

	Mirkin's	Wallace Index-1	Wallace Index-2	Fowlkes-Mallows	Rand
Pima-Diabetes	4.48	3.75	4.42	4.89	4.09
Liver-Disorders	3.98	4.30	4.11	4.45	3.84
Wisconsin Breast-Cancer	4.06	4.17	3.65	4.19	4.10
Wine	4.58	4.17	3.80	4.21	3.86
Image Segmentation	3.41	4.32	4.44	4.46	4.70
Letters Recognition	4.11	3.95	4.28	4.11	3.93
	Jacard	M.V.E	M.E.	ALL	
Pima-Diabetes	4.18	4.05	4.13	10.00	
Liver-Disorders	4.10	5.02	4.06	10.00	
Wisconsin Breast-Cancer	4.34	3.97	4.03	10.00	
Wine	4.71	3.78	4.02	10.00	
Image Segmentation	4.23	3.93	4.48	10.00	
Letters Recognition	4.23	4.31	4.19	10.00	

selection (or feature subset selection) and classifier-based ensemble selection, a threshold of 3 feature subsets or classifiers was applied as the minimum number of feature subsets or classifiers, and the experiments were repeated 30 times.

Note that the MOGA solutions are non-dominated (known as Pareto-optimal) solutions. In order to approach these solutions, we applied a non-dominated Sorting Genetic Algorithm (NSGA2), developed by Deb (13). NSGA2 maintains the dual objective of the MOGA by using a fitness assignment scheme, which prefers non-dominated solutions, and a crowded distance strategy, which preserves diversity among the solutions of each non-dominated front.

First, we note that the MOGA search based on clustering diversity indices gives a larger population than the single GA does for classifier-free ensemble selection (Table XXXIV, XXXVI, XXXVIII). Although their population sizes are larger, the feature subsets selected with the MOGA generally, but not always, perform better than those selected with the single GA (Table XXXIX).

Table XXXIX

The significance p value of the recognition rates between classifier-free MOGA search and classifier-free GA search

	Pima -Diabete	Liver -Disorder	Wisconson Breast Cancer	Wine	Image Segmentation	Letter Recognition
KNN	1e-06	1e-07	2e-09	8e-04	2e-09	6e-04
QDC	2e-09	0.0829	0.2513	2e-09	1e-09	2e-09
PWC	2e-09	0.3482	0.1891	8e-04	2e-09	2e-09

By contrast, the MOGA search based on ME or MVE does not perform better than the single GA search for classifier-based ensemble selection. This is understandable, since ME or MVE benefit directly from the classifier outputs, with the result that the maximum ensemble size does not help much in improving the results.

Interestingly, we observe that, with the MOGA search, most objective functions, including clustering diversities for classifier-free ensemble selection and ME and MVE for classifier-based ensemble selection, gave similar performances (Table XXXIII, XXXV, XXXVII). The reasonably small standard deviations indicate that their performances are quite stable

in different replications. There seems to be no index which is apparently best for classifier-free ensemble selection and for classifier-based ensemble selection. The best solutions seem to be problem-dependent. According to the 'no free lunch' theorem (105; 106), there is no single search algorithm that will always be the best for all problems. This phenomenon can be observed in our experiments.

Although the experiments suggest that the MOGA scheme for classifier-free ensemble selection might be applicable in Random Subspaces, the problems extracted from the UCI Machine Learning Repository usually consist of a small number of samples in low feature dimensions. Furthermore, given the constrained feature space dimensions, the classifier pool is composed of only 10 classifiers in our experiment. These constraints make the result less convincing, although we believe that the MOGA scheme for classifier-free ensemble selection might offer more advantages in a more complex problem with a larger classifier pool. We thus carried out a larger-scale experiment on a problem with more features and larger classifier pools, and hence the next experiment on a 10-class handwritten-numeral problem with 132 features and 100 classifiers.

6.4 Evaluation of Objective Functions for Ensemble Selection on a Handwritten Numeral Recognition Problem

Although the experiments on the UCI machine learning problems suggest that a classifier-free ensemble selection scheme might be applicable, these experiments were carried out on small databases (apart from the letter recognition problem, where the number of samples ≤ 3000) with a small number of features (apart from the breast cancer problem, where the number of features ≤ 20) and relatively small pools (total classifiers $= 10$). In other words, we knew that clustering diversity might work in classifier-free ensemble selection, but only for small-scale problems.

We wanted to know whether or not classifier-free ensemble selection would be applicable in a large-scale problem. Similar to the experiments on problems extracted from the UCI, these experiments were executed with both the single GA search and the MOGA search.

The experiments were performed on a 10-class handwritten-numeral problem. The data were extracted from $NISTSD19$, essentially as in (99). We first defined 100 feature subspaces for classifier-free ensemble selection (or feature subset selection), each feature subspace containing 32 features extracted from the total of 132 features. For classifier-based ensemble selection, these 100 feature subspaces were used to train 100 corresponding KNN classifiers. We used nearest neighbor classifiers ($K = 1$) for the KNN classifiers.

Several databases were used:

- Training set:

  Containing 5000 data points ($NISTSD19\ hsf\_\{0-3\}$), this set was used to create 100 KNN in Random Subspaces for classifier-based ensemble selection. Note that, since classifier-free ensemble selection does not require classifiers, this set was not used for classifier-free ensemble selection until the final evaluation stage. Note that this set is used only for the KNN classifiers and not for search purposes.

- Optimization set:

  Containing 10000 data points ($NISTSD19\ hsf\_\{0-3\}$), this set was used for the GA and the MOGA search for both classifier-free ensemble selection and classifier-based ensemble selection. In the case of classifier-free ensemble selection, we measured the clustering diversities of various combinations of feature subsets, and, in the case of classifier-based ensemble selection, we measured the ME and MVE of various ensembles of classifiers.

  For both the GA and MOGA search algorithms, we set at 128 the number of individuals in the population and 500 generations, which means that $64,000$ ensembles were evaluated in each experiment. The mutation rate was set to $\frac{1}{L}$, where L is the

length of the mutated binary string (21), and the crossover probability was set to 50%. During the whole search, a threshold of 3 feature subsets or classifiers was applied as the minimum number of feature subsets or classifiers for both classifier-free ensemble selection and classifier-based ensemble selection. All the experiments were carried out with 8 different objective functions (6 clustering diversity measures for classifier-free ensemble selection, ME and MVE for classifier-based ensemble selection) and 30 replications.

- Validation set:

  Containing 10000 data points ($NISTSD19\ hsf\_\{0-3\}$), this set was used to evaluate all the individuals according to the defined objective function, and then to store those individuals in a separate archive after each generation (86) (see Fig. 38) for both classifier-free ensemble selection and classifier-based ensemble selection. Note that the archive mechanism is designed to avoid the overfitting of the defined objective functions, and has been shown to be capable of doing so (86), and that these objective functions may or may not represent classification accuracy. Moreover, at this stage, there are no classifiers for classifier-free ensemble selection.

  For classifier-free ensemble selection, the objective functions are clustering diversities, and thus we evaluated them on the validation set and stored the individuals of its pareto front in a separate archive. For classifier-based ensemble selection, the objective functions are ME and MVE, and thus we evaluated ensemble performances using ME or MVE as fusion functions on the validation set and stored their pareto front in an archive.

  The validation set was also used for the final evaluation of the classifier-free MOGA search. Since the classifier-free MOGA search gives a group of solutions, and because each solution is an ensemble of feature subsets, it is difficult to say which solution will be the best in terms of recognition rate. As a result, these solutions need to be further evaluated. To evaluate these solutions of combinations of feature

subsets, we would need to construct EoCs based on the groups of feature subspaces found, and then evaluate the performances of these ensembles (Fig. 42 & Fig. 43). The solutions stored in the archive were used to construct ensembles using the training set, and their performances evaluated on the validation set. The best solution found on the validation set was then evaluated on the test set.

- Test set:
  Containing 60089 data points ($NISTSD19\ hsf\_\{7\}$), this set was used to evaluate the ensembles selected by classifier-free ensemble selection and by classifier-based ensemble selection. A MAJ is used as the fusion function for classifier combination, because of its stable performance as reported in literature (89).

Note that, according to the definition of the validation set, we used the global validation of all solutions for each generation and the best solutions were maintained in an external archive. The best solution defined in terms of ME in the Pareto front was selected, and its performance evaluated on the test set.

6.4.1 Single Genetic Algorithm for Ensemble Selection for Handwritten Numeral Recognition

We performed a number of experiments directly, using the various objective functions for ensemble selection that had been evaluated by the GA search. We tested 6 clustering diversity measures for classifier-free ensemble selection (or feature subset selection), and ME and MVE for classifier-based ensemble selection. We then compared the performances of the EoCs selected by the two selection methods.

For classifier-based ensemble selection, the EoCs selected by MVE achieved an average 96.45% classification accuracy, while those selected by ME had only a 94.18% recognition rate (Table XL; Fig. 39). Note that the EoCs found by MVE have, in general,$19 \sim 35$ classifiers. However, for classifier-free ensemble selection, the GA search led to the minimum

Table XL

The average recognition rates on test data of ensembles searched by GA with different objective functions including: original clustering diversity measures, compared with mean classifier errors and majority voting errors. The simple majority voting was used as the fusion functions, and the ensemble sizes were indicated in parenthesis

ALL
96.28 % (100.00)

Classifier-Based Ensemble Selection

ME	MVE
94.18 $\pm$ 0.00% (3.00 $\pm$ 0.00)	**96.45** $\pm$ 0.05% (24.53 $\pm$ 3.58)

Classifier-Free Ensemble Selection

Wallace Index-1	Wallace Index-2	Fowlkes-Mallows
92.55 $\pm$ 0.55% ((3.00 $\pm$ 0.00)	92.61 $\pm$ 0.43 % (3.00 $\pm$ 0.00)	93.06 $\pm$ 0.14% (3.00 $\pm$ 0.00)
Rand	Jacard	Mirkin's
92.25 $\pm$ 0.56% (3.00 $\pm$ 0.00)	92.22 $\pm$ 0.10% (3.00 $\pm$ 0.00)	93.03 $\pm$ 0.50% (3.00 $\pm$ 0.00)

number of feature subsets (Fig. 40). Nevertheless, there is a huge gap between the performances of classifier-free ensemble selection using clustering diversity indices and those of classifier-based ensemble selection using MVE. We note that even classifier-based ensemble selection using simple ME can perform better than classifier-free ensemble selection using clustering diversity measures as objective functions.

However, this does not mean that the idea of classifier-free ensemble selection is not a valid one. As we have already stated, the major problem of the GA search is its convergence to the minimum feature subset size (3 feature subsets), and thus the problem resides more in the search algorithm than in the choice of objective functions. That is why we applied MOGA for classifier-free ensemble selection.

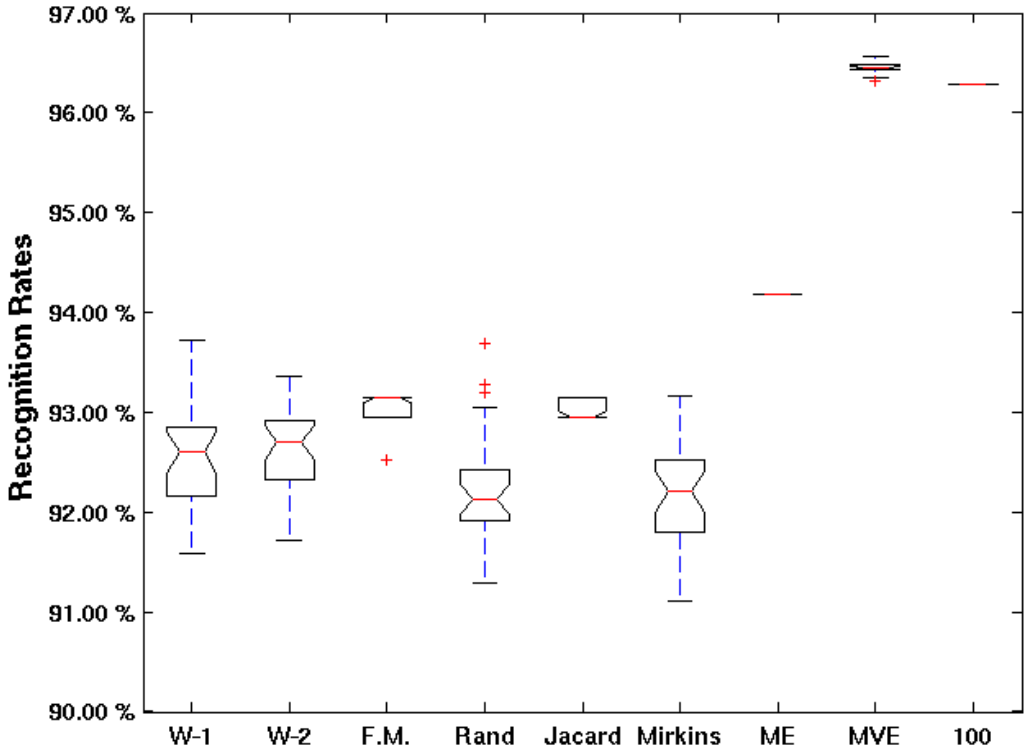

Figure 39 The average recognition rates achieved by EoCs selected by modified clustering diversities with the single GA, compared with Mean Classifier Error (ME), Majority Voting Error (MVE), and the ensemble of all (100) knn classifiers

6.4.2 Multi-Objective Genetic Algorithms for Ensemble Selection for Handwritten Numeral Recognition

For classifier-free ensemble selection, the use of the MOGA search emphasizes the optimization of the clustering indices, as well as the maximization of the number of feature subsets. While the latter is no less relevant to better ensemble performance, it does avoid the problem of minimum ensemble size convergence that occurred in the GA search. While a MOGA search might not be necessary for classifier-based ensemble selection, we

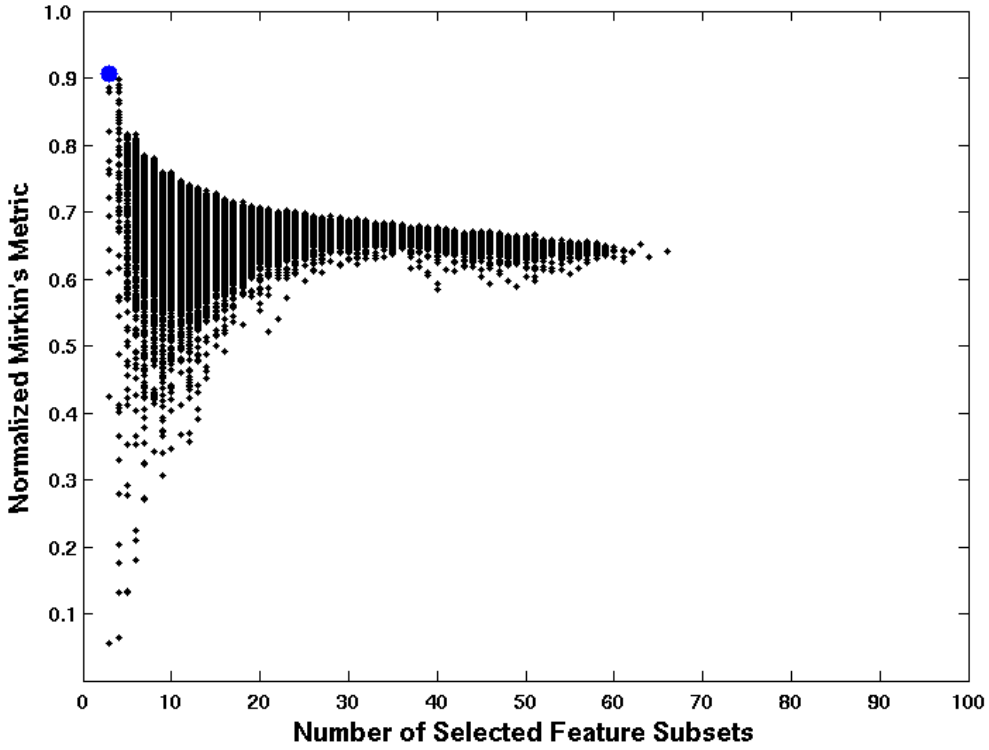

Figure 40 The evaluated population (diamonds) and and selected solution (the circle) based on the single GA search with Mirkin's Metric as the objective function. The number of selected feature subsets is shown to illustrate the process of the convergence into the minimum feature subset size

performed one nonetheless, so that we could compare the results of classifier-based ensemble selection with those of classifier-free ensemble selection.

First, we note that, because we used a MOGA, classifier-free ensemble selection with clustering diversity indices no longer converged to 3 feature subsets (Fig. 42). In general, the population selected from the pareto front has about half the feature subsets of the total pool (see Table XLI). This could allow further, more refined ensemble selection.

Moreover, we note that, in general, the feature subsets selected by classifier-free ensemble selection with clustering diversity indices construct adequate ensembles. The recognition

Table XLI

The average recognition rates on test data of ensembles searched by MOGA with different objective functions including: original clustering diversity measures, three approximations of classifier diversity measures, compared with mean classifier errors and majority voting errors. The simple majority voting was used as the fusion functions, and the ensemble sizes were indicated in parenthesis

ALL
96.28 % (100.00)

Classifier-Based Ensemble Selection

ME	MVE
96.26 $\pm$ 0.08% (48.83 $\pm$ 5.75)	96.25 $\pm$ 0.04% (49.25 $\pm$ 5.59)

Classifier-Free Ensemble Selection

Wallace Index-1	Wallace Index-2	Fowlkes-Mallows
96.24 $\pm$ 0.08% (50.88 $\pm$ 5.34)	96.25 $\pm$ 0.06 % (51.08 $\pm$ 4.46)	96.25 $\pm$ 0.08% (50.42 $\pm$ 4.93)
Rand	Jacard	Mirkin's
96.23 $\pm$ 0.08% (51.95 $\pm$ 4.09)	96.26 $\pm$ 0.06% (52.91 $\pm$ 4.63)	96.19 $\pm$ 0.08% (50.75 $\pm$ 4.61)

Table XLII

The p-value of hypothesis test on the recognition rates of ensembles selected by various objective functions compared with that of the ensemble of all classifiers

Mirkin's	Wallace Index-1	Wallace Index-2	Fowlkes-Mallows	Rand	Jacard	M.V.E	M.E.
0.0001	0.2005	0.2005	0.0428	0.2005	0.5847	0.8555	0.0161

rates achieved by these ensembles are very close to those achieved when all the classifiers are used (Fig. 41). In fact, the significances are usually $p \geq 0.01$ (Table XLII).

For classifier-based ensemble selection, ME also benefits from the MOGA scheme, and even slightly outperforms MVE as an objective function in a MOGA (See Table XLI). By contrast, MVE did not perform quite as well as in a single GA, but the difference is rather small (0.20%). With a MOGA, MVE selected 49.25 classifiers on average, many more than it did with the simple GA.

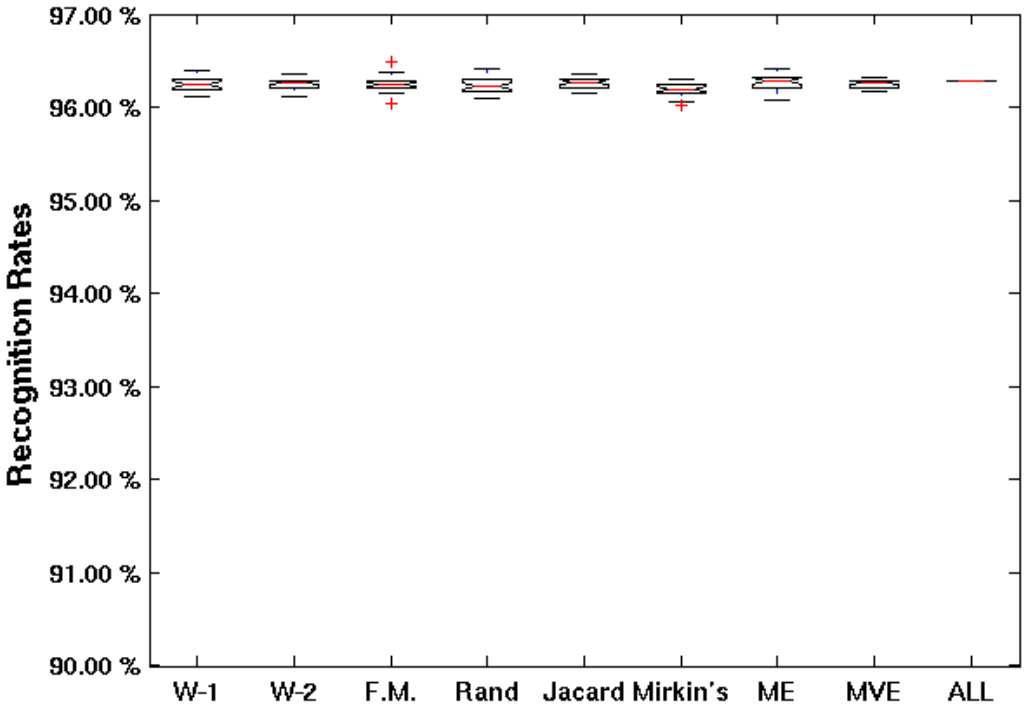

Figure 41 Box plot of the classifier-free ensemble selection schemes using MOGA compared with the classifier-based ensemble selection using Mean Error (ME) and Majority Voting Error (MVE) as objective functions

The results of using the clustering diversities in classifier-free ensemble selection are encouraging, and all of them performed as well as the ensemble of all classifiers, but the ensemble sizes were cut in half. Furthermore, there is no clear difference among the various clustering diversity measures (Fig. 41). This indicates that data diversity can be used to carry out ensemble selection in Random Subspaces, and that the proposed classifier-free ensemble selection scheme using clustering diversity measures as objective functions does work.

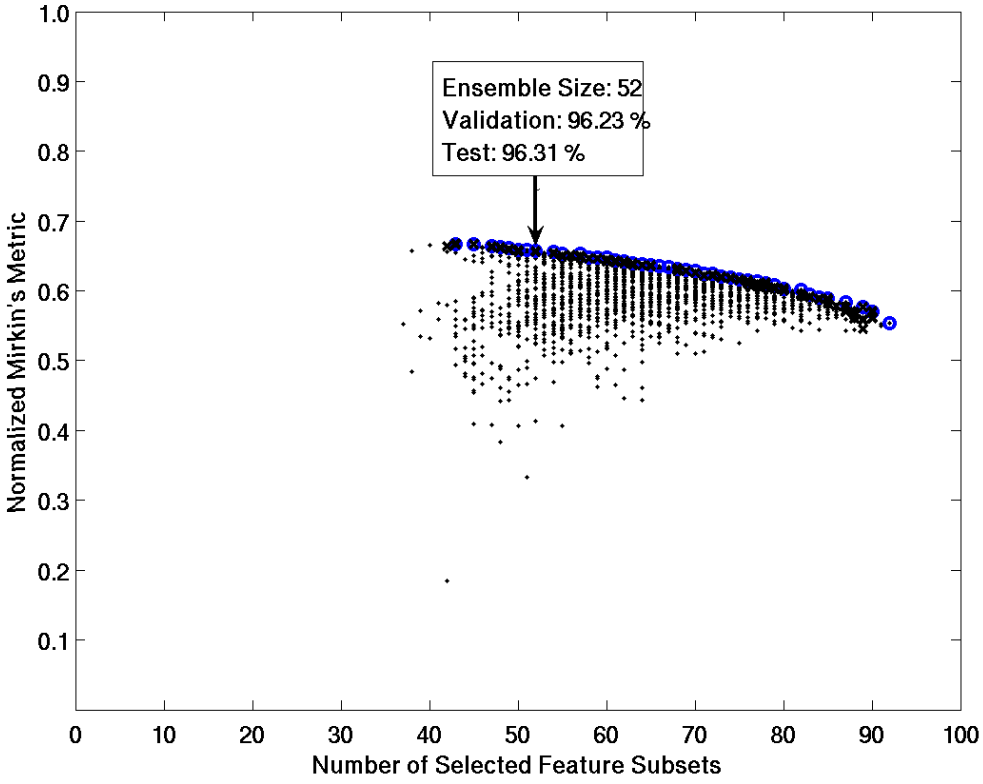

Figure 42 The pareto front of the MOGA search for the classifier-free ensemble selection scheme. The evaluated population (diamonds), the population in the pareto front (circles) and the validated solution (crosses) based on the MOGA search with Mirkin's Metric and the number of selected feature subsets the objective functions. The best performance evaluated on the validation set is shown in the text boxes

6.4.3 Classifier-Free Ensemble Selection Combined with Pairwise Fusion Functions for Handwritten Numeral Recognition

While MAJ is one of the fusion functions most often used for combining classifiers, it is not necessarily the optimum choice. In our experiment on handwritten numeral recognition, in which all the ensembles were combined with MAJ, classifier-based ensemble selection using MVE as the objective function, which uses MAJ to evaluate the ensem-

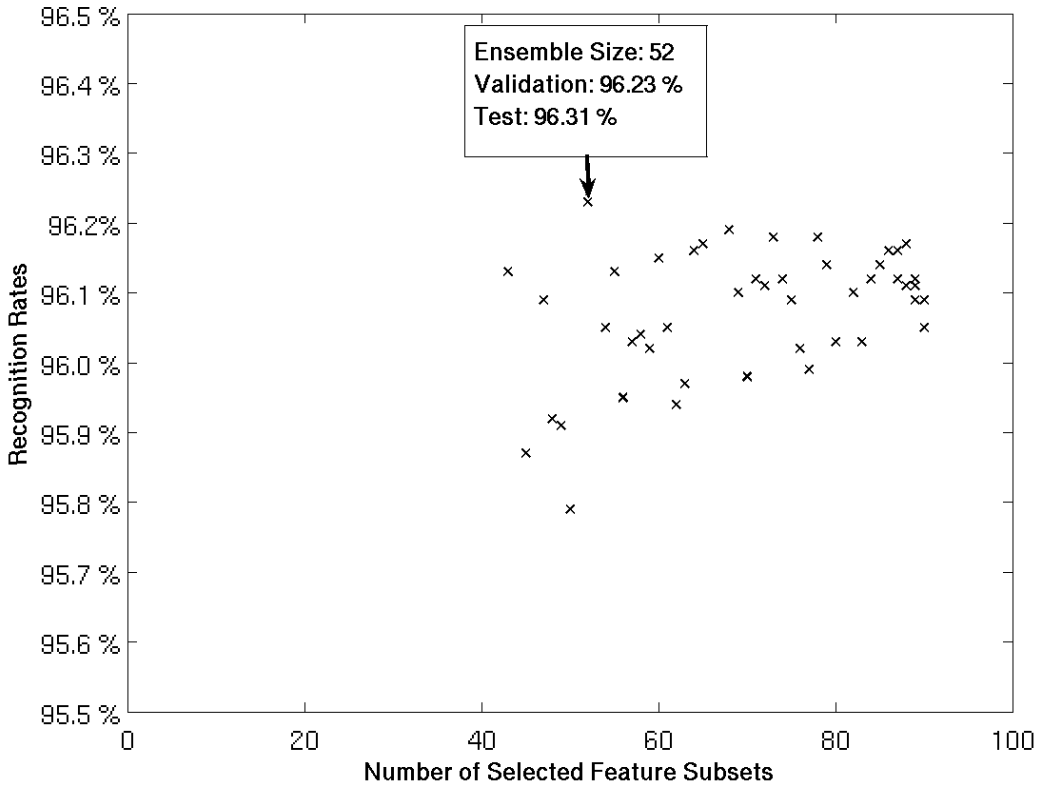

Figure 43 The validated recognition rates of individuals on pareto front. E.S. = Ensemble Size; V.R.R. = Validation Recognition Rate in percents

bles, performed better than classifier-free ensemble selection using clustering diversity as the objective function.

However, if we apply other fusion functions - such as the pairwise fusion matrix with the majority voting rule (PFM-MAJ) (59; 60) - the classifier-based ensemble selection using MVE might not be the best scheme. It turns out that the performances of ensembles selected by classifier-free ensemble selection can be further improved by using better fusion functions. As we can see in Table XLIII, the recognition rates of ensembles applying PFM-MAJ are apparently better than those applying the simple MAJ.

Moreover, for the MOGA search, when PFM-MAJ was used as the fusion function, classifier-free ensemble selection using clustering diversity indices outperformed the classifier-based ensemble selection using MVE.

Table XLIII

The average recognition rates on test data of ensembles searched by MOGA with different objective functions. The pairwise confusion matrix applying the pairwise-majority voting was used as the fusion functions. The ensemble sizes are the same as those in Table. XLI

ALL
96.28 % (100.00)

Classifier-Based Ensemble Selection

ME	MVE
96.89 $\pm$ 0.05% (48.83 $\pm$ 5.75)	96.78 $\pm$ 0.09 (49.25 $\pm$ 5.59)

Classifier-Free Ensemble Selection

Wallace Index-1	Wallace Index-2	Fowlkes-Mallows
96.91 $\pm$ 0.05% (50.88 $\pm$ 5.34)	96.90 $\pm$ 0.04 % (51.08 $\pm$ 4.46)	96.90 $\pm$ 0.04% (50.42 $\pm$ 4.93)
Rand	Jacard	Mirkin's
96.90 $\pm$ 0.04% (51.95 $\pm$ 4.09)	96.89 $\pm$ 0.03% (52.91 $\pm$ 4.63)	96.88 $\pm$ 0.08% (50.75 $\pm$ 4.61)

6.5 Discussion

In this chapter, we examined whether or not clustering diversity can represent the data diversity of different feature subsets in Random Subspaces, and whether or not the use of clustering diversity as the data diversity measure could allow us to apply a classifier-free ensemble selection scheme.

First, for classifier-free ensemble selection, we used the single GA as the search algorithm. We found that, with the clustering diversity indices as objective functions, it tends to converge to the minimum number of feature subsets, which makes a classifier-free ensemble selection scheme less useful.

Then, in order to compensate for the problem of the minimum feature subset convergence of the clustering diversities, we used the MOGA as the search algorithm. The clustering diversity measures yielded encouraging performances as objective functions for the classifier-free ensemble selection scheme.

However, we note that the proposed scheme for classifier-free ensemble selection bears the additional cost of the clustering and on MOGA search. But, in general, the cost of the clusterings is much less than the cost of training classifiers such as the Support Vector Machine or the Multi-Layer Neural Network. Moreover, with the help of eq. 6.5 $\sim$ eq. 6.8, comparison of the clusterings takes a relatively short time. For the MOGA search, the additional objective - the number of feature subsets - does not require complicated calculation.

The only major cost is the evaluation of the solutions found on the pareto front after the MOGA search. This requires the training of a classifier for each feature subset selected to evaluate the performances of ensembles, so that the best ensemble can be chosen. Compared with a traditional ensemble selection scheme, which requires the training of all classifiers and combinations of all the ensembles evaluated, the proposed scheme offers an interesting alternative. This approach will be especially attractive for tackling problems with a large classifier pool and time-consuming classifier training.

6.6 Conclusion

In this chapter, we argue that clustering diversities actually represent the data diversities of different feature subsets in the Random Subspaces ensemble creation method. These data diversities can be measured with the help of clustering diversities without any classifier training. As a result, the feature subsets can be selected by clustering diversities to construct the classifiers in Random Subspaces.

Applying the MOGA search, we show that the ensembles selected by the clustering diversities had performances comparable to those selected by MVE, which is regarded as one of the best objective functions for ensemble selection (89). The results are encouraging. Based on our exploratory work, we have drawn up some implications for the classifier-free ensemble selection approach:

a. In Random Subspaces, with the MOGA search the clustering diversity measures are good objective functions for ensemble selection.

b. In Random Subspaces, the ensembles selected by the different clustering diversity measures have so far been found to have similar performances based on the MOGA search.

Even though the clustering diversities might only be able to represent data diversities in Random Subspaces, for Bagging, which only use a part of the samples, there is still no adequate measure for their data diversities. It will be of great interest to figure out how to measure the data diversities in Bagging. Finally, we have to mention that, due to its special ensemble generating mechanism, the scheme is not likely to be applicable in Boosting.

CHAPTER 7

CONCLUSION

7.1 Contributions

In this document, we present our five major contributions to the improvement of EoCs: a new ensemble creation method for ensembles of HMM (EoHMM) classifiers based on different codebook sizes, a new ensemble selection method based on the combination of the diversity and classifier accuracy, a dynamic ensemble selection method based on the concept of the oracle, a classifier-free ensemble selection based on clustering diversity and a pairwise fusion matrix for classifier combination.

To demonstrate the usefulness of these methods, we carried out various experiments on problems extracted from the UCI Machine Learning Repository, as well as handwritten numeral digits extracted from NIST SD19. In addition, we have focused on improving EoHMM classifiers. We generated the basic HMM classifiers using different codebook sizes (and thus different codebooks). Once these HMM classifiers had been generated, we performed ensemble selection using a compound diversity function which combines the diversity between classifiers and classifier accuracies. Following ensemble selection, we used the pairwise fusion matrix for classifier combination. We demonstrated that the new ensemble creation method (using different codebook sizes), the new ensemble selection method (using compound diversity functions) and the new classifier combination method (using the pairwise fusion matrix) all contribute to the improvement of EoHMM classifiers.

Dynamic ensemble selection is regarded solely as an alternative in our work. Unlike static ensemble selection (selection of an ensemble for all samples) and dynamic classifier selection (selection of a classifier for each sample), it selects one ensemble for each test sample. The method presented uses the concept of the oracle. We showed that this method worked

on the problems extracted from the UCI Machine Learning Repository, as well as on the handwritten numeral digits extracted from NIST SD19 using KNN classifiers. This is the first dynamic ensemble selection method to be presented in the literature.

Another alternative that we offered is so-called "classifier-free ensemble selection". We tried to measure the data diversity of different feature subspaces using clustering diversity measures. Because the data diversity of different feature subspaces can be measured, we can select those feature subspaces that have the maximum diversities. The feature subspaces with high diversity will generate classifiers which also have high diversity. This method is the first ensemble selection method presented in the literature based on the concept of data diversity. However, we need to remember that this method applies only on classifiers generated with the Random Subspaces ensemble creation method, and cannot be applicable on other ensemble creation methods, including Bagging and Boosting.

7.2 Future Works

A number of avenues for future work are possible. The first derives from the fact that EoHMM classifiers have thus far only been created based on different codebook sizes. Since we did not optimize the number of the states for each HMM, we could use different states and different codebooks to create EoHMM classifiers. We shall expect a higher diversity among classifiers and probably a better recognition results on EoHMMs.

The second derives from the fact that our pairwise fusion matrix transformation for classifier combination is based merely on classifier pairs. We instinctively feel, however, that a similar method based on three classifiers could work, and that we could construct fusion matrices based on the output of any three, and then on four, five, six, or more classifiers. It would therefore be advisable to test different degrees of transformation for classifier combination in the future.

The third derives from the fact that the new dynamic ensemble selection method is based on the concept of the oracle. But, in order to find the most adequate ensemble for a test sample, we measured the Euclidean distance between this test sample and other training samples. We did not weight the Euclidean distance measured. If we do so, we might find a more adequate oracle for the test sample.

The fourth derives from the fact that the new classifier-free ensemble selection method only works for the Random Subspaces ensemble creation method and not to other methods, such as Bagging and Boosting. But, would it be possible to measure data diversity for other ensemble creation methods? If so, then classifier-free ensemble selection will be also possible for Bagging and Boosting. It would therefore be of great interest to investigate this question further.

To conclude, our work offers a number of contributions on different aspects of a multiple classifier system. We managed to improve the pattern recognition results by using ensembles of multiple classifiers, and we refined the techniques of ensemble creation, ensemble selection, and classifier combination. This is not to say that we have achieved our goal, however. Just as research is a never-ending process, we look forward to a journey of discovery in seeking improvements to the state of this art in the future.

APPENDIX 1

The Random Subspaces ensemble creation method

Random Subspaces is an ensemble creation method (49) that uses different feature subspaces to create an ensemble of classifiers. Under Random Subspaces, we train each classifier using all samples in certain feature subsets. Since different classifiers are trained with different feature subsets, these classifiers might give different outputs in classification. In general, we fix the size of feature subsets that classifiers are trained with, and the size of feature subsets is known as the cardinality of Random Subspaces.

To illustrate, we give an example below. Suppose that we have some data points for classifier training, and each data point has M features. Now we can decide only use $\hat{M}$ features for classifier training, so $\hat{M}$ is the cardinality of Random Subspaces. To select $\hat{M}$ features from the total M features, we have $C_M^{\hat{M}}$ choices, and that is the maximum number of classifiers that we can generate.

For example, all data points have 6 features. If we decide to use only 3 of these features to train each classifier, then the cardinality is 3. Since only 3 of 6 features are used for classifier training, we have $C_6^3 = \frac{6\times5\times4}{3\times2\times1} = 20$ possibilities of composition of classifiers. As a result, the maximum number of classifiers with this cardinality is 20.

As we can observe that the sufficient number of available features is one of the crucial keys for Random Subspaces ensemble creation method. Ho described that Random Subspaces method is best when the dataset has a large number of features and samples, and is not good when the dataset has very few features coupled with a very small number of samples (49).

However, Ho also observed that Random Subspaces method is good when there is certain redundancy in the dataset, especially in the collection of features. Consequently, Random Subspaces method is especially valuable for tasks involving low-level features (49). Note that in order to have enough classifiers, in some cases it might be desirable to generate additional features by using original features. By this way, even though the generated

features are correlated with original features, a enlarged feature space will allow more classifiers to be created with Random Subspaces.

Although is has been observed that the ensemble accuracy improves when the number of classifiers increases, Ho suggested that using half of feature components yielded the best ensemble accuracy (49). Nevertheless, when the number of features is small, there is a trade-off between the cardinality of Random Subspaces and the accuracy of single classifiers. It is thus important to assure that the cardinality used will guarantee a minimum accuracy of single classifiers.

APPENDIX 2

The Effects of the Class Size and of the Ensemble Size on the Correlation between the Classifier Diversity and the Ensemble Accuracy

Even though a number of studies have targeted on the correlation measurements between the classifier diversity and the ensemble accuracy, the influences of the class dimension and the ensemble size get relatively little attention. In this appendix, we try to figure out their impacts on the correlation measurements.

For a sample x in a T-class problem, suppose that the correct class is $i, 1 \leq i \leq T$. The ensemble will give correct output only under the condition $\forall j,\ c(i)_T > c(j)_T$, for $1 \leq i, j \leq T, i \neq j$, where $c(i)_T$ is the number of classifiers making a decision on class i, and $c(j)_T$ is the number of classifiers making a wrong decision on another class j, in a T-class problem. Under the condition $\forall j, c(i)_T > c(j)_T$, the $c(i)_T$ can decrease, and the $c(j)_T$ can increase, and the ensemble can still give the correct output.

Suppose that, for a certain problem, for a sample x, the correct class label $t(x)$ is $i, 1 \leq i \leq T$, then the probability of the sample x being classified as class j is $P(c(j)_T|t(x) = i)$, we have

$$\sum_{j=1}^{T} P(c(j)_T|t(x) = i) = 1,\ \ 1 \leq i, j \leq T \tag{2.1}$$

If the number of classes increases to $T + 1$ classes, the equation above will become :

$$\sum_{j=1}^{T+1} P(c(j)_{T+1}|t(x) = i) = 1,\ \ 1 \leq i, j \leq T + 1 \tag{2.2}$$

Compared with the eq. 2.1, the probability $P(c(T + 1)_{T+1}|t(x) = i)$ is added to the eq. 2.2. This term can be regarded as the sum of the probabilities of classifying the sample x as class j in the case of T classes but as class $T + 1$ in the case of $T + 1$ classes. This term can be further decomposed as :

$$P(c(T+1)_{T+1}|t(x) = i) = \sum_{j=1}^{T} P(c(T+1)_{T+1}, c(j)_T|t(x) = i),\ \ 1 \leq i, j \leq T \tag{2.3}$$

where $P(c(T+1)_{T+1}, c(j)_T|t(x)=i)$ is the probability of classifying the sample as class j in the problem T classes but as $T+1$ in the problem of $T+1$ class, note that $1 \leq j \leq T$. If we suppose that samples classified as class j in the problem T classes will only be classified as the original class j or as the new class $T+1$ in the problem of $T+1$ class, then we can write :

$$P(c(j)_T|t(x)=i) = \frac{P(c(T+1)_{T+1}, c(j)_T|t(x)=i)}{P(c(T+1)_{T+1}, c(j)_T|t(x)=i) + P(c(j)_{T+1}, c(j)_T|t(x)=i)}, \quad 1 \leq i, j \leq T \tag{2.4}$$

For the problem with T classes, given L classifiers, then we can define the margin $m(T)$ as :

$$m(T) = L \cdot (P(c(i)_T|t(x)=i) - P(c(j)_T|t(x)=i)) \tag{2.5}$$

For the same problem, if we add an independent class, i.e., if the total number of classes increases to $T+1$, the margin $m(T+1)$ would be :

$$m(T+1) = L \cdot (P(c(i)_{T+1}|t(x)=i) - P(c(j)_{T+1}|t(x)=i)) \tag{2.6}$$

Inasmuch as the added class $T+1$ is independent of all other T classes, we suppose that the class $T+1$ does not change the proportional posterior probabilities of outputs among T classes. This means that in a one-against-one manner of classification, class $T+1$ does not interfere in the classification between class f_i and class f_j, with $1 \leq i, j \leq T,\ i \neq j$. In other words, samples classified as class i in the problem T classes will only be classified as the original class i or as the new class $T+1$ in the problem of $T+1$ class, but never as

another class j. Based on this assumption, we have :

$$P(c(i)_{T+1}|t(x)=i) = P(c(i)_{T+1}, c(i)_T|t(x)=i) \quad (2.7)$$

$$P(c(j)_{T+1}|t(x)=i) = P(c(j)_{T+1}, c(j)_T|t(x)=i) \quad (2.8)$$

$$\begin{aligned} m(T) = L \cdot (&P(c(i)_{T+1}, c(i)_T|t(x)=i) + P(c(T+1)_{T+1}, c(i)_T|t(x)=i) \\ &-P(c(j)_{T+1}, c(j)_T|t(x)=i) - P(c(T+1)_{T+1}, c(j)_T|t(x)=i)) \end{aligned} \quad (2.9)$$

$$m(T+1) = L \cdot (P(c(i)_{T+1}, c(i)_T|t(x)=i) - P(c(j)_{T+1}, c(j)_T|t(x)=i)) \quad (2.10)$$

Using eq. 2.9 and eq. 2.10, we obtain the difference in the margins $m(T)$ and $m(T+1)$:

$$\begin{aligned} m^*(T) = m(T) - m(T+1) = \\ L \cdot (P(c(T+1)_{T+1}, c(i)_T|t(x)=i) - P(c(T+1)_{T+1}, c(j)_T|t(x)=i)) = \\ L \cdot P^*(T) \end{aligned} \quad (2.11)$$

Suppose that the newly added class $T+1$ is independent of all other T classes, since $P(c(i)_T|t(x)=i) \geq P(c(j)_T|t(x)=i)$, we will have $P(c(T+1)_{T+1}, c(i)_T|t(x)=i) \geq P(c(T+1)_{T+1}, c(j)_T|t(x)=i)$. This will lead to $m^*(T) \geq 0$, i.e., $m(T+1) \leq m(T)$.

That means, when the number of classes T increases, we will probably get a smaller $m(T)$. Moreover, the margin $m(T)$ is also proportional to the number of classifiers L. Good estimation of ensemble accuracy will require high class problems and a small number of classifiers in ensembles.

APPENDIX 3

Classifier Diversity Measures

In chapter 2 and in chapter 6, we used some classifier diversity measures in our experiments. We thus feel the need to give the details of their definitions in this appendix.

The traditional concept of diversity is composed of the terms of correct / incorrect classifier outputs. By comparing these correct / incorrect outputs among classifiers, their respective diversity can be calculated. In this section, we provide an overview of traditional diversity measures dealt with in this thesis:

a. Pairwise diversity measures

Diversity is measured between two classifiers. In the case of multiple classifiers, diversity is measured on all possible classifier-pairs, and global diversity is calculated as the average of the diversities on all classifier-pairs. That is, given L classifiers, $\frac{L\times(L-1)}{2}$ pairwise diversities $d_{12}, d_{13}, ..., d_{(L-1)L}$ will be calculated, and the final diversity $\bar{d}$ will be its average (66):

$$\bar{d} = 2 \times \frac{\sum_{ij} d_{ij}}{L \times (L-1)}, i \leq j \tag{3.1}$$

This type of diversity includes: Q-statistics (1; 5), the correlation coefficient (66), the disagreement measure (49) and the double fault (29).

b. Non-Pairwise diversity measures

There are others diversities that are not pairwise, i.e. they are not calculated by comparing classifier-pairs, but by comparing all classifiers directly. This type of diversity includes: the Entropy measure (66), Kohavi-Wolpert variance (61), the measurement of interrater agreement (5; 25), the measure of difficulty (47), generalized diversity (80) and coincident failure diversity (80).

Most research suggests that neither type of diversity is capable of achieving a high degree of correlation with ensemble accuracy, as only very weak correlation can be observed (66). To understand how they work, and why one might be better than another, we detail the

definitions of the diversity measures evaluated in this section. In general, to calculate the diversity measures among classifiers, either we count the number of correctly / wrongly classified samples for each classifier pair, which gives us pairwise diversity measures, or we count the number of correctly / wrongly classified classifiers for each sample, which produces non-pairwise diversity measures.

For pairwise diversity measures, suppose that we have 2 classifiers D_i and D_k. We should define :

a. N^{11} as the number of samples correctly classified by both D_i and D_k

b. N^{10} as the number of samples correctly classified by D_i but not by D_k

c. N^{01} as the number of samples correctly classified by D_k but not by D_i

d. N^{00} as the number of samples incorrectly classified by both D_i and D_k

Now, the total number of samples N should be :

$$N = N^{11} + N^{10} + N^{01} + N^{00} \tag{3.2}$$

For non-pairwise diversity measures, suppose that there are L classifiers; for each sample x_j, we define the number of classifiers that correctly classify x_j as $l(x_j)$, and the probability of a randomly drawn sample x_j having $l(x_j) = L - i, 0 \leq i \leq L$ as p_i. Using these elements, we can define the following diversity measures :

a. Disagreement Measure (DM) (49)

This is a ratio between the number of observations on which one classifier is correct and the other is incorrect to the total number of observations.

$$dm_{i,k} = \frac{N^{01} + N^{10}}{N^{11} + N^{10} + N^{01} + N^{00}} \tag{3.3}$$

The DM index is especially interesting for us, for this index has a strong relationship with clustering validity index. See appendix 4 for details.

b. Double-Fault (DF) (29)

This is the proportion of the samples that have been misclassified by both classifiers :

$$df_{ik} = \frac{N^{00}}{N^{11} + N^{10} + N^{01} + N^{00}} \tag{3.4}$$

c. Kohavi-Wolpert Variance (KW) (61)

$$kw = \frac{1}{NL^2} \sum_{j=1}^{N} l(x_j)(L - l(x_j)) \tag{3.5}$$

d. Interrater Agreement (INT) (25)

Define $\bar{p}$ as the average individual classification performance :

$$\bar{p} = \frac{1}{LN} \sum_{j=1}^{N} l(x_j) \tag{3.6}$$

Then, the interrater agreement is defined as :

$$int = 1 - \frac{\frac{1}{L} \sum_{j=1}^{N} l(x_j)(L - l(x_j))}{N(L-1)\bar{p}(1-\bar{p})} \tag{3.7}$$

For pairwise use, interrater agreement can also be defined as :

$$\frac{2(N^{11}N^{00} - N^{01}N^{10})}{(N^{11} + N^{10})(N^{01} + N^{00}) + (N^{11} + N^{01})(N^{10} + N^{00})} \tag{3.8}$$

e. Entropy Measure (EN) (66)

The entropy measure is defined as :

$$en = \frac{1}{N}\sum_{j=1}^{N}\frac{1}{L-[L/2]}\min\{l(x_j), L-l(x_j)\} \tag{3.9}$$

f. Measure of Difficulty (DIFF) (47)

We define a discrete random variable X_j taking values in $\{\frac{0}{L}, \frac{1}{L}, \cdots, 1\}$ and denoting the proportion of classifiers that correctly classify a sample x drawn randomly from all the samples. Then, the measure of difficulty is defined by calculating the variance of X as $Var(X)$.

g. Generalized Diversity (GD) (80)

First we define $p(1)$ and $p(2)$ based on p_i:

$$p(1) = \sum_{i=1}^{L}\frac{i}{L}p_i \tag{3.10}$$

$$p(2) = \sum_{i=1}^{L}\frac{i(i-1)}{L(L-1)}p_i \tag{3.11}$$

Then generalized diversity is defined as :

$$gd = 1 - \frac{p(2)}{p(1)} \tag{3.12}$$

h. Coincident Failure Diversity (CFD) (80)

This is a modification of gd and is defined as :

$$cfd = \frac{1}{1-p_0}\sum_{i=1}^{L}\frac{L-i}{L-1}p_i, p_0 < 1 \tag{3.13}$$

$$cfd = 0, p_0 = 1 \tag{3.14}$$

i. Q-Statistics (Q) (1; 5)

$$Q_{ik} = \frac{N^{11}N^{00} - N^{01}N^{10}}{N^{11}N^{00} + N^{01}N^{10}} \tag{3.15}$$

j. Correlation Coefficient (COR) (66)

This is defined as :

$$cc = \frac{N^{11}N^{00} - N^{01}N^{10}}{((N^{11} + N^{10})(N^{01} + N^{00})(N^{11} + N^{01})(N^{00} + N^{10}))^{\frac{1}{2}}} \tag{3.16}$$

Of the diversity measures defined above, DM, DF, Q and COR are pairwise, and the others are non-pairwise. These diversity measures are designed for ensemble selection, but no significant correlation has been observed between them and ensemble accuracy.

APPENDIX 4

Justification of Disagreement Measure (DM) as a Classifier Diversity Index

Assume that the clustering upon the single feature f_i generates $\hat{K}_i$ clusters. So for cluster i

$$\sigma_i^2 = \varepsilon[(x-\mu_i)^2] = \int_{\infty}^{-\infty}(x-\mu_i)^2 p(x)dx \tag{4.1}$$

$$\mu_i = z_i, p(x) = \frac{1}{|C_i|}, \sigma_i^2 = (\frac{1}{|C_i|}\sum_{x\in C_i}\{\|x-z_i\|^2\}) = \hat{S}_i \tag{4.2}$$

$$d_{ij} = |z_i - z_j| = |\mu_i - \mu_j| \tag{4.3}$$

where σ_i^2 is the standard variance, and μ_i is the mean value, $|C_i|$ is the number of samples, and z_i is the centroid, for the cluster i. d_{ij} indicates the distance between two clusters i and j. Here we note that the difference between S_i, which is used by DB index, and $\hat{S}_i$, which is showed above, is merely a calculation of square. So we rewrite the elements of measure of between-clusters distances and within-cluster scatter in DB index as :

$$S_i = (\frac{1}{|C_i|}\sum_{x\in C_i}\{\|x-z_i\|\}) = \check{\sigma}_i \tag{4.4}$$

$$R_i = \max_{j,j\neq i}\{\frac{\check{\sigma}_i + \check{\sigma}_j}{|\mu_i - \mu_j|}\} \tag{4.5}$$

It can be shown that the DB index based on R_i is a reasonable measure for one single feature. Take into account the discriminant function $g_i(x)$ as the probability of the sample x belonging to class ω_i, we can use the minimum-error-rate criterion and re-write it as :

$$g_i(x) = p(\omega_i|x) = p(x|\omega_i)\cdot p(\omega_i) \tag{4.6}$$

$$g_i(x) = \ln p(x|\omega_i) + \ln p(\omega_i) \tag{4.7}$$

where $p(\omega_i)$ is a priori probability for the likelihood of belonging to class ω_i. When it is a Gaussian distribution for cluster i, then :

$$p_i(x) = \frac{1}{2\pi\sigma_i} \exp[\frac{-(x-\mu_i)^2}{2(\sigma_i)^2}] \tag{4.8}$$

So the discriminant function for a single sample x under Gaussian distribution is :

$$g_i(x) = \frac{-\|x-\mu_i\|^2}{2(\sigma_i)^2} + \ln p(w_i) \tag{4.9}$$

When no knowledge about a priori probability is available, then $\ln p(w_i)$ can be ignored. to simplify our notation we write :

$$g_i(x) = \frac{-\|x-\mu_i\|}{(\sigma_i)} \tag{4.10}$$

$$-g_i(x) \cdot (\sigma_i) = \|x-\mu_i\| \tag{4.11}$$

Where a factor of 2 is eliminated and the square term is replaced by its distance. Since any x not between μ_i and μ_j can lead to the following :

$$|\mu_i - \mu_j| = |\|x-\mu_i\| - \|x-\mu_j\|| \tag{4.12}$$

$$R_i = \max_{j,j\neq i}\{\frac{\check{\sigma_i} + \check{\sigma_j}}{\|g_j(x)\cdot(\sigma_j) - g_i(x)\cdot(\sigma_i)\|}\} \tag{4.13}$$

In case the variance is equal for each cluster, i.e., $\sigma_i = \sigma_j$, and $\check{\sigma}_i = \check{\sigma}_j$, then :

$$\hat{R}_i = \max_{j,j\neq i}\{\frac{1}{\|g_j(x) - g_i(x)\|}\} \tag{4.14}$$

So the DB index indicates the reverse of difference of discriminant functions of two classes, minimization of DB index is equal to maximization of the difference of discriminant functions of different classes. However, when the sample x is just between μ_i and μ_j, the the right term of difference of discriminant functions is :

$$\|g_j(x) \cdot (\sigma_j) - g_i(x) \cdot (\sigma_i)\| = |\mu_i + \mu_j - 2 \cdot x| \tag{4.15}$$

$$0 \leq |\mu_i + \mu_j - 2 \cdot x| \leq |\mu_i - \mu_j| \tag{4.16}$$

Since this term depends on the value of x, it is hard for DB index to take into account this condition, but note that the measure $|\mu_i - \mu_j|$ is just its bound value.

At the end, we would like to mention that, when the clusters have different variance values, i.e., $\sigma_i \neq \sigma_j$, or $\check{\sigma}_i \neq \check{\sigma}_j$, DB index uses this factor as a weight of the probability of a class. Just use σ_i instead of $\check{\sigma}_i$, and σ_j instead of $\check{\sigma}_j$, then we have :

$$R_i = \max_{j,j\neq i}\{\frac{1}{\|g_j(x) \cdot \frac{\sigma_j}{\check{\sigma}_i+\check{\sigma}_j} - g_i(x) \cdot \frac{\sigma_i}{\check{\sigma}_i+\check{\sigma}_j}\|}\} \tag{4.17}$$

APPENDIX 5

From Classifier Diversity to Clustering Diversity: A Case Study of Disagreement Measure

5.1 Introduction

All ensemble creation methods generate diverse classifiers with the diverse data subsets, and we wonder whether it is possible to select the data subsets before we train the classifiers for the EoC. The problem is to define a data diversity so that we can use it to do the data subset selection.

The main difficulty is to conceive a data diversity measure that can predict the classifier diversity based on the different training data. In other words, given any two data subsets d_i, d_j, the data diversity between them $Div_d(d_i, d_j)$ should be strongly correlated with the classifier diversity $Div_c(c_i, c_j)$, where c_i and c_j are classifiers trained with the data subsets d_i and d_j, respectively. If the data diversity measure Div_d can help us find a number of suitable data subsets without classifier training, then it can reduce the time for the classifier training. If Div_d can help us find adequate data subsets for the ensemble construction directly, then it can further reduce the time for the ensemble selection.

Since data points might have very different distributions in different feature subspaces, it might be possible to measure the data distributions in different feature subspaces as a measure of data diversity for the Random subspace. Given different feature subsets, if we use the same clustering algorithm with the fixed parameters to carry out clustering on them, it is possible that the clustering diversity between the different feature subsets indicates the data diversity between them.

To verify this hypothesis, we discussed the relationship of classifier diversity and clustering diversity in different feature subspaces, and showed that there is a strong connection between diversity measure (DM), a classifier diversity measure, and Mirkin's metric, a clustering diversity measure. We went further and show how to have better approximation of DM from the Mirkin's Metric. Three approximations of DM based on Mirkin's Metric were shown in appendix 6.

In the next section, we discuss the connection between classifier diversity measures and clustering diversity measures, we propose three approximations of classifier diversity from clustering diversity based on various hypothesis. The correlation measurement between classifier diversity measures and clustering diversity measures is then carried out. Discussion and conclusion are in the last sections.

5.2 The Relationship between the Disagreement Measure (DM) and Mirkin's Metric

Based on the definitions of the classifier diversity and the clustering diversity mentioned in the above sections, we need to figure out their connections and whether it is possible to approximate classifier diversity from a clustering diversity under some circumstances. But to start, some basic assumptions must be done.

5.2.1 Concept on 2-clusters clustering

For the development in this section, we make the following assumptions:

a. The data set is a 2-class problem.

b. The data set can be perfectly partitioned into 2 clusters.

c. For each cluster, all the samples in one cluster belong to the same class.

d. Both classes have the similar number of samples.

To get into this discussion, suppose that we have binary classes x, y, and two classifiers D_i, D_k, then we can establish the table below (Table XLIV): where N_{xx} is the number of samples classified as x by both D_i and D_k, N_{xy} is the number of samples classified as x by D_i but as y by D_k, N_{yx} is the number of samples classified as y by D_i but as x by D_k,

Table XLIV

Key concept for relating clustering diversity to classifier diversity

	D_k classify (x)	D_k classify (y)
D_i classify (x)	N_{xx}	N_{xy}
D_i classify (y)	N_{yx}	N_{yy}

N_{yy} is the number of samples classified as y by both D_i and D_k. Intuitively, these three equations stand :

$$N_{xx} + N_{xy} + N_{yx} + N_{yy} = N \tag{5.1}$$

$$N_{xx} + N_{yy} = N_{11} + N_{00} \tag{5.2}$$

$$N_{xy} + N_{yx} = N_{10} + N_{01} \tag{5.3}$$

This table and these equations allow us to have an insight on the relation between the clustering diversity and classifier diversity. Suppose that, for each classifier, all the samples classified as class x can form a cluster, and those classified as y can form another cluster. By this means, the comparing of two classifiers D_i, D_k can be seen as the comparing of two clusters C_i, C_k, where each class in D_i forms a cluster in C_i, and each class in D_k forms a cluster in C_k.

By using the same technique of counting the pairwise samples for comparing clustering from the contingency table, we can get C_{11} by comparing the samples in the same blocks. We get 4 blocks, so in each block we have $\frac{m(m-1)}{2}$ sample-pairs if there are m samples in this block. By summing up the sample-pairs counts in these 4 blocks, we get the C_{11}:

$$C_{11} = \frac{N_{xx}(N_{xx}-1)}{2} + \frac{N_{xy}(N_{xy}-1)}{2} + \frac{N_{yx}(N_{yx}-1)}{2} + \frac{N_{yy}(N_{yy}-1)}{2} \tag{5.4}$$

For calculating C_{10}, C_{01} and C_{00}, we apply the formulas we obtain before. For C_{10} we count sample-pairs on the same row but not on the same columns, for C_{01} we count sample-pairs on the same column but not on the same row, for C_{00} we count sample-pairs neither on the same column nor on the same row.

$$C_{00} = N_{xx}N_{yy} + N_{xy}N_{yx} \tag{5.5}$$

$$C_{10} = N_{xx}N_{xy} + N_{yy}N_{yx} \tag{5.6}$$

$$C_{01} = N_{xx}N_{yx} + N_{yy}N_{xy} \tag{5.7}$$

Using these terms instead of $C_{11}, C_{10}, C_{01}, C_{00}$ in clustering diversity measures, one can clear find its logical mechanism :

a. Wallace Indices

$$Wallace - 1 : W_i(C_i, C_k) = \frac{C_{11}}{C_{11} + N_{xx}N_{xy} + N_{yx}N_{yy}} \tag{5.8}$$

$$Wallace - 2 : W_k(C_i, C_k) = \frac{C_{11}}{C_{11} + N_{xx}N_{yx} + N_{xy}N_{yy}} \tag{5.9}$$

b. Fowlkes-Mallows Index

$$F(C_i, C_k) = (W_i(C_i, C_k)W_k(C_i, C_k))^{\frac{1}{2}} \tag{5.10}$$

c. Rand Index

$$R(C_i, C_k) = \frac{C_{11} + N_{xx}N_{yy} + N_{xy}N_{yx}}{\frac{C(C-1)}{2}} \tag{5.11}$$

d. Jacard Index

$$J(C_i, C_k) = \frac{C_{11}}{C(C-1) - 2(N_{xx}N_{yy} + N_{xy}N_{yx})} \tag{5.12}$$

e. Mirkin's Metric

$$K(C_i, C_k) = 2(N_{xx} + N_{yy})(N_{xy} + N_{yx}) \tag{5.13}$$

As we can see, most of the indices contain the terms that we cannot have a direct interpretation on the terms of $N_{11}, N_{10}, N_{01}, N_{00}$. The only exception is the Mirkin's metric, which can be written as :

$$K(C_i, C_k) = 2(N_{11} + N_{00})(N_{10} + N_{01}) \tag{5.14}$$

And, it is evident that Mirkin's metric has a strong relationship with the disagreement measure used in the classifier diversity.

$$K(C_i, C_k) = 2 \cdot DM \cdot N \cdot (N_{11} + N_{00}) \tag{5.15}$$

We intend to get the measure as close to Dis as possible by clustering. Without any class label available in clustering, we can still approximate $N_{11} + N_{00}$ by $N_{xx} + N_{yy}$. The problem resides on obtaining $N_{xx} + N_{yy}$, and they could not be obtained directly. We need a precondition to proceed the approximation, we suppose that both classes have the similar number of samples, i.e.,

$$N_{xx} = N_{yy} \tag{5.16}$$

$$N_{xy} = N_{yx} \tag{5.17}$$

The approximation is not straightforward, and we need to discuss three different cases below :

a. 50% diversity (according to the disagreement measure)

If classifiers disagree with each other on half of samples, we have $N_{xx} = N_{xy}$ and $N_{yy} = N_{yx}$, i.e., we have diversity as 50% by the definition of disagreement measure, as a result :

$$N_{xx} = N_{yy} = N_{xy} = N_{yx} \tag{5.18}$$

$$N_{xy}^2 + N_{yx}^2 = 2 \cdot N_{xx} \cdot N_{yy} \tag{5.19}$$

Consequently, using above two equations and eq. 5.1, we get :

$$\begin{aligned}(N_{xx} + N_{yy})^2 = N_{xx}^2 + N_{yy}^2 + 2 \cdot (N_{xx} \cdot N_{yy}) = \\ N_{xx}^2 + N_{yy}^2 + N_{xy}^2 + N_{yx}^2 = 2 \cdot C_{11} + N\end{aligned} \tag{5.20}$$

b. 0% diversity (according to the disagreement measure)

If both classifiers are almost identical, in this case $N_{xy} = N_{yx} = 0$, and $N_{xx} + N_{yy} = N$, thus we get $N_{xx} \cdot N_{yy} = \frac{N^2}{4}$, as a consequence :

$$\begin{aligned}(N_{xx} + N_{yy})^2 = N_{xx}^2 + N_{yy}^2 + 2 \cdot (N_{xx} \cdot N_{yy}) = \\ N_{xx}^2 + N_{yy}^2 + \frac{N^2}{2} = 2 \cdot C_{11} + N + \frac{N^2}{2}\end{aligned} \tag{5.21}$$

c. Diversity Parameter

The above two cases are easy to calculate, but are not suitable in most of the situations, where the diversity is neither 0 nor $\frac{1}{2}$. In fact, in practice most of the classifiers shall agree with each other on a large part of the samples but disagree on a smaller portion of them. i.e., the diversity shall be between 0 and $\frac{1}{2}$. To have a more general approximation, we set up a diversity parameter α, where $\alpha = 0$ will lead up to the case of 50% diversity, and $\alpha = 1$ means the diversity is 0:

$$(N_{xx} + N_{yy})^2 = 2 \cdot C_{11} + N + \alpha \cdot \frac{N^2}{2} \tag{5.22}$$

This is actually the approximation of $(N_{xx}+N_{yy})^2$. To satisfy the condition of this estimation, we simplify the situation in a 2-class classification problem, and we need to suppose that each class has similar number of samples $N_{11} + N_{00}$ by $N_{xx} + N_{yy}$. When this condition is satisfied, using eq. 5.15, we define the approximation of DM from Mirkin's Metric based on 2-clusters hypothesis as :

$$E(2C)_{i,k} \equiv \frac{K(C_i, C_k)}{2N \cdot (2 \cdot C_{11} + N + \frac{N^2}{2} \cdot \alpha)^{\frac{1}{2}}} \tag{5.23}$$

The hypothesis of the 2-clusters might not hold in most problems. However, we can extend the approximations with the multi-clusters hypothesis $E(MC)$ and with the multi-clusters with the concern of the variation of the information hypothesis $E(VI)$. Based on multiple clusters hypothesis, the approximation of DM would be :

$$E(MC)_{i,k} \equiv \frac{M \cdot (K(C_i, C_k))}{2N(((2 \cdot C_{11} + N) \cdot M^2 + \frac{N^2}{2} \cdot \alpha)^{\frac{1}{2}})} \tag{5.24}$$

where M is the number of clusters (See vi For details). Moreover, taking into account the variation of information, the approximation would be :

$$E(VI)_{i,k} \equiv \frac{\frac{M}{N} \cdot (\frac{K(C_i,C_k)}{2} - 2 * (t \cdot M - 1) \cdot (C_{11} + \frac{N}{2}))}{((2 \cdot C_{11} + N) \cdot t \cdot M^2 + \frac{N^2}{2} \cdot \alpha)^{\frac{1}{2}}} \tag{5.25}$$

where t is a measure concerning the variation of information (See appendix 6 for details). Now we do know that there is a close relationship between DM and Mirkin'n Metric, but there is still a question that needs to be answered: Is there a strong correlation between them?

To answer this question, we need to carry out the correlation measurements on synthetic data as well as on the UCI machine learning problems.

5.3 Correlation Measurements between the Classifier Diversity and the Clustering Diversity

5.3.1 Proof of Concept: Correlation Measurements with K-Nearest Prototype Classifiers on Synthetic Problems

At the previous sections we propose three modified clustering diversities derived from Mirkin's Metric to estimate the classifier diversity close to disagreement measure (DM): the estimation of diversity based on 2-clusters hypothesis (E(2C), eq. 5.23), the estimation of diversity based on multiple-clusters hypothesis (E(MC), eq. 5.24, see appendix 6), and the estimation of diversity based on multiple-clusters hypothesis but also corrected by variation of information (E(VI), eq. 5.25, see appendix 6), diversity parameter α is set as 0.3. To know whether these estimations make any sense, and whether there are correlations between the clustering diversities and DM, we first carried out the proof of concept on the synthetic data. 5 different synthetic data were generated with different numbers of

clusters and different numbers of classes; the clusters were formed with Gaussian distribution centered at the different centroids, these data were generated in a feature space of 6 dimensions (Table XLV; Fig. 44).

Table XLV

The synthetic databases generated for proof of concept

database	number of classes	cnumber of clusters	number of train samples	number of test samples	number of features	number of cardinality
Synthetic $2-2$	2	2	1000	1000	6	2
Synthetic $2-4$	2	4	1000	1000	6	2
Synthetic $2-6$	2	6	1000	1000	6	2
Synthetic $3-3$	3	3	1000	1000	6	2
Synthetic $4-4$	4	4	1000	1000	6	2

The basic classifiers were constructed based on Random Subspaces with fixed cardinality (cardinality $= 2$ in the experiments). For each database, we generated 15 classifiers with different feature subspaces. All centroids have the data points with the standard deviation equal to 1.

The synthetic data were generated so that all clusters were partly merged, and they had different degrees of the overlapping. Given the number of the clusters, each classifier got its centroids by applying simple K-Means clustering, then the classification was done by carrying out K-Nearest Prototypes (KNP), with $K = 1$. Once all classifiers were constructed, they were randomly selected as a member of ensemble. The probability of being selected is the same for all classifiers ($p = 0.3$). For each ensemble, we calculated the correlation between the disagreement measure (DM) (49) as the classifier diversity and the 9 following clustering diversities: 2 Wallace Indices, Fowlkes-Mallows Index, Rand Index, Jacard Index, Mirkin's Metric, and 3 different types of the estimations: the estimation for simple 2 classes problems (E(2C), eq. 5.23), the estimation for multiple classes problems (E(MC), eq. 5.24), and the estimation for multiple classes problems using the variation of information (E(VI), eq. 5.25).

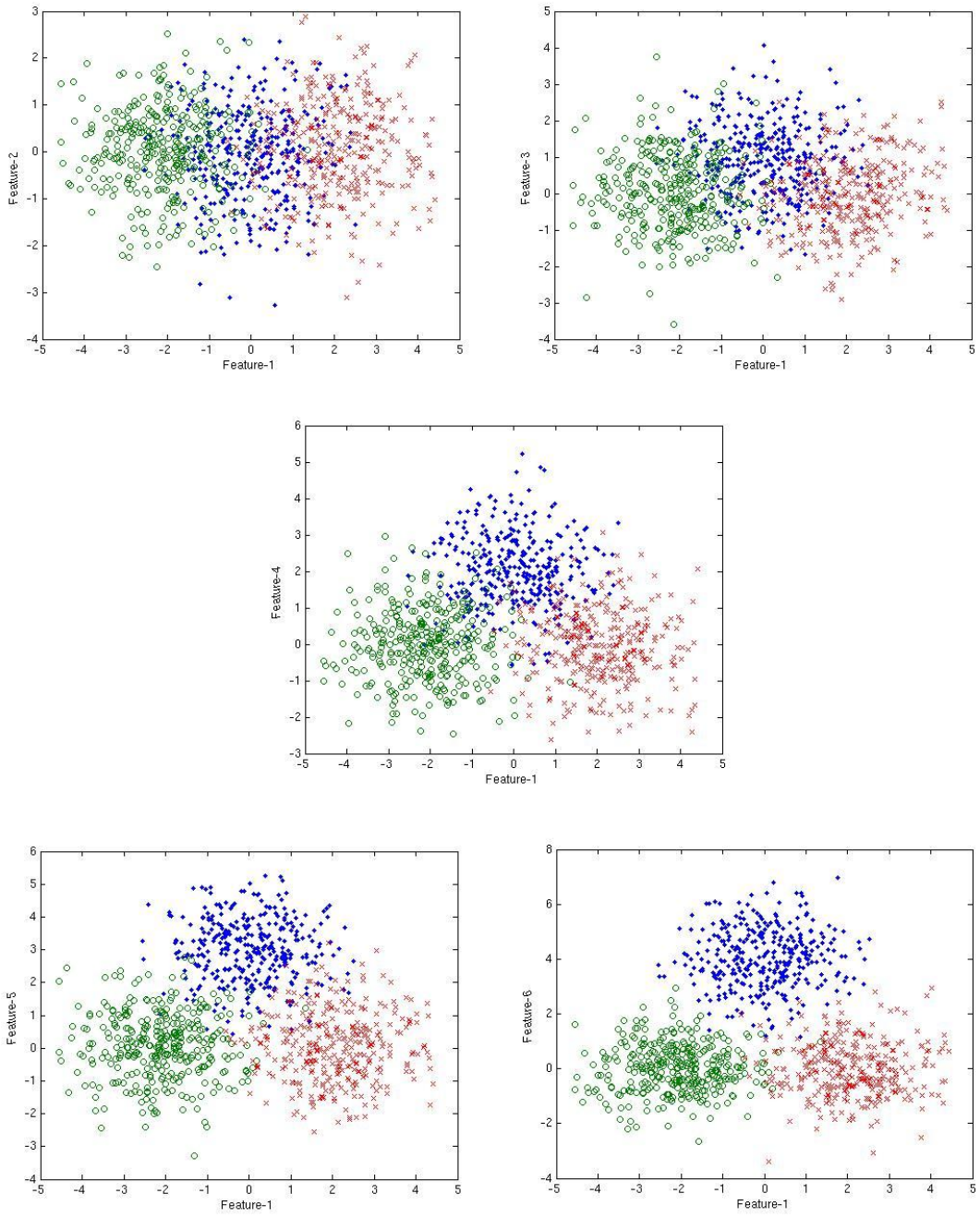

Figure 44 The data points in different feature subspaces. There are 3 classes and the feature dimension is 6

Table XLVI

The centroids of the generated synthetic clusters

Syn. $2-2$	class-1 (cluster-1)	class-2 (cluster-2)
	$(1, 1.2, 1.4, 1.6, 1.8, 2)$	$(1, 1, 1, 1, 1, 1)$

Syn. $2-4$	class-1 (cluster-1)	class-1 (cluster-2)	class-2 (cluster-3)	class-2 (cluster-4)
	$(-2, 1, 2, 2, 1, 0)$	$(2, -1, 3, -3, -1, -1)$	$(0, -1, 0, -2, -3, -2)$	$(2, 1, -2, 3, 0, -1)$

Syn. $2-6$	class-1 (cluster-1) class-2 (cluster-4)	class-1 (cluster-2) class-2 (cluster-5)	class-1 (cluster-3) class-2 (cluster-6)
	$(-3, 2, 6, 10, 14, 20)$ $(-3, -6, -10, -14, -20, -24)$	$(0, -4, -8, -12, -16, -22)$ $(0, 8, 12, 16, 22, 26)$	$(3, 6, 10, 14, 18, 24)$ $(3, -10, -14, -18, -24, -28)$

Syn. $3-3$	class-1 (cluster-1)	class-2 (cluster-2)	class-3 (cluster-3)
	$(0, 2, 4, 6, 12, 14)$	$(-2.1, 4, 6, 8, 14, 16)$	$(2.1, 6, 8, 10, 16, 18)$

Syn. $4-4$	class-1(cluster-1)	class-2 cluster-2)	class-3 (cluster-3)	class-4 (cluster-4)
	$(-2, 1, 1.5, 2, 2.5, 3)$	$(2, -1, -1.5, -2, -2.5, -3)$	$(4, -3, -4, -4, -6, -7)$	$(6, 5, 4, 4, 6, 7)$

As we expected, all three approximations have very strong correlations with DM (Fig. 45). E(2C) is slightly better than E(MC), but with the use of information, E(VI) achieves the best correlation with DM. Surprisingly, other original clustering diversity measures also show the strong correlations with DM, even though they do not go through any adjustment. Wallace-1 is the clustering diversity measure with the best correlation with DM, but E(VI) has very close performance (Table XLVII). To summarize, the proof of concept approves the estimation of DM from the Mirkin's Metric. It also suggests a strong correlation between DM and the clustering diversities using K-Nearest Prototypes as the classification method.

5.3.2 Correlation Measurements on UCI Machine Learning Problems

To understand more about the connections between the clustering diversities and DM, we measured their correlations on problems extracted from the UCI machine learning

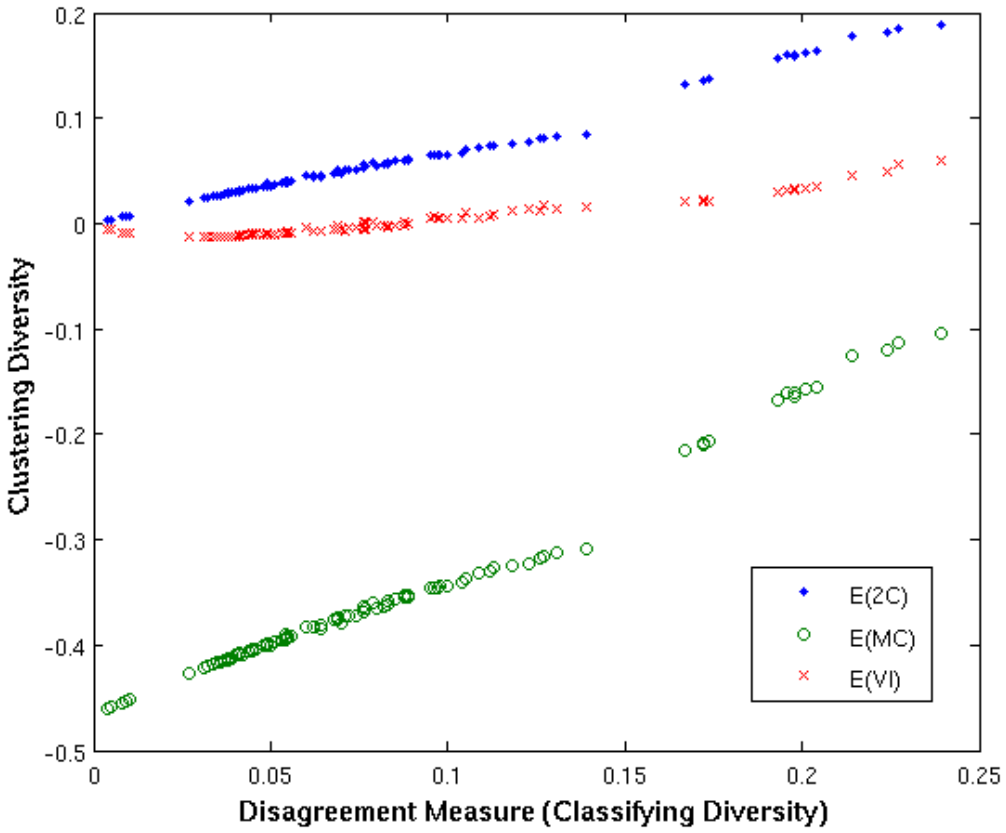

Figure 45 The relationships between DM and 3 approximations: E(2C), E(MC) and E(VI) on the synthetic data $4-4$

repository, and with more sophisticated classification algorithms. Several requirements are concerned for the selection of pattern recognition problems. First, to avoid the dimensional curse during the training, each database must have sufficient samples concerning its feature dimension. Second, to avoid identical samples to be trained in Random Subspace, only databases without symbolical features are used. Third, to simplify the problem we do not use databases with missing features. According to the requirements enlisted above, we carried out our experiments on 6 databases selected from the UCI Machine Learning Data Repository (Table XLVIII). For each of 6 databases and each of 5 classification algorithms, 18 classifiers were generated as the pool for base classifiers. Classifiers were then selected

Table XLVII

The correlations between the disagreement measure (DM) and the clustering diversities in the synthetics data. The nearest prototype (the centroid of the nearest cluster) is used as the classification method

	Mirkin's	E(2C)	E(MC)	E(VI)	W-1	W-2	F.M.	Rand	Jacard
Synthetic $2-2$	0.99	0.99	0.99	0.99	-0.99	-0.99	-0.99	-0.99	-0.99
Synthetic $2-4$	0.99	0.99	0.99	0.97	-0.98	-0.99	-0.99	-0.99	-0.99
Synthetic $2-6$	0.97	0.97	0.98	0.94	-0.94	-0.97	-0.96	-0.97	-0.97
Synthetic $3-3$	0.84	0.84	0.83	0.82	-0.82	-0.84	-0.78	-0.83	-0.84
Synthetic $4-4$	0.89	0.89	0.88	0.96	-0.96	-0.89	-0.88	-0.89	-0.89

Table XLVIII

The problems extracted from the UCI Machine Learning Data Repository for the correlation measurements between DM and the clustering diversities

database	number of classes	number of clusters	number of train samples	number of test samples	number of features	number of cardinality
Pima-Diabetes	2	3	384	384	8	4
Liver-Disorders	2	5	144	144	6	3
Wisconsin Breast-Cancer	2	12	284	284	30	5
Wine	3	4	88	88	13	6
Image Segmentation	7	53	210	2100	19	4
Letters Recognition	26	87	10000	10000	16	12

from this pool to construct ensembles. In our experiments, we apply Normal Densities Based Linear Classifiers (LDC), Quadratic Discriminant Classifiers (QDC), K-Nearest Neighbors Classifiers (KNN), Parzen Windows Classifiers (PWC) and Radial Basis Neural Network Classifiers (RBN) (19) for the classification tasks. For each test, we randomly selected classifiers to construct the ensemble, and each classifier had the same probability ($p = 0.3$) to be chosen as a member of Ensemble of classifiers. Thus the correlations were measured for ensembles with the different numbers of classifiers, and then the mean values of correlations were calculated. To get the accurate measure, for each database and each classification algorithm, 3000 ensembles were constructed for the correlation measurement.

Each classifier was created in different feature subspace and used all of training samples. We carried out the correlation measurement between the disagreement measure (DM) (49) as the classifier diversity and 9 clustering diversities, including 2 Wallace Indices, Fowlkes-Mallows Index, Rand Index, Jacard Index, Mirkin's Metric, and 3 modified clustering indices (E(2C), eq. 5.23; E(MC), eq. 5.24; E(VI), eq. 5.25) derived from Mirkin's Metric, the diversity parameter α is set as 0.3.

Table XLIX

The correlations between the clustering diversities and the disagreement measure (DM) in UCI databases

	Mirkin's	E(2C)	E(MC)	E(VI)	W-1	W-2	F.M.	Rand	Jacard
Pima-Diabetes	0.40	0.41	0.37	0.40	-0.40	-0.39	-0.32	-0.37	-0.40
Liver-Disorders	0.57	0.58	0.58	0.48	-0.48	-0.58	-0.56	-0.58	-0.57
Wisconsin Breast-Cancer	0.61	0.64	0.69	0.72	-0.72	-0.61	-0.63	-0.62	-0.61
Wine	0.56	0.56	0.57	0.52	-0.52	-0.57	-0.58	-0.57	-0.56
Image Segmentation	0.38	0.38	0.34	0.20	-0.20	-0.35	-0.37	-0.37	-0.38
Letters Recognition	0.52	0.52	0.58	0.51	-0.51	-0.57	-0.56	-0.57	-0.52

First, we notice there are still correlations between the three approximations and DM (Table XLIX), but much less strong than those we observed in the synthetic data with KNP (Table XLVII). Second, we note that in general, E(MC) has the better performance with E(2C), but with the use of the variation of information, E(VI) does not improve the correlation and apparently worse than E(MC). This indicates that the variation of information might differ hugely from one cluster-pair to another cluster-pair. Third, other clustering diversity measures also shows the comparable correlation with DM, but none of them outperforms E(MC). Since we used various classification algorithms, including more sophisticated ones such as RBN, QDC and PWC, the boundaries between classes are more complicated than the simple clustering can define, this might be a major cause of the loss of the correlation between clustering diversity measures and DM.

So far, we now know that, in general, there exist correlations between DM - a classifier diversity measure - and the clustering diversities in Random subspace. We know that the

data diversity can lead to the classifier diversity, and this data diversity can be measured in Random subspace using clustering diversity.

5.4 Discussion

In this work, we examined whether the clustering diversity can represent the data diversity of different feature subsets in random subspaces, and whether the use of the clustering diversity as the data diversity measure could allow us to apply a classifier-free ensemble selection scheme.

For the use of the clustering diversity, we show that there is a strong connection between the Disagreement Measure, a classifier diversity measure, and Mirkin's metric, one of the clustering diversity measures. We derived the $E(2C)$, $E(MC)$ and $E(VI)$ to better approximate Disagreement Measure from Mirkin's metric. The proposed approximations were shown to have the strong correlations with Disagreement Measure. We also observed the strong correlations between other clustering diversity measures and Disagreement Measure. The correlations between the clustering diversity and Disagreement Measure indicate that the data diversity can be somehow approximated even before the construction of classifiers for the Random subspace.

5.5 Conclusion

In general, the classifier diversities are used to construct an ensemble for the better classification, and the clustering diversities are used to construct an ensemble for the better clustering. They have different purposes, and their relationship was not fully investigated. In this work, we conclude that there is a close relationship between Mirkin's metric and Disagreement Measure, and we further derived the approximation of Disagreement Measure based on Mirkin's metric. We observed strong correlations between the Disagreement Measure and most clustering diversities.

Given that this is the first exploratory work on the relationship between classifier diversities the clustering diversities, we tried to figure out the correlations between them and carried out necessary experiments. Due to the complexity of the derivation of $E(MC)$ and $E(VI)$, we do not include them in this appendix, but leave them in the appendix 6 for interested readers.

APPENDIX 6

The Approximation of the Disagreement Measure Based on Mirkin's Metric

The appendix 5 demonstrate experimentally that there is a strong correlation between classifying diversities and clustering diversities. We scanned most classifying diversities and clustering diversities, and conclude that we might figure out a close relationship between Mirkin's metric, a clustering diversity measure, and Disagreement Measure, a classifying diversiy measure.

In this appendix, we try to approximate Disagreement Measure using only Mirkin's metric. The objective is to approximate a possible classifier diversity when only clustering result is given. Apparently, since there is no available label during the clustering, this approximation is under a number of assumption. However, by carrying out these approximations and measuring the correlations between the approximations and the true classifier diversity, we might have an insight into the circumstances under which an approximation of a classifier diversity is feasible, under which a strong correlation with a classifier diversity exists, and under which we can carry out an classifier-free ensemble selection that presented in the chapter 6 in this thesis.

For this purpose, we propose three different approximations of Disagreement Measure based on Mirkin's metric. All three approximations, $E(2C)$, $E(MC)$ and $E(VI)$, are based on various circumstances. Note that $E(2C)$ has been derived in the appendix 5, as well as the correlation measurements between three approximations and Disagreement Measure. In this appendix, we simply give some details on how the approximations of $E(MC)$ and $E(VI)$ are obtained.

To justify the need of $E(MC)$ and $E(VI)$, we can point out that the data points belonging to one class will, in general, form more than one cluster, and thus the hypothesis made for $E(2C)$ was extremely simplified. We are interested in having better approximation of the Disagreement Measure from the Mirkin's Metric based on more general conditions. These approximations are somehow complicated, and due to the limit of the space we are unable to provide all the details but only the important concepts, assumptions and derivations.

At the end, we need to mention that these approximations are not essential for the classifier-free ensemble selection scheme introduced in chapter 6. However, since they do suggest that there is a strong relationship between a classifier diversity and a clustering diversity, we decided to add these approximations in this appendix for interested readers.

6.1 Extension on Multi-clusters clustering: $E(MC)$

In the appendix 5 we assume that the data classified as two classes can be clustered into two clusters. This assumption, however, is simplified, and in real problems we usually have more than one cluster for each class. To deal with this problem, we have to reformulate our hypothesis. We suppose that, if the data can be classified into two classes based on a classifier, then, it is possible that they can be clustered into several clusters. In this case, each class might have more than one cluster, but the members of a cluster belong to the same class. For the development in this section, we make the following assumptions:

a. The data set is a 2-class problem.

b. The data set can be perfectly partitioned into K clusters, $K \geq 2$.

c. For each cluster, all the samples in one cluster belong to the same class.

d. Both classes have similar number of samples.

e. Both classes have similar number of clusters.

f. For the samples classified as the same class by both classifiers, they are clustered in the same cluster by both clusterings.

We assume that, for classifier D_i, samples classified as the class x are clustered into M_{xo} clusters, with $N_{xx} + N_{xy}$ samples in total. The samples classified as the class y are clustered into M_{yo} clusters, in this case we have $N_{yy} + N_{yx}$ samples. For classifier D_k, the

samples classified as the class x are clustered into M_{ox} clusters, with $N_{xx} + N_{yx}$ samples. The samples classified as the class y are clustered into M_{oy} clusters, with $N_{yy} + N_{xy}$ samples (Fig.46). But the relation between clusters is quite complicated. It depends on

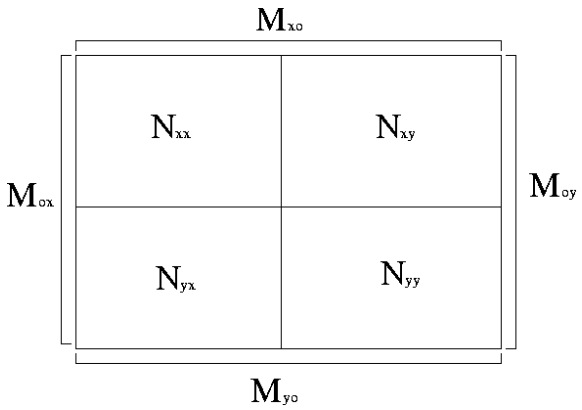

Figure 46 In a two class problem, with class x and class y, each class can form multiple clusters. For classifier D_i, $N_{xx} + N_{xy}$ samples are classified as class x and clustered into M_{xo} clusters, and $N_{yx} + N_{yy}$ samples are classified as class y and clustered into M_{yo} clusters; for classifier D_k, $N_{xx} + N_{yx}$ samples are classified as class x and clustered into M_{ox} clusters, and $N_{xy} + N_{yy}$ samples are classified as class y and clustered into M_{oy} clusters

the geometrical properties in feature space, and cannot be analyzed easily. Still, we can set measures about entropy, mutual information and variation of information, and continue our discussion. First we have to assume that for N_{ij} samples and M_{kl} clusters, each cluster has the size of $\frac{N_{ij}}{M_{kl}}$ samples. It is reasonable when each cluster has the similar distance of radius and the samples have similar density of the distribution. This is quite fair especially in K-Means clustering.

6.1.1 Concept of mutual Information and Variation of Information

Then we introduce the concept of the mutual information and the variation of information (72) here. For a clustering C, suppose that we have K clusters, we can calculate the global

entropy of this clustering by summing up the entropy of each cluster :

$$H(C) = -\sum_{k=1}^{K} P(k) \log P(k) \tag{6.1}$$

where $P(k)$ is the probability that a sample belongs to cluster k. Then, for two clusterings C and C^*, we can make the definition of the probability that a sample belongs to cluster C_k in clustering C and belongs to C_{k^*} in clustering C^*:

$$P(k, k^*) = \frac{|C_k \bigcap C_{k^*}|}{n} \tag{6.2}$$

where n is the number of total samples. Then the mutual information between clustering C and C^* can be defined as :

$$I(C, C^*) = \sum_{k=1}^{K} \sum_{k^*=1}^{K^*} P(k, k^*) \log \frac{P(k, k^*)}{P(k)P(k^*)} \tag{6.3}$$

Apparently this will satisfy :

$$I(C, C^*) \leq min(H(C), H(C^*)) \tag{6.4}$$

The variation of information (72) is defined as :

$$VI(C, C^*) = H(C) + H(C^*) - 2I(C, C^*) \tag{6.5}$$

Considering our problem, we add four variations of the information measures to evaluate the relations between four groups of clustering. The samples are labeled as N_{xx}, N_{xy}, N_{yx}, and N_{yy}. For C_i clustering, $N_{xx} + N_{xy}$ samples are clustered into M_{xo} clusters, and $N_{yx} + N_{yy}$ samples are clustered into M_{yo} samples. For C_k clustering, $N_{xx} + N_{yx}$ samples are clustered into M_{ox} samples, and $N_{xy} + N_{yy}$ samples are clustered into M_{oy} samples (Table L). Their values reflect the degree of the variation of information. When

Table L

Definition of the four variations of information measures

	$M_{xo}(for\ N_{xx}\&N_{xy})$	$M_{yo}(for\ N_{yy}\&N_{yx})$
$M_{ox}(for\ N_{xx}\&N_{yx})$	M_{xx}	M_{yx}
$M_{oy}(for\ N_{yy}\&N_{xy})$	M_{xy}	M_{yy}

the clustering M_{xo} and M_{ox} are totally random, for $P(k), k \in M_{xo}$, and $P(k^*), k^* \in M_{ox}$, we have $P(k, k^*) = P(k)P(k^*)$, so that $I(C, C^*) = 0, M_{xo} \in C, M_{ox} \in C^*$. The variation of information $VI(C, C^*) = H(C) + H(C^*)$, and we set the value of variation of the information measure M_{xx} as $M_{xo} \cdot M_{ox}$.

However, when M_{xo} and M_{ox} have the same number of clusters, i.e., $K = K^*$, and these clusters maintain the same partition for all N_{xx} samples, the clusterings are identical for sharing samples, in this case, $P(k, k^*) = P(k) = P(k^*)$, and $I(C, C^*) = H(C) = H(C^*), M_{xo} \in C, M_{ox} \in C^*$, so we get the zero variation of information, $VI(C, C^*) = 0$. As a result, we set the value of M_{xx} as $M_{xx} = M_{xo} = M_{ox}$, and the similar definition for other three variations of information measures. Later we will explain what the use of these variations of information measures is.

$$min(M_{xo}, M_{ox}) \leq M_{xx} \leq M_{xo} \cdot M_{ox} \tag{6.6}$$

$$min(M_{xo}, M_{oy}) \leq M_{xy} \leq M_{xo} \cdot M_{oy} \tag{6.7}$$

$$min(M_{oy}, M_{yo}) \leq M_{yy} \leq M_{oy} \cdot M_{yo} \tag{6.8}$$

$$min(M_{yo}, M_{ox}) \leq M_{yx} \leq M_{yo} \cdot M_{ox} \tag{6.9}$$

6.1.2 Decomposition of the Counting of Sample-Pairs

According to the definition of pairwise samples measure, we can calculate the value of C_{11}, C_{10}, C_{01} and C_{00}. These calculations will need some maneuvers, and we detail the process of the decomposition of these terms below.

a. Decomposition of the Sample-Pairs in C_{11}

For N_{xx}, there are $M_{xo} \cdot M_{ox}$ blocks. Using eq. 5.16, eq. 5.17, suppose that each cluster has the same number of samples, we just simply set for each block there are $S = \frac{N_{xx}}{M_{xo} \cdot M_{ox}}$ samples, with $B = M_{xo} \cdot M_{ox}$ blocks. As a result, we can calculate the number of sample-pairs in C_{11} for samples labels as N_{xx}:

$$C_{11}(N_{xx}) = \frac{B \cdot S \cdot (S-1)}{2} = \frac{M_{xo} \cdot M_{ox}(\frac{N_{xx}}{M_{xo} \cdot M_{ox}})(\frac{N_{xx}}{M_{xo} \cdot M_{ox}} - 1)}{2} = \frac{N_{xx}(\frac{N_{xx}}{M_{xo} \cdot M_{ox}} - 1)}{2} \tag{6.10}$$

As we denote $M_{xx} = M_{xo} \cdot M_{ox}$, we can re-write:

$$C_{11}(N_{xx}) = \frac{(\frac{N_{xx}^2}{M_{xx}} - N_{xx})}{2} \tag{6.11}$$

We do the same process for $C_{11}(N_{xy})$, $C_{11}(N_{yx})$, $C_{11}(N_{yy})$, and calculate their sum to obtain C_{11}:

$$C_{11} = \widetilde{C_{11}} = C_{11}(N_{xx}) + C_{11}(N_{xy}) + C_{11}(N_{yx}) + C_{11}(N_{yy}) \tag{6.12}$$

$$= \frac{N_{xx}^2}{2 \cdot M_{xx}} - \frac{N_{xx}}{2} + \frac{N_{yy}^2}{2 \cdot M_{yy}} - \frac{N_{yy}}{2} + \frac{N_{xy}^2}{2 \cdot M_{xy}} - \frac{N_{xy}}{2} + \frac{N_{yx}^2}{2 \cdot M_{yx}} - \frac{N_{yx}}{2} \tag{6.13}$$

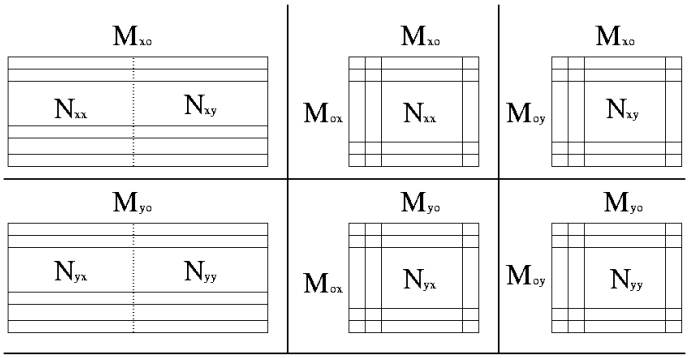

Figure 47 Assuming each class can form multiple clusters, we hope to derive the relation between the clustering diversity and the classifier diversity. We show an example of how to calculate C_{10}: For 4 partitions, 6 different relationships must be considered and calculated. The similar calculation can be applied on C_{01}

b. Decomposition of the Sample-Pairs in C_{10} and C_{01}

The similar analysis can be used to find C_{10} and C_{01}. Remember that we have multiple clusters, but all these clusters can be analyzed via 4 blocks: N_{xx}, N_{xy}, N_{yx}, and N_{yy} (See Fig.46). For C_i clustering, $N_{xx} + N_{xy}$ samples are clustered into M_{xo} clusters, and $N_{yx} + N_{yy}$ samples are clustered into M_{yo} clusters. For C_k clustering, $N_{xx} + N_{yx}$ samples are clustered into M_{ox} clusters, and $N_{xy} + N_{yy}$ samples are clustered into M_{oy} clusters.

In order to calculate C_{10}, one must consider the sample-pairs clustered into different clusters by clustering C_i but into the same clusters by clustering C_k (Fig.47). We have 6 cases here: the sample-pairs in N_{xx}, the sample-pairs in N_{xy}, the sample-pairs in N_{xx} and in N_{xy}, the sample-pairs in N_{yx}, the sample-pairs in N_{yy}, the sample-pairs in N_{yx} and in N_{yy} (See Fig. 46 and Fig.47). For one thing, considering $C_{10}(N_{xx})$, we need to count the sample-pairs on $B = M_{xo} \cdot M_{ox}$ blocks among N_{xx} samples, each block has $S = \frac{N_{xx}}{M_{xo} \cdot M_{ox}}$ samples, for C_{10} we also need to count the number of samples of each cluster clustered by C_k (See Fig. 47), denote as $S_k = \frac{N_{xx}}{M_{xo}}$. According to the definition of C_{10}, we need to take into consideration the number for the cluster-pairs in the same cluster under C_i but not under C_k. Since for each sample, there are $(S_k - S)$ other samples with which it can form sample-pairs in the same cluster under C_i but not under C_k, we can count the total sample-pairs as :

$$\frac{B \cdot S \cdot (S_k - S)}{2} = \frac{M_{xo} \cdot M_{ox} \frac{N_{xx}}{M_{xo} \cdot M_{ox}} (\frac{N_{xx}}{M_{xo}} - \frac{N_{xx}}{M_{xo} \cdot M_{ox}})}{2} \tag{6.14}$$

The similar process can be realized on $C_{10}(N_{xy})$, $C_{10}(N_{xx}, N_{xy})$, $C_{10}(N_{yx})$, $C_{10}(N_{yy})$, $C_{10}(N_{yx}N_{yy})$. We write a a short summary for sample-pairs accounted for C_{10} (Table LI).

By summing them up, and by denoting $M_{xx} = M_{ox} \cdot M_{xo}$,$M_{xy} = M_{xo} \cdot M_{oy}$, $M_{yx} = M_{ox} \cdot M_{yo}$,$M_{yy} = M_{oy} \cdot M_{yo}$, we can get C_{10}

$$C_{10} = \widehat{C_{10}} + \widetilde{C_{10}} \tag{6.15}$$

$$\widehat{C_{10}} = \frac{N_{xx}N_{xy}}{M_{xo}} + \frac{N_{yy}N_{yx}}{M_{yo}} \tag{6.16}$$

Table LI

Decomposition of C_{10} by Fig.47

source of C_{10}	number of blocks	number of samples	number of sample-pairs per block	number of total sample-pairs
$C_{10}(N_{xx})$	$M_{xo} \cdot M_{ox}$	N_{xx}	$\frac{N_{xx}}{M_{xo} \cdot M_{ox}} (\frac{N_{xx}}{M_{xo}} - \frac{N_{xx}}{M_{xo} \cdot M_{ox}})$	$\frac{M_{xo} \cdot M_{ox} \frac{N_{xx}}{M_{xo} \cdot M_{ox}} (\frac{N_{xx}}{M_{xo}} - \frac{N_{xx}}{M_{xo} \cdot M_{ox}})}{2}$
$C_{10}(N_{xy})$	$M_{xo} \cdot M_{oy}$	N_{xy}	$\frac{N_{xy}}{M_{xo} \cdot M_{oy}} (\frac{N_{xy}}{M_{xo}} - \frac{N_{xy}}{M_{xo} \cdot M_{oy}})$	$\frac{M_{xo} \cdot M_{oy} \frac{N_{xy}}{M_{xo} \cdot M_{oy}} (\frac{N_{xy}}{M_{xo}} - \frac{N_{xy}}{M_{xo} \cdot M_{oy}})}{2}$
$C_{10}(N_{xx}, N_{xy})$	M_{xo}	$N_{xx} \& N_{xy}$	$\frac{N_{xx}}{M_{xo}} \frac{N_{xy}}{M_{xo}}$	$\frac{M_{xo} \frac{N_{xx}}{M_{xo}} \frac{N_{xy}}{M_{xo}}}{2}$
$C_{10}(N_{yx})$	$M_{yo} \cdot M_{ox}$	N_{yx}	$\frac{N_{yx}}{M_{ox} \cdot M_{yo}} (\frac{N_{yx}}{M_{yo}} - \frac{N_{yx}}{M_{yo} \cdot M_{ox}})$	$\frac{M_{ox} \cdot M_{yo} \frac{N_{yx}}{M_{ox} \cdot M_{yo}} (\frac{N_{yx}}{M_{yo}} - \frac{N_{yx}}{M_{yo} \cdot M_{ox}})}{2}$
$C_{10}(N_{yy})$	$M_{yo} \cdot M_{oy}$	N_{yy}	$\frac{N_{yy}}{M_{yo} \cdot M_{oy}} (\frac{N_{yy}}{M_{yo}} - \frac{N_{yy}}{M_{yo} \cdot M_{oy}})$	$\frac{M_{yo} \cdot M_{oy} \frac{N_{yy}}{M_{yo} \cdot M_{oy}} (\frac{N_{yy}}{M_{yo}} - \frac{N_{yy}}{M_{yo} \cdot M_{oy}})}{2}$
$C_{10}(N_{yx} N_{yy})$	M_{yo}	$N_{yy} \& N_{yx}$	$\frac{N_{yy}}{M_{yo}} \frac{N_{yx}}{M_{yo}}$	$\frac{M_{yo} \frac{N_{yy}}{M_{yo}} \frac{N_{yx}}{M_{yo}}}{2}$

$$\begin{aligned} \widetilde{C_{10}} = & \frac{N_{xx}^2}{2}(\frac{1}{M_{xo}} - \frac{1}{M_{xx}}) + \frac{N_{xy}^2}{2}(\frac{1}{M_{xo}} - \frac{1}{M_{xy}}) \\ & + \frac{N_{yy}^2}{2}(\frac{1}{M_{yo}} - \frac{1}{M_{yy}}) + \frac{N_{yx}^2}{2}(\frac{1}{M_{yo}} - \frac{1}{M_{yx}}) \end{aligned} \tag{6.17}$$

For C_{01}, we do the similar calculation as C_{10}, then we get :

$$C_{01} = \widehat{C_{01}} + \widetilde{C_{01}} \tag{6.18}$$

$$\widehat{C_{01}} = \frac{N_{xx} N_{yx}}{M_{ox}} + \frac{N_{yy} N_{xy}}{M_{oy}} \tag{6.19}$$

$$\begin{aligned} \widetilde{C_{01}} = & \frac{N_{xx}^2}{2}(\frac{1}{M_{ox}} - \frac{1}{M_{xx}}) + \frac{N_{xy}^2}{2}(\frac{1}{M_{oy}} - \frac{1}{M_{xy}}) \\ & + \frac{N_{yy}^2}{2}(\frac{1}{M_{oy}} - \frac{1}{M_{yy}}) + \frac{N_{yx}^2}{2}(\frac{1}{M_{ox}} - \frac{1}{M_{yx}}) \end{aligned} \tag{6.20}$$

c. Decomposition of the Samples-Pair in C_{00}

For the calculation of C_{00}, the similar method can be used. But it is somehow more complicated. Because in each block N_{xx}, N_{xy}, N_{yx}, N_{yy}, we get multiple clusters. So the samples in the same block may be in different clusters under both clustering C_i and C_k. That means we have 10 different cases here: the sample-pairs in N_{xx} and in N_{yy}, and the sample-pairs in N_{xy} and in N_{yx}, all of which will contribute to C_{00}. But for the sample-pairs in N_{xx} and in N_{xy}, the sample-pairs in N_{xx} and in N_{yx}, the sample-pairs in N_{yy} and in N_{xy}, the sample-pairs in N_{yy} and in N_{yx}, most sample-pairs will contribute to C_{00}, but not all of them. For the sample-pairs in N_{xx}, the sample-pairs in N_{xy}, the sample-pairs in N_{yy}, the sample-pairs in N_{yx}, there are fewer sample-pairs will become C_{00}. By summing them up, we get the value of C_{00}:

$$C_{00} = \ddot{C_{00}} + \widehat{C_{00}} + \widetilde{C_{00}} \tag{6.21}$$

$$\ddot{C_{00}} = N_{xx}N_{yy} + N_{xy}N_{yx} \tag{6.22}$$

$$\begin{aligned}\widehat{C_{00}} = {} & N_{xx}N_{xy}(1 - \frac{1}{M_{xo}}) + N_{xx}N_{yx}(1 - \frac{1}{M_{ox}}) \\ & + N_{yy}N_{xy}(1 - \frac{1}{M_{oy}}) + N_{yy}N_{yx}(1 - \frac{1}{M_{yo}})\end{aligned} \tag{6.23}$$

$$\begin{aligned}\widetilde{C_{00}} = {} & \frac{N_{xx}^2}{2}(1 - \frac{1}{M_{xo}} - \frac{1}{M_{ox}} + \frac{1}{M_{xx}}) \\ & + \frac{N_{xy}^2}{2}(1 - \frac{1}{M_{xo}} - \frac{1}{M_{oy}} + \frac{1}{M_{xy}}) \\ & + \frac{N_{yy}^2}{2}(1 - \frac{1}{M_{yo}} - \frac{1}{M_{oy}} + \frac{1}{M_{yy}}) \\ & + \frac{N_{yx}^2}{2}(1 - \frac{1}{M_{yo}} - \frac{1}{M_{ox}} + \frac{1}{M_{yx}})\end{aligned} \tag{6.24}$$

We note that all these terms satisfy that :

$$C_{11} + C_{10} + C_{01} + C_{00} = \frac{N(N-1)}{2} \tag{6.25}$$

where N is the number of total samples.

$$N = N_{xx} + N_{xy} + N_{yx} + N_{yy} \tag{6.26}$$

As we look at these terms, we can find that all $\widehat{C_{01}}$, $\widehat{C_{10}}$, and $\widehat{C_{00}}$ depend on the number of clusters M_{xo}, M_{ox}, M_{yo} and M_{oy}, but there is no terms as M_{xx}, M_{xy}, M_{yx}, M_{yy}, i.e., they are independent from the variation of information. $\widetilde{C_{00}}$, $\widetilde{C_{10}}$, $\widetilde{C_{01}}$ and $\widetilde{C_{00}}$ contain the terms as M_{xx}, M_{xy}, M_{yx}, M_{yy}, i.e., as a result, they depend heavily on the variation of information. $\ddot{C_{00}}$ is the original term of C_{00} for two-clusters problems, it is absolutely independent. When the number of clusters increases, it is clear that there is a huge increase in C_{00}, too. The variations of information measure, bounded by the number of clusters, will also increase, and this lead to a quick decrease of C_{11}. The increase of the number of clusters will also lead to the decrease of C_{10} and C_{01}, but if the variation of information is low, then we have a lower slope in the curve of its decline, and vice versa.

It is interesting because the measures of the variation of information, based on the geometrical properties of feature space and bounded by the number of clusters do matter if we consider the diversity of clustering. If we look at Mirkin's metric, we can get :

$$\frac{K(C_i, C_k)}{2} = (C_{10} + C_{01}) = \widehat{C_{01}} + \widetilde{C_{01}} + \widehat{C_{10}} + \widetilde{C_{10}} \tag{6.27}$$

6.1.3 Approximation of Classifier Diversity in Multi-Clusters Clustering

In cases with very low variation of information, i.e., if we assume that for samples classified as the same class by both classifiers, they are clustered in the same cluster by both clusterings, we have $M_{xy} \simeq \{M_{xo}, M_{oy}\}$, $M_{yx} \simeq \{M_{yo}, M_{ox}\}$, $M_{xx} \simeq \{M_{xo}, M_{ox}\}$, $M_{yy} \simeq \{M_{yo}, M_{oy}\}$, we get :

$$\frac{K(C_i, C_k)}{2} = C_{01} + C_{10} \tag{6.28}$$

If two clustering use the same number of clusters, i.e., $M_{xo} + M_{yo} \simeq M_{ox} + M_{oy}$, and if two classes have the similar number of samples, i.e., $M_{xo} \simeq M_{yo}$ and $M_{ox} \simeq M_{oy}$, we get $M \simeq \{M_{yo}, M_{ox}, M_{oy}, M_{xo}\}$, so the Mirkin's metric will become :

$$\frac{K(C_i, C_k)}{2} = \frac{(N_{xx} + N_{yy})(N_{xy} + N_{yx})}{M} + 2 \cdot (C_{11} + \frac{N}{2}) \cdot (M - 1) \tag{6.29}$$

This is easy to transform, given that $N_{xx} + N_{yy} = N_{11} + N_{00}$, and $N_{xy} + N_{yx} = N_{10} + N_{01}$.

$$\frac{K(C_i, C_k)}{2} = \frac{(N_{xx} + N_{yy})(N_{10} + N_{01})}{M} + 2 \cdot (C_{11} + \frac{N}{2}) \cdot (M - 1) \tag{6.30}$$

$$= \frac{N \cdot (N_{xx} + N_{yy})(DM_{i,k})}{M} + 2 \cdot (C_{11} + \frac{N}{2}) \cdot (M - 1) \tag{6.31}$$

In the condition mentioned before, the term $N_{xx} + N_{yy}$ can be derived from C_{11}. Given the diversity parameter α, we can estimate that :

$$N_{xx} + N_{yy} = ((2 \cdot C_{11} + N) \cdot M^2 + \frac{N^2}{2} \cdot \alpha)^{\frac{1}{2}} \tag{6.32}$$

So now we can write :

$$\frac{K(C_i, C_k)}{2} = \frac{N(((2 \cdot C_{11} + N) \cdot M^2 + \frac{N^2}{2} \cdot \alpha)^{\frac{1}{2}})(DM_{i,k})}{M} \tag{6.33}$$

Finally, DM can be approximated by the clustering diversity, Mirkin's metric based on multiple clusters hypothesis :

$$E(MC)_{i,k} \equiv \frac{M \cdot (K(C_i, C_k))}{2N(((2 \cdot C_{11} + N) \cdot M^2 + \frac{N^2}{2} \cdot \alpha)^{\frac{1}{2}})} \tag{6.34}$$

6.2 Extension on Clustering with Variation of Information: $E(VI)$

For the development in this section, we make the following assumptions:

a. The data set is a 2-class problem.

b. The data set can be perfectly partitioned into K clusters, $K \geq 2$.

c. For each cluster, all samples in one cluster belong to the same class.

d. Both classes have similar number of samples.

e. Both classes have similar number of clusters.

f. The variation of information is similar for all cluster-pairs.

When the clustering has high variation of information, they form very different clusters for the samples in the same class. To well understand its properties, first we have to assume that the variation of information measures can be represented with variation coefficients t_1, t_2, t_3, t_4. The variation coefficient t_1 is a factor that concerns the difference between M_{xx} and $M_{xo} \cdot M_{ox}$, and the variation coefficient t_2 is a factor that concerns the difference

between M_{xy} and $M_{xo} \cdot M_{oy}$, etc. We define t_1, t_2, t_3 and t_4 as follows :

$$M_{xx} = t_1 \cdot M_{xo} \cdot M_{ox} \tag{6.35}$$
$$\max(\frac{1}{M_{xo}}, \frac{1}{M_{ox}}) \leq t_1 \leq 1 \tag{6.36}$$
$$M_{xy} = t_2 \cdot M_{xo} \cdot M_{oy} \tag{6.37}$$
$$\max(\frac{1}{M_{xo}}, \frac{1}{M_{oy}}) \leq t_2 \leq 1 \tag{6.38}$$
$$M_{yy} = t_3 \cdot M_{oy} \cdot M_{yo} \tag{6.39}$$
$$\max(\frac{1}{M_{yo}}, \frac{1}{M_{oy}}) \leq t_3 \leq 1 \tag{6.40}$$
$$M_{yx} = t_4 \cdot M_{yo} \cdot M_{ox} \tag{6.41}$$
$$\max(\frac{1}{M_{yo}}, \frac{1}{M_{ox}}) \leq t_4 \leq 1 \tag{6.42}$$

Given no knowledge about t_1, t_2, t_3 and t_4, we need to simplify the calculation and thus suppose that the variation of information will have similar values for different cluster-pairs. Given two clustering, we assume that :

$$t_1 = t_2 = t_3 = t_4 = t \tag{6.43}$$

As we stated before, we assume that both clustering have the similar number of clusters, each class has the similar number of samples. These assumptions are necessary to deal with M_{xo}, M_{ox}, M_{yo}, M_{oy}. When the numbers of clusters in two clustering are similar, we get :

$$M = \{M_{yo}, M_{ox}, M_{oy}, M_{xo}\} \tag{6.44}$$

Here, M is supposed to be the number of clusters for correct and for incorrect classified samples, so M can be estimated as :

$$M = \frac{N}{2} \tag{6.45}$$

Using the variation coefficient t as a general variation coefficient, we can simplify the calculation :

$$t \cdot M^2 = \{M_{xx}, M_{xy}, M_{yx}, M_{yy}\} \tag{6.46}$$

$$\frac{1}{M} \leq t \leq 1 \tag{6.47}$$

Actually, under our framework of problems, the entropy and the mutual information can be re-written as :

$$H(C) = \sum_{k=1}^{K} \frac{1}{M} \log M \tag{6.48}$$

$$I(C, C^*) = \sum_{k=1}^{K} \sum_{k^*=1}^{K^*} \frac{1}{t \cdot M^2} \log \frac{1}{t} \tag{6.49}$$

The variation of information maintains the same term :

$$VI(C, C^*) = H(C) + H(C^*) - 2I(C, C^*) \tag{6.50}$$

If clustering are totally random, $t = 1$ and $I(C, C^*) = 0$, we have the maximum variation of information as $VI(C, C^*) = 2H(C) = 2H(C^*)$. On the other side, if two clusterings are identical, $t = \frac{1}{M}$, we have $I(C, C^*) = H(C) = H(C^*)$, so the variation of information will be zero, $VI(C, C^*) = 0$. Indeed, the variation coefficient t is designed to reflect the

degree of the variation of information. We can set a linear function to estimate t:

$$t = \frac{M-1}{(H(C)+H(C^*))\cdot M}\cdot VI(C,C^*) + \frac{1}{M} \tag{6.51}$$

where the bounds $t = \frac{1}{M}$ for $VI(C,C^*) = 0$, and $t = 1$ for $VI(C,C^*) = H(C) + H(C^*)$ will be satisfied. Considering the clustering diversity, the Mirkin's metric will be :

$$\frac{K(C_i,C_k)}{2} = (C_{10}+C_{01}) = \widehat{C_{01}} + \widetilde{C_{01}} + \widehat{C_{10}} + \widetilde{C_{10}} \tag{6.52}$$

To elimintae the terms $\widetilde{C_{10}}$ and $\widetilde{C_{01}}$, one can calculate :

$$\widehat{C_{01}} + \widehat{C_{10}} = \frac{K(C_i,C_k)}{2} - 2*(tM-1)\cdot(C_{11}+\frac{N}{2}) \tag{6.53}$$

Developing the terms $\widehat{C_{01}}$ and $\widehat{C_{10}}$, we can get :

$$(N_{xx}+N_{yy})(N_{xy}+N_{yx}) = M\cdot(\frac{K(C_i,C_k)}{2} - 2*(tM-1)\cdot(C_{11}+\frac{N}{2})) \tag{6.54}$$

Gievn that $N_{xx}+N_{yy} = N_{11}+N_{00}$, and $N_{xy}+N_{yx} = N_{10}+N_{01}$, and the disagreement measure $DM_{i,k} = \frac{N_{10}+N_{01}}{N}$, we can write :

$$DM_{i,k}\cdot(N_{xx}+N_{yy}) = \frac{M}{N}\cdot(\frac{K(C_i,C_k)}{2} - 2*(tM-1)\cdot(C_{11}+\frac{N}{2})) \tag{6.55}$$

Again, here we need to solve the value of $N_{xx}+N_{yy}$. We can get the approximation from C_{11} and α:

$$N_{xx}+N_{yy} = ((2\cdot C_{11}+N)\cdot t\cdot M^2 + \frac{N^2}{2}\cdot\alpha)^{\frac{1}{2}} \tag{6.56}$$

Then the classifier diversity DM, can be approximated by Mirkin's metric taking into account the variation of information :

$$E(VI)_{i,k} \equiv \frac{\frac{M}{N} \cdot (\frac{K(C_i,C_k)}{2} - 2 * (t \cdot M - 1) \cdot (C_{11} + \frac{N}{2}))}{((2 \cdot C_{11} + N) \cdot t \cdot M^2 + \frac{N^2}{2} \cdot \alpha)^{\frac{1}{2}}} \tag{6.57}$$

Notice that if we set $t = \frac{1}{M}$, i.e., the situation of zero variation of information, we can get the eq. 6.34.

BIBLIOGRAPHY

[1] A. A. Afifi and S. P. Azen, "Statistical Analysis: A Computer Oriented Approach," Second Edition, New York: Academic Press, 1979

[2] Y. Amit and D. Geman, "Shape quantization and recognition with randomized trees", *Neural Computation*, no. 9, pp 545-1588, 1997

[3] N. Arica and F. T. Y. Vural, "A shape descriptor based on circular Hidden Markov Model," *In 15th Intl. Conf. on Pattern Recognition (ICPR00)*, 2000

[4] S. Bandyopadhyay and U. Maulik, "Non-parametric Genetic Clustering : Comparison of Validity Indices," *IEEE Transactions on Systems, Man and Cybernetics Part-C*, vol. 31, no. 1, pp. 120-125, 2001

[5] R. E. Banfield, L. O. Hall, K. W. Bowyer and W. P. Kegelmeyer, "A New Ensemble Diversity Measure Applied to Thinning Ensembles," International Workshop on Multiple Classifier Systems (MCS 2003), pp. 306 - 316, 2003

[6] Y. Bengio, "Markovian Models for Sequential Data", *Neural Computing Surveys*, vol. 2, pp. 129-162, 1999

[7] L. Breiman, "Random Forests", *Machine Learning*, no. 45, pp. 5-32, 2001

[8] A. Britto Jr., "A Two-Stage HMM-Based Method for Recognizing Handwritten Numeral Strings," *Ph.D. Thesis*, Pontifical Catholic University of Paraná, 2001

[9] A. Britto, R. Sabourin, F. Bortolozzi and C.Y. Suen, "Recognition of Handwritten Numeral Strings Using a Two-Stage Hmm-Based Method", *International Journal on Document Analysis and Recognition (IJDAR 5)*, no. 2-3, pp. 102-117, 2003

[10] G. Brown, J. Wyatt and P. Sun, "Between Two Extremes: Examining Decompositions of the Ensemble Objective Function," *International Workshop on Multiple Classifier Systems (MCS 2005)*, pp. 296-305, 2005

[11] G. Brown, J. Wyatt, R. Harris and X. Yao, "Diversity Creation Methods: A Survey and Categorisation," *International Journal of Information Fusion*, vol. 6, no. 1, pp. 5-20, 2005

[12] J. Cao, M. Ahmadi and M. Shridhar, "Recognition of handwritten numerals with multiple feature and multistage classifier", *Pattern Recognition*, vol. 28, no. 2, pp. 153-160, 1995

[13] K. Deb, "Multi-Objective Optimization using Evolutionary Algorithms", Wiley 2001, 2nd edition, 2002

[14] L. Didaci and G. Giacinto, "Dynamic Classifier Selection by Adaptive k-Nearest-Neighbourhood Rule," *International Workshop on Multiple Classifier Systems (MCS 2004)*, pp. 174-183, 2004

[15] L. Didaci, G. Giacinto, F. Roli and G. L. Marcialis, "A study on the performances of dynamic classifier selection based on local accuracy estimation," *Pattern Recognition*, vol. 38, no. 11, pp. 2188-2191, 2005

[16] Dietterich T. G., "Machine Learning for Sequential Data: A Review.", *In Structural, Syntactic, and Statistical Pattern Recognition*, Lecture Notes in Computer Science, vol. 2396, pp. 15-30, Springer-Verlag, 2002

[17] E. Dimitriadou, A. Weingessel, and K. Hornik, "Voting-merging: An ensemble method for clustering", *Artificial Neural Networks-ICANN 2001*, pp. 217-224, 2001

[18] P. Domingos, "A Unified Bias-Variance Decomposition and its Applications," *International Conference on Machine Learning (ICML 2000)*, pp. 231-238, 2000

[19] R. P. W. Duin, "Pattern Recognition Toolbox for Matlab 5.0+," available free at: ftp://ftp.ph.tn.tudelft.nl/pub/bob/prtools

[20] R. P. W. Duin, "The Combining Classifier: To Train or Not to Train?" *16th International Conference on Pattern Recognition (ICPR)*, vol. 2, pp. 20765, 2002

[21] A. E. Eiben, R. Hinterding, and Z. Michalewicz, "Parameter control in evolutionary algorithms", *In IEEE Transactions on Evolutionary Computation*, vol.3, no. 2, pp. 124-141, 1998

[22] Eppstein D., "Fast hierarchical clustering and other applications of dynamic closest pairs", *In Proceedings of the Ninth ACM-SIAM Symposium on Discrete Algorithms*, pp. 619–628, 1998

[23] X. Fern and C. Brodley, "Random projection for high dimensional data: A cluster ensemble approach", *In Proceedings of the 20th International Conference on Machine Learning (ICML)*, pp. 186-193, 2003

[24] B. Fischer and J. M. Buhmann, "Bagging for Path-Based Clustering", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 25, no. 11, pp. 1411-1415, 2003

[25] J. L. Fleiss, B. Levin, and M. C. Paik, "Statistical Methods for Rates and Proportions," Second Edition, New York: John Wiley & Sons, 2003

[26] A. L. N. Fred and A. K. Jain, "Data Clustering Using Evidence Accumulation," *In Proceedings of the International Conference on Pattern Recognition (ICPR 2004),* pp. 276-280, 2002

[27] B. E. Geman, S. and R. Dorsat, "Neural Networks and the Bias/Variance Dilemma," *Neural Computation*, no. 4, pp. 1-58, 1992

[28] G. Giacinto and F. Roli, "Methods for Dynamic Classifier Selection," *International Conference on Image Analysis and Processing (ICIAP 1999)*, pp. 659-664, 1999

[29] G. Giacinto and F. Roli, "Design of effective neural network ensembles for image classification purposes," *Image and Vision Computing*, vol. 19, no. 9-10, pp. 699-707, 2001

[30] G. Giacinto and F. Roli, "Dynamic Classifier Selection Based on Multiple Classifier Behaviour," *Pattern Recognition*, vol. 34, no. 9, pp. 179-181, 2001

[31] A. Grove and D. Schuurmans, "Boosting in the limit: Maximizing the Margin of Learned Ensembles," *In Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pp. 692-699, 1998

[32] S. Guenter and H. Bunke, "A new combination scheme for HMM-based classifiers and its application to handwriting recognition," *Proc. 16th Int. Conference on Pattern Recognition*, Vol. II, 332 - 337, 2002

[33] S. Guenter and H. Bunke, "Creation of classifier ensembles for handwritten word recognition using feature selection algorithms," *Proc. 8th Int. Workshop on Frontiers in Handwriting Recognition*, pp. 183 - 188 , 2002

[34] S. Guenter and H. Bunke, "Generating classifier ensembles from multiple prototypes and its application to handwriting recognition," *Proceedings of the 3rd International Workshop on Multiple Classifier Systems*, pp. 179 - 188, 2002

[35] S. Guenter and H. Bunke, "New Boosting Algorithms for Classification Problems with Large Number of Classes Applied to a Handwritten Word Recognition Task," *Proceedings of the 4th International Workshop on Multiple Classifier Systems*, pp. 326 - 335, 2003

[36] S. Guenter and H. Bunke (2003), "Ensembles of Classifiers for Handwritten Word Recognition," *International Journal of Document Analysis and Recognition*, Volume 5, Number 4, 224 - 232, 2003

[37] S. Guenter and H. Bunke, "Fast Feature Selection in an HMM-based Multiple Classifier System for Handwriting Recognition," *Pattern Recognition, Proceedings of the 25th DAGM Symposium*, pp. 289-296, 2003

[38] S. Guenter and H. Bunke, "Off-line Cursive Handwriting Recognition - On the Influence of Training Set and Vocabulary Size in Multiple Classifier Systems," *Proceedings of the 11th Conference of the International Graphonomics Society*, 2003

[39] S. Guenter and H. Bunke, "Ensembles of classifiers derived from multiple prototypes and their application to handwriting recognition," *International Workshop on Multiple Classifier Systems (MCS 2004)*, pp. 314-323, 2004

[40] S. Guenter and H. Bunke, "Optimization of weights in a multiple classifier handwritten word recognition system using a genetic algorithm," *Electronic Letters of Computer Vision and Image Analysis*, vol. 3, no. 1, pp. 25 - 44, 2004

[41] S. Guenter and H. Bunke, "Off-line cursive handwriting recognition using multiple classifier systems - on the influence of vocabulary, ensemble, and training set size," *Optics and Lasers in Engineering*, vol. 43, pp. 437-454, 2005

[42] V. Gunes, M. Ménard, P. Loonis and S. Petit-Renaud, "A Multiple Classifier System Using Ambiguity Rejection for Clustering-Classification Cooperation", *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 8, no.6, pp. 747-762, 2000

[43] V. Gunes, M. Ménard, P. Loonis and S. Petit-Renaud, "A Fuzzy Petri Net for Pattern Recognition: Application to Dynamic Classes", Knowledge and Information Systems, vol. 4, no. 1, pp. 112-128, 2002

[44] V. Gunes, M. Ménard, P. Loonis, S. Petit-Renaud, "Combination, Cooperation And Selection Of Classifiers: A State Of The Art", *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 17, no. 8, pp. 1303-1324, 2003

[45] Halkidi M., Batistakis Y. and Vazirgiannis M., "On Clustering Validation Techniques," *Journal of Intelligent Information Systems*, vol. 17 , no. 2-3, 2001

[46] Halkidi M., Batistakis Y. and Vazirgiannis M., "Clustering Validity Checking Methods: Part II," *SIGMOD Record*, vol. 31, no. 3, pp. 19-27, 2002

[47] L. K. Hansen and P. Salamon, "Neural Network Ensembles," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 993-1001, 1990

[48] L. K. Hansen, C. Liisberg and P. Salamon, "The error-reject tradeoff," *Open Systems and Information Dynamics*, (1997) vol. 4, pp. 159-184

[49] T. K. Ho, "The random space method for constructing decision forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 832-844, 1998

[50] Y. S. Huang and C.Y. Suen, "A method of combining multiple experts for the recognition of unconstrained handwritten numerals," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (1995) vol. 17, pp. 90-93

[51] X. Huang, A. Acero and H. Hon, "Spoken Language Processing - a guide to theory, algorithm, and system development," *Prentice Hall PTR*, 2001

[52] S. Jaeger, "Using Informational Confidence Values for Classifier Combination: An Experiment with Combined On-Line/Off-Line Japanese Character Recognition," *9th International Workshop on Frontiers in Handwriting Recognition,*, pp. 87-92, 2004

[53] G. James, "Variance and Bias for General Loss Functions," *Machine Learning*, vol. 51, no. 2, pp. 115-135, 2003.

[54] E. Johnson and H. Kargupta, "Collective, hierarchical clustering from distributed, heterogeneous data", *In Large-Scale Parallel KDD Systems*, pp. 221–244, 1999

[55] K. Kimura, S. Inoue T. Wakabayashi, S. Tsuruoka and Y. Miyake, "Handwritten Numeral Recognition using Autoassociative Neural Networks," *in Proceedings of the International Conference on Pattern Recognition (ICPR)*, vol. 1, pp. 166-171, 1998

[56] J. Kittler, M. Hatef, R. Duin, and J. Matas, "On Combining Classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 20, no. 3, pp. 226–239, 1998

[57] J. Kittler and F. M. Alkoot, "Relationship of Sum and Vote Fusion Strategies," *Multiple Classifier Systems (MCS)*, pp. 339-348, 2001

[58] A. H. R. Ko, R. Sabourin and A. Britto Jr,, "Combining Diversity and Classification Accuracy for Ensemble Selection in Random Subspaces" , *IEEE World Congress on Computational Intelligence (WCCI 2006) - International Joint Conference on Neural Networks*, 2006.

[59] A. H. R. Ko, R. Sabourin and A. Britto Jr,, "Evolving Ensemble of Classifiers in Random Subspace" , *Genetic and Evolutionary Computation Conference*, 2006.

[60] A. H. R. Ko, R. Sabourin, A. Britto Jr, A. and L. Oliveira, "Pairwise Fusion Matrix for Combining Classifiers", *Pattern Recognition* (Accepted for publication, January 2007).

[61] R. Kohavi, and 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, 1996

[62] A. Krogh and J. Vedelsby, "Neural Network Ensembles, Cross Validation, and Active Learning," *Advances in Neural Information Processing Systems*, vol. 7, pp. 231–238, 1995

[63] L. I. Kuncheva, M. Skurichina, and R. P. W. Duin, "An Experimental Study on Diversity for Bagging and Boosting with Linear Classifiers," *International Journal of Information Fusion*, vol. 3, no. 2, pp. 245-258, 2002

[64] L. I. Kuncheva, "A Theoretical Study on Six Classifier Fusion Strategies," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 281-286, 2002

[65] L. I.Kuncheva, "Switching between selection and fusion in combining classifiers: An experiment",*IEEE Transactions On Systems Man And Cybernetics, Part B*, vol. 32, no. 2, pp. 146-156, 2002

[66] L. I. Kuncheva and C. J. Whitaker, "Measures of Diversity in Classifier Ensembles and their Relationship with the Ensemble Accuracy," *Machine Learning*, vol. 51, no. 2, pp. 181-207, 2003

[67] L. I. Kuncheva and S.T. Hadjitodorov, "Using Diversity in Cluster Ensembles", *In Proceedings of IEEE International Conference on Systems, Man and Cybernetics,* Part B, pp. 1214-1219, 2004

[68] L. I. Kuncheva, "Classifier Ensembles for Changing Environments", *International Workshop on Multiple Classifier Systems (MCS)*, vol. 3077, pp. 1-15, 2004

[69] L. I. Kuncheva, "Combining Pattern Classifiers. Methods and Algorithms", Wiley, 2004.

[70] E. B. Mansilla and T. K. Ho, "On Classifier Domains of Competence", *17th International Conference on Pattern Recognition (ICPR'04)*, vol. 1, pp. 136-139, 2004

[71] U. Maulik and S. Bandyopadhyay, "Performance Evaluation of Some Clustering Algorithms and Validity Indices," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 24,

no. 12, pp. 1650-1654, 2002

[72] M. Meila, "Comparing clusterings", Technical Report 418, *UW Statistics Department,* 2002

[73] P. Melville and R. J. Mooney, "Creating Diversity in Ensembles Using Artificial Data," *Information Fusion*, vol. 6, no. 1, pp. 99-111, 2005

[74] J. Milgram, M. Cheriet and R. Sabourin, "Estimating Accurate Multi-class Probabilities with Support Vector Machines," *International Joint Conference on Neural Networks 2005 (IJCNN)*, pp. 1906-1911, 2005

[75] L. S. Oliveira, R. Sabourin, F. Bortolozzi and C. Y. Suen, "Automatic Recognition of Handwritten Numerical Strings: A Recognition and Verification Strategy", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 24, no. 11, pp. 1438-1454, 2002

[76] L. S. Oliveira, M. Morita, R. Sabourin and F. Bortolozzi, "Multi-Objective Genetic Algorithms to Create Ensemble of Classifiers," *In Proceedings of the 3rd International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005)*, pp 592-606, 2005

[77] D. Opitz and R. Maclin, "Popular Ensemble Methods: An Empirical Study,"*Journal of Artificial Intelligence Research*, vol. 11, pp. 169-198, 1999

[78] M. K. Pakhira, S. Bandyopadhyay and U. Maulik, "Validity index for crisp and fuzzy clusters," *Pattern Recognition*, vol. 37, No. 3, pp. 487-501, 2004

[79] B. Park and H. Kargupta, "Distributed Data Mining: Algorithms, Systems, and Applications", *Data Mining Handbook*, Lawrence Erlbaum Associates, 2003

[80] D. Partridge and W. Krzanowski, "Software diversity: practical statistics for its measurement and exploitation," *Information and Software Technology*, vol. 39, pp. 707-717, 1997

[81] E. Pekalska, M. Skurichina and R. P. W. Duin, "Combining Dissimilarity-Based One-Class Classifiers," *International Workshop on Multiple Classifier Systems (MCS 2004)*, pp. 122-133, 2004

[82] Y. Qian and C. Y. Suen, "Clustering Combination Method", *In Proceedings of the International Conference on Pattern Recognition (ICPR 2000),* pp. 2732-2735, 2000

[83] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proceedings of the IEEE*, 77(2):257 – 286, 1989

[84] L. R. Rabiner and B. H. Juang, "Fundamentals of Speech Recognition," *Prentice-Hall*, 1993

[85] P. Radtke, R. Sabourin and T. Wong, "Intelligent Feature Extraction for Ensemble of Classifiers", *8th International Conference on Document Analysis and Recognition (ICDAR)*, pp. 866-870, 2005

[86] P. Radtke, T. Wong and R. Sabourin, "An Evaluation of Over-Fit Control Strategies for Multi-Objective Evolutionary Optimization", *IEEE World Congress on Computational Intelligence (WCCI) - International Joint Conference on Neural Networks (IJCNN)*, 2006.

[87] S. Raudys, "Experts' boasting in trainable fusion rules," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1178 - 1182, 2003

[88] D. Ruta and B. Gabrys, "Analysis of the Correlation between Majority Voting Error and the Diversity Measures in Multiple Classifier Systems," *In Proceedings of the 4th International Symposium on Soft Computing*, 2001

[89] D. Ruta and B. Gabrys, "Classifier Selection for Majority Voting," *International Journal of Information Fusion*, pp. 63-81, 2005

[90] R. E. Schapire, Y. Freund, P. Bartlett and W. S. Lee, "Boosting the Margin: A New Explanation for the Effectiveness of Voting Methods," *Annals of Statistics*, vol. 26, no. 5, pp. 1651–1686, 1998

[91] J. Seo and B. Shneiderman, "Interactively Exploring Hierarchical Clustering Results," *IEEE Computer*, vol. 35, no. 7, pp. 80-86, 2002

[92] C. A. Shipp and L. I. Kuncheva, "Relationships Between Combination Methods and Measures of Diversity in Combining Classifiers," *International Journal of Information Fusion*, vol. 3, no. 2, pp. 135 - 148, 2002

[93] M. Skurichina, L. I. Kuncheva and R. P. W. Duin, "Bagging and Boosting for the Nearest Mean Classifier: Effects of Sample Size on Diversity and Accuracy," *International Workshop on Multiple Classifier Systems (MCS 2002)*, pp. 62-71, 2002

[94] P. Smyth, D. Heckerman and M. I. Jordan, "Probabilistic independence networks for hidden Markov probability models", *Neural Computation*, vol. 9, pp. 227–269, 1997

[95] A. Strehl and J. Ghosh, "Cluster Ensembles - A Knowledge Reuse Framework for Combining Multiple Partitions," *Journal of Machine Learning Research,* no. 3, pp. 583-617, 2002

[96] D. M. J. Tax, M. Van Breukelen, R. P. W. Duin, and J. Kittler, "Combining Multiple Classifiers by Averaging or by Multiplying," *Pattern Recognition*, vol. 33, no. 9, pp.1475-1485, 2000

[97] A. Topchy, A. Jain, and W. Punch, "Combining multiple weak clusterings", *In Proceedings of IEEE International Conference on Data Mining (ICDM 03)*, 2003

[98] A. Topchy, A. K. Jain, W. Punch, "Clustering Ensembles: Models of Consensus and Weak Partitions", *To appear in IEEE Transactions on Pattern Analysis and Machine Intelligence*

[99] G. Tremblay, R. Sabourin, and 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, 2004

[100] K. Turner and J. Ghosh, "Error Correlation and Error Reduction in Ensemble Classifiers," *Connection Science*, vol. 8, no. 3-4, pp. 385-404, 2006

[101] N. Ueda and R. Nakano, "Generalization error of ensemble estimators," *In Proceedings of International Conference on Neural Networks (ICNN 1996)*, pp. 90-95, 1996

[102] X. Wang, "Durationally constrained training of HMM without explicit state durational", *Proceedings of the Institute of Phonetic Sciences 18*, pp. 111-130, 1994

[103] G. I. Webb and Z. Zheng, "Multistrategy Ensemble Learning: Reducing Error by Combining Ensemble Learning Techniques," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 8, pp. 980-991, 2004

[104] K. D. Wernecke, "A coupling procedure for discrimination of mixed data," *Biometrics*, vol. 48, pp. 97-506, 1992

[105] D. Whitley, "Functions as Permutations: Regarding No Free Lunch, Walsh Analysis and Summary Statistics". *Parallel Problem Solving from Nature (PPSN 2000)*, pp. 169-178, 2000

[106] D. H. Wolpert and W.G. Macready, "No free lunch theorems for search", *IEEE Transactions on Evolutionary Computation*, 1997

[107] K. Woods, W. P. Kegelmeyer Jr, and K. Bowyer, "Combination of multiple classifiers using local accuracy estimates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 4, pp. 405–410, 1997

[108] X. L. Xie and G. Beni, "A validity measure for fuzzy clustering",*IEEE Transactions of Pattern Analysis and Machine Intelligence*, pp. 841-847, 1991

[109] 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, pp. 418-435, 1992

[110] N. Yamaguchi and N. Ishii, "Combining classifiers in error correcting output coding," *Systems and Computers in Japan*, vol. 35, no. 4, pp. 9-18, 2004

[111] H. Zouari, L. Heutte, Y. Lecourtier and A. Alimi, "Building Diverse Classifier Outputs to Evaluate the Behavior of Combination Methods: the Case of Two Classifiers," *International Workshop on Multiple Classifier Systems (MCS 2004)*, pp. 273-282, 2004