İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

SUPERVISED AND SEMI-SUPERVISED LEARNING USING INFORMATIVE FEATURE SUBSPACES

Ph.D. Thesis by
Yusuf YASLAN

Department : Computer Engineering

Programme : Computer Engineering

DECEMBER 2010

İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

SUPERVISED AND SEMI-SUPERVISED LEARNING USING INFORMATIVE FEATURE SUBSPACES

Ph.D. Thesis by
Yusuf YASLAN
(504042506)

Date of submission : **19 October 2010**
Date of defence examination : **02 December 2010**

Supervisor (Chairman) : **Assoc. Prof. Dr. Zehra ÇATALTEPE (ITU)**
Members of the Examining Committee : **Prof. Dr. Muhittin GÖKMEN (ITU)**
Assoc. Prof. Dr. Berrin YANIKOĞLU (SU)
Assoc. Prof. Dr. Tunga GÜNGÖR (BU)
Assist. Prof. Dr. Şule GÜNDÜZ – ÖĞÜDÜCÜ (ITU)

DECEMBER 2010

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİ İÇEREN ÖZNİTELİK ALT UZAYLARI İLE EĞİTMENLİ VE YARI EĞİTMENLİ ÖĞRENME

DOKTORA TEZİ
Yusuf YASLAN
(504042506)

Tezin Enstitüye Verildiği Tarih : 19 Ekim 2010
Tezin Savunulduğu Tarih : 02 Aralık 2010

Tez Danışmanı : Doç. Dr. Zehra ÇATALTEPE (İTÜ)
Diğer Jüri Üyeleri : Prof. Dr. Muhittin GÖKMEN (İTÜ)
Doç. Dr. Berrin YANIKOĞLU (SÜ)
Doç. Dr. Tunga GÜNGÖR (BÜ)
Yrd. Doç. Dr. Şule GÜNDÜZ – ÖĞÜDÜCÜ (İTÜ)

ARALIK 2010

FOREWORD

First of all I would like to thank to my supervisor Dr. Zehra Çataltepe for her support and effort during my academic research. I believe that without her enthusiasm on research, this thesis would never be finished. Thanks also go to her for proofreading the thesis book that significantly improved the quality of the thesis.

Next I would like to thank to the progress committee members Dr. Berrin Yanıkoğlu and Dr. Şule Gündüz-Öğüdücü for their valuable time and comments in the periodical meetings.

During the last couple years Kenan Kule has been my roommate. He didn't hesitate to help me whenever I needed. I would like to thank him for his help. I also thank to members and research assistants of the Computer Engineering Department at Istanbul Technical University: Berk Canberk, Tolga Ovatman, Burak Kantarcı, Melike Erol-Kantarcı, Çağatay Talay, Nagehan İlhan, Figen Şentürk, Gülnur Selda Kuruoğlu, Aycan Atak and Mustafa Ersen. Besides during my assistantship at the Department, Tacettin Ayar and Dr. A. Cüneyd Tantuğ were always with me with their friendships. I'd like to also thank to them.

Last I would like to thank my mother, my father and my brother. I could have never achieved my goals without their valuable support.

December 2010

Yusuf Yaslan
Computer Engineer, M.Sc

TABLE OF CONTENTS

Page

ABBREVIATIONS

RAS	**:** Random Subspaces
Rel-RAS	**:** Relevant Random Subspaces
mRMR-RAS	**:** minimum Redundancy Maximum Relevance Random Subspaces
RASCO	**:** Random Subspaces for Co-training
Rel-RASCO	**:** Relevant Random Subspaces for Co-training
mRMR-RASCO	**:** minimum Redundancy Maximum Relevance Random Subspaces for Co-training
KNN	**:** K - Nearest Neighbour
LDC	**:** Linear Discriminant Classifier
SVM	**:** Support Vector Machines
RM	**:** Recursively More
KW	**:** Kohavi - Wolpert
LOD	**:** Low Order Diversity
ITS	**:** Information Theoretic Score
ITA	**:** Information Theoretic Accuracy
ITD	**:** Information Theoretic Diversity
MID	**:** Mutual Information Distance
MIQ	**:** Mutual Information Quotient

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OF SYMBOLES

X	:	Training dataset
X_i	:	ith training instance
x_{ij}	:	jth feature of the ith training instance
F_j	:	jth feature vector of dataset X
d	:	Instance dimensionality
n	:	Number of training instance
C_k	:	kth classifier in an ensemble
C_E	:	Ensemble classifier
S_k	:	kth feature subspace
l	:	Training dataset labels
K	:	Number of classifiers in an ensemble
H	:	Entropy
I	:	Mutual information
d_{kj}	:	Decision of the kth classifier on class j in an ensemble
$\hat{X}^k$	:	Training dataset obtained using subspace, S_k, and training dataset
m	:	Number of selected features in a subspace
V	:	Relevance betwen features and class labels
W	:	Redundancy betwen features
S^m	:	Subspace with m features
e^m	:	Classification error obtained by the classifier trained using subspace S^m
$\bar{e}$	:	Mean individual classification error
μ	:	Portion of the labeled training dataset used to train a classifier
L	:	Labeled training dataset in semi-supervised learning
L_i	:	ith Labeled training instance in semi-supervised learning
U	:	Unlabeled training dataset in semi-supervised learning
U_i	:	ith Unlabeled training instance in semi-supervised learning
r	:	Number of unlabeled training instance

SUPERVISED AND SEMI-SUPERVISED LEARNING USING INFORMATIVE FEATURE SUBSPACES

SUMMARY

Ensemble of classifiers aims to produce accurate recognition results by training several classifiers and combining their outputs. It may also benefit from diversity of classifiers used. However, for high dimensional data choosing subspaces randomly, as in RAS (Random Subspaces) algorithm, may produce diverse but inaccurate classifiers. On the other hand, in many different fields ranging from web mining to speech recognition, unlabeled data have become abundant and there have been many efforts to benefit from unlabeled data. Co-training is one of the successful semi-supervised learning algorithm that trains two classifiers on different feature views and uses the unlabeled data in an iterative way for re-training these classifiers. Recently, a multi-view Co-training algorithm, RASCO (Random Subspace Method for Co-training), which obtains different feature splits using random subspace method was proposed and shown to result in smaller errors than the traditional Co-training. However RASCO has the possibility to use diverse but inaccurate classifiers during Co-training on account of selecting subspaces randomly.

In this thesis we propose to obtain subspaces for classifier ensembles by means of drawing features with probabilities which are generated in an intelligent way. Two feature subspace selection methods for ensemble of classifiers are proposed and applied on different supervised and semi-supervised learning scenarios.

The first algorithm is the relevant random subspace method which produces the relevant random subspaces using the relevance values obtained by mutual information between features and class labels. This method is used in Rel-RAS and Rel-RASCO algorithms where Rel-RAS is the relevant random subspace method for supervised learning and Rel-RASCO is the relevant random subspace method for Co-training.

The second algorithm is the minimum redundancy and maximum relevance feature subspace selection method that modifies the mRMR (Minimum Redundancy Maximum Relevance) feature selection algorithm to produce random feature subspaces that are relevant and non-redundant. The second method is used in the mRMR-RAS and mRMR-RASCO algorithms where mRMR-RAS is the minimum redundancy maximum relevance random subspace method for supervised learning and mRMR-RASCO is the minimum redundancy maximum relevance random subspace method for Co-training.

Experimental results on five real and synthetic datasets with K-Nearest Neighbour (KNN), Linear Discriminant (LDC), decision tree and Support Vector Machines (SVM) classifiers show that the proposed algorithms generally outperform supervised algorithm, RAS and semi-supervised algorithms, RASCO and Co-training (at the beginning and end of semi-supervised algorithms) based on the accuracy achieved. On the other hand diversity of the classifiers in ensemble is suspected to affect the ensemble accuracy and there have been many works investigating the relationship between classifier diversity and ensemble accuracy. The proposed algorithms are also evaluated in terms of classifier diversity using Kohavi Wolpert (KW) Variance. We have shown that the classifier diversity with Rel-RAS, mRMR-RAS and Rel-RASCO, mRMR-RASCO are slightly less than the classifier diversity with RAS and RASCO respectively. This result is due to the fact that classifiers combined in Rel-RAS, mRMR-RAS and Rel-RASCO, mRMR-RASCO algorithms more agree on class labels of test data than RAS and RASCO algorithms respectively. In the experiments algorithms are also evaluated using approximately Recursively More characteristic (RM characteristic) definition of feature subspaces. It is shown that the subspaces generated using the proposed algorithms are more RM characteristic than the subspaces generated in RAS and RASCO in terms of mean accuracies of the individual classifiers. Besides, t-tests of the test results are given.

In addition to KW-Variance diversity measure, information theory based low order diversity (LOD) and information theoretic scores (ITS) of the classifier ensembles are analyzed. In our experiments it is found that information theory based low order diversity has a similar tendency with KW-variance. On the other hand we found out that ensemble accuracy of the algorithms can be explained with information theoretic score (ITS) and under the same conditions (same number of classifiers in the ensembles, same training set etc.), higher the ITS higher the classification accuracy.

BİLGİ İÇEREN ÖZNİTELİK ALT UZAYLARI İLE EĞİTMENLİ VE YARI EĞİTMENLİ ÖĞRENME

ÖZET

Sınıflandırıcı toplulukları (classifier ensembles) birçok sınıflandırıcıyı eğitip, bu sınıflandırıcıların kararlarını birleştirerek, sınıflandırma başarımını arttırmayı hedeflemektedir. Aynı zamanda sınıflandırıcıların çeşitliliği (diversity) sınıflandırma başarımın arttırılmasına yarar sağlayabilmektedir. Fakat yüksek boyutlu öznitelik vektörlerinin bulunduğu verilerde öznitelik altuzaylarını (subspace), RAS (Random Subpaces) algoritmasında olduğu gibi rastgele seçmek sınıflandırıcı çeşitliliğini sağlamakta fakat düşük başarımlı sınıflandırıcılar oluşturabilmektedir. Öte yandan, web madenciliğinden ses tanımaya kadar birçok alanda çok miktarda etiketsiz veriye erişilebilmekte ve bu etiketsiz verilerden yararlanmak için yoğun çalışmalar yapılmaktadır. Birlikte Öğrenme (Co-training) algoritması, farklı iki öznitelik görünümünde sınıflandırıcı eğiterek, özyineli olarak etiketsiz veriyi etiketleyen ve bu yeni etiketlenmiş verileri de kullanarak sınıflandırıcıları yeniden eğiten başarılı bir yarı-eğiticili öğrenme algoritmasıdır. Son dönemde, rastgele seçilmiş öznitelik altuzaylarını kullanan RASCO (Random Subspace Method for Co-training) algoritması önerilmiş ve geleneksel Birlikte Öğrenme algoritmasından daha düşük hataya sahip olduğu gösterilmiştir. Bununla beraber RASCO algoritması öznitelik alt uzaylarını rastgele seçtiği için sınıflandırıcı çeşitliliği arttırmakta fakat başarım oranı düşük sınıflandırıcılara sahip olabilme olasılığı bulunmaktadır.

Bu tez çalışması kapsamında sınıflandırıcı toplulukları için öznitelik altuzaylarının, daha akıllı bir şekilde elde edilmiş olasılık değerleri kullanılarak seçilmesi önerilmiştir. Sınıflandırıcı toplulukları için iki öznitelik alt uzay seçim yöntemi önerilmiş; eğiticili ve yarı-eğiticili farklı öğrenme yöntemlerine uygulanmıştır.

Tez kapsamında önerilen ilk yöntem; öznitelik altuzaylarını öznitelikler ve sınıf etiketleri arasındaki karşılıklı bilgi miktarını (mutual information) kullanarak oluşturan ilişkili rastgele altuzaylar (relevant random subspaces) yöntemidir. Bu yöntem, eğiticili öğrenme için ilişkili ve rastgele alt uzay metodu kullanan, Rel-RAS, ve yarı-eğiticili Birlikte Öğrenme için ilişkili ve rastgele alt uzay metodu kullanan, Rel-RASCO, algoritmalarında öznitelik altuzaylarının seçimi için kullanılmıştır.

İkinci yöntem; mRMR (Minimum Redundancy Maximum Relevance) öznitelik seçme algoritması üzerinde değişiklik yapılarak elde edilen öznitelik altuzaylarını öznitelikler ve sınıf etiketleri arasındaki karşılıklı bilgi miktarını ve özniteliklerin kendi aralarındaki karşılıklı bilgi miktarını dikkate alarak oluşturan en düşük artıklık ve en yüksek ilişkili rastgele altuzaylar (minimum Redundancy Maximum Relevance random subspaces) yöntemidir. Bu ikinci yöntem, eğiticili öğrenme için en düşük artıklık ve en yüksek ilişkili rastgele alt uzay metodu kullanan, mRMR-RAS, ve yarı-eğiticili Birlikte Öğrenme için ilişkili ve artıksız rastgele alt uzay metodu kullanan, mRMR-RASCO, algoritmalarında öznitelik alt uzaylarının seçimi için kullanılmıştır.

Beş adet gerçek ve sentetik veri kümeleri üzerinde K-En Yakın Komşu (K-Nearest Neighbour, KNN), Doğrusal Ayırtaç (Linear Discriminant, LDC), Karar Ağacı (decision tree) ve Destek Vektör Makinaları (Support Vector Machines) sınıflandırıcıları ile elde edilen sonuçlar, önerilen algoritmaların eğiticili algoritmalarda RAS'tan ve yarı-eğiticili algoritmalarda RASCO ve Birlikte Öğrenme (yarı-eğiticili öğrenme algoritmalarının başlangıcındaki ve algoritma sonundaki başarımlarından) algoritmalarından elde edilen sınıflandırma başarımı açısından daha başarılı olduklarını göstermektedir.

Öte yandan sınıflandırıcı topluluklarında, sınıflandırıcı çeşitliliğinin sınıflandırma başarımına etkisi bulunduğu düşünülmekte ve sınıflandırıcı çeşitliliği ile sınıflandırma başarımı arasındaki ilişki ile ilgili birçok çalışma bulunmaktadır. Tez çalışması kapsamında önerilen algoritmaların çeşitliliği Kohavi Wolpert (KW) Varyans'ı ile incelenmiştir. Test sonuçlarından Rel-RAS, mRMR-RAS ve Rel-RASCO, mRMR-RASCO algoritmalarının sınıflandırıcı çeşitliliği RAS ve RASCO algoritmalarının sınıflandırıcı çeşitliliğinden çok az düşük olduğu görülmüştür. Bu sonuç Rel-RAS, mRMR-RAS ve Rel-RASCO, mRMR-RASCO algoritmaları ile birleştirilen sınıflandırıcıların test kümelerindeki sınıf etiketleri üzerinde RAS ve RASCO algoritmalarına göre elde edilen sonuçlardan daha fazla uyuşmalarından kaynaklanmaktadır. Algoritmalar öznitelik alt uzaylarının, yaklaşık özyineli olarak daha fazla karakteristik (approximately Recursively More characteristic (RM characteristic)) olma tanımı kullanılarak da incelenmiştir. Önerilen algoritmalarla elde edilen öznitelik alt uzaylarının RAS ve RASCO algoritmalarına göre sınıflandırıcıların bireysel başarımlarının ortalamaları dikkate alındığında daha RM-karateristik olduğu gösterilmiştir. Buna ek olarak test sonuçları üzerinde t-test sonuçları verilmiştir.

Sınıflandırıcı çeşitliliklerinin KW-Varyans ölçümlerine ek olarak, bilgi kuramı (information theoretic) tabanlı düşük düzeyli çeşitlilik ölçütü (low order diversity) ve sınıflandırıcı topluluklarının bilgi kuramı sayısı (information theoretic scores-ITS) incelenmiştir. Testlerden elde edilen sonuçlarda bilgi kuramı tabanlı düşük düzeyli çeşitlilik ölçütünün KW-varyansı ile benzer bir davranış gösterdiği görülmüştür. Öte yandan bilgi kuramı sayısı (ITS) ve sınıflandırıcı toplulukları arasında doğrudan bir ilişki görülmüştür. Aynı koşullar altında (toplulukta bulunan eşit sayıda sınıflandırıcı, aynı eğitim kümesi vs.) ITS değerinin yükselmesi sınıflandırma başarımının yükselmesine karşı geldiği görülmüştür.

1. INTRODUCTION

The easy access of data in many fields produced pattern recognition problems with high dimensional feature spaces. Generally one can either train a single classifier with/without feature selection/extraction or train multiple classifiers on feature subspaces and combine them [2]. However, when the number of instances are small compared to the number of features, we may face small sample size problem (curse of dimensionality) [3]. Feature selection methods have been shown to increase classification performance while defying the curse of dimensionality [4, 5]. They estimate the feature quality using a measure such as information gain, Gini index or chi-square test [6]. However they usually do not consider redundancy of the selected features. Recently Peng et. al. proposed a powerful method called, minimum Redundancy and Maximum Relevance (mRMR) [7] feature selection algorithm that gives an ordering of the features based on their relevance to the class label and redundancy between features. The mRMR method aims to select the next feature as uncorrelated as possible with the current subspace of selected features.

In addition to high dimensional feature spaces, it is also common to face unlabeled data in many fields ranging from bioinformatics to web mining. Semi-supervised learning methods have gained great importance with the availability of unlabeled data and they stand between supervised and unsupervised learning. Based on the availability of one or more sets of input features, with or without labels and ability to query some inputs, different combinations of datasets and hence learning algorithms to learn them can be considered. Scenarios of different input/output feature availability given in [1] are shown in Figure 1.1.

The learning methods that are applicable to each of the scenarios in Figure 1.1 is described as follows [1]:

a) Unsupervised learning: Without label information, each object is represented by one set of features. There isn't any information about data labels. Unsupervised learning or clustering aims to find the similar structure among the objects and cluster

the similar objects into same groups [8,9]. This is the scenario shown in Figure 1.1(a). An illustration of clustering on two dimensional feature space is given in Figure 1.2(a) where objects are separated into two clusters. Note that, defining a similarity measure for clustering is one of the crucial steps in unsupervised learning and depending on the similarity measure, cost function and input patterns different clustering algorithms lead to different results.

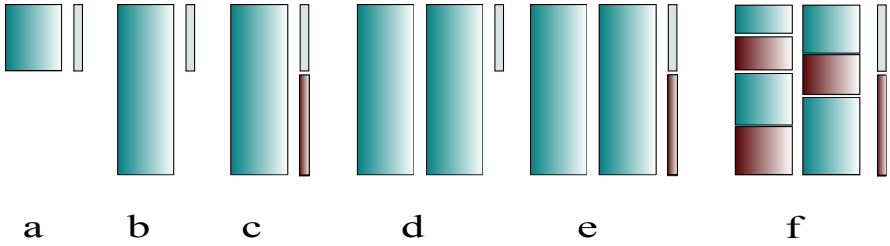
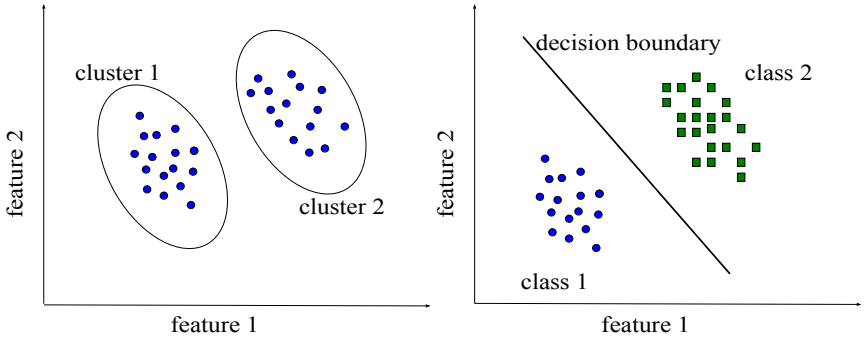


Figure 1.1: Scenarios of different input/output feature availability (Extended from [1]). Rows correspond to objects/instances. Wide boxes are feature matrices, narrow boxes correspond to labels. Available data are represented in blue, missing data that can be queried by a learning algorithm are represented in purple color.

b) Supervised learning: Each object is represented by one set of features and one label. In supervised learning a set of training data is available and classifier/regressor is designed by using this a priori information [8–10]. This is the scenario shown in Figure 1.1(b). If the target label of the problem is continuous then the supervised learning problem is called **regression**. Otherwise, if the labels have discrete values then the supervised learning problem is called **classification**. The aim is to find a mapping from input features to output labels and the mapping needs to minimize an appropriate error function on training data. An illustration of classification in two dimensional feature space is given in Figure 1.2(b), where objects are classified into two different classes.

c) Semi-Supervised learning: Some object labels are available however the other parts' labels are missing and not available. Learning in this case, using both labeled and unlabeled data, is known as semi-supervised learning [11] (Figure 1.1(c)). Detailed description of Semi-Supervised learning methods is given in Chapter 4. Transductive learning is a special case of semi-supervised learning where the unlabeled instances are actually test instances [12]. There are many extensions [13] to semi-supervised learning and some of them based on active learning and Co-training are defined below. Co-training is detailed Section 4.2.



(a) Unsupervised learning (clustering) (b) Supervised learning (classification)

Figure 1.2: a) Unsupervised and b) supervised learning algorithms illustration on two dimensional feature space.

d) Active learning of labels: Some object labels are available however the other parts' labels are missing but can be acquired [14] (Figure 1.1(d)). Any algorithm for the labeled and unlabeled data can be also used for active learning by selecting random points for collecting the labeled data and selecting the remaining part as unlabeled. However the aim of active learning is to outperform such algorithms [10]. Also this approach assumes the availability of an "oracle" that can label the unlabeled data points in the presence of a question. The intention of active learning is to select the most informative unlabeled examples by asking minimum number of questions to the "oracle" [13].

e) Co-training: A number of feature sets are available, but some of the objects have missing labels that cannot be acquired (Figure 1.1e). Co-training algorithm [15] is an iterative algorithm, that trains different classifiers on different feature views and updates these views by labeling the unlabeled data and adding them to the training set during the iterations. Detailed description of Co-training algorithm is given in Chapter 4.

f) Active learning of labels with co-training: A number of feature sets are available, but some of the objects have missing labels that can be acquired by asking questions to an "oracle" (Figure 1.1f).

g) Active learning of features and labels: Some of the features have missing values and some of the objects have not been labeled. However active learning of features and labels can be achieved by asking an "oracle" (Figure 1.1g). More detailed description can be found in [1].

In some applications data samples obtained from various sources may be represented in different multiple ways (or views), for example, web pages can be represented using text, image and video information. The learning problems summarized in Figure 1.1 e), f) and g) work on two different feature views. When multiple feature views are available, instead of training one classifier on the concatenated feature views, using multiple classifier systems can be useful [16]. On the other hand, on high dimensional feature spaces one can obtain different feature views artificially as in Random Subspaces Method (RAS) [17]. The RAS method selects the feature subspaces randomly for classifier ensembles and are shown to perform well using different classifiers such as K-Nearest Neighbors (KNN) [18], decision trees [17], pseudo Fisher linear classifier [19]. However, RAS method may not perform well when there are irrelevant or redundant features.

1.1 Contributions of the Thesis

The main contributions of this thesis are on relevant and non redundant random subspaces for supervised and semi-supervised learning.

1) Relevant Random Subspaces (Rel-RAS) and minimum Redundancy Maximum Relevance Random Subspaces (mRMR-RAS) Algorithms:

Feature selection and classifier ensembles, on both supervised and semi-supervised learning, are crucial problems in pattern recognition. On the other hand, selecting the relevant features and eliminating the redundant ones is a big issue in feature selection [20]. It has been found that selecting the most relevant features may not result in good classification performance [4]. Therefore redundancy among features is also studied [7, 21]. However training one classifier alone on the selected feature subset may not always give good classification accuracy. Besides, depending on the pattern recognition problem one can obtain many feature views and use classifier ensembles.

One of the main contributions of this thesis is made on classifier ensembles. Ensemble learning algorithms may benefit from diversity of classifiers used. However, for high dimensional data choosing subspaces randomly, as in Random Subspaces (RAS) algorithm, may produce diverse but inaccurate classifiers. On the other hand, if there are many irrelevant features and redundancy, RAS may produce subspaces of features

that are not suitable for good classification (See Section 3.2 for RAS algorithm). In order to eliminate these problems, we introduce two subspace selection methods for ensemble of classifiers. The first algorithm is the relevant random subspace method which produces relevant random subspaces using the relevance scores of the features obtained by mutual information between features and class labels. The second algorithm is the relevant and non-redundant random subspace selection that modifies the mRMR feature selection algorithm to produce random feature subspaces that are relevant and non-redundant. These feature subspace selection methods are used in Rel-RAS (Relevant Random Subspaces) and mRMR-RAS (minimum Redundancy Maximum Relevance Random Subspaces) supervised algorithms respectively during subspace selection.

2) Relevant Random Subspaces for Co-training (Rel-RASCO) and minimum Redundancy Maximum Relevance Random Subspaces for Co-training (mRMR-RASCO) Algorithms:

The use of unlabeled data is a challenging problem. Many algorithms have been proposed to benefit from unlabeled data [12]. It has been shown that using ensemble of classifiers increases the classification performance on semi-supervised learning as well [22, 23]. Co-training is a type of semi-supervised learning that uses unlabeled data on two different feature views. Previously we proposed a classifier combination method for Co-training algorithm [24]. The Co-training algorithm is extended for multiple feature views by Wang et. al. [23] and named as Random Subspace Method for Co-training (RASCO).

The next contribution of the thesis is made on semi-supervised ensemble learning by using the proposed feature subspace selection algorithms. Relevant Random Subspaces for Co-training (Rel-RASCO) and Relevant and Non-Redundant Random Subspaces for Co-training (mRMR-RASCO) algorithms are proposed for semi-supervised learning and they outperform the RASCO [23] algorithm which uses random subspaces for Co-training. The proposed algorithms are compared using the RM-characteristics of feature subspaces on both supervised and semi-supervised learning cases. It is shown that the proposed algorithms are more RM-characteristics in the mean of the classification accuracy.

3) Diversity Analysis of the Classifier Ensembles:

The last contribution of the thesis is on diversity analysis of the classifier ensembles. The analysis of the algorithms are based on KW-Variance diversity measure [2], information theoretic based low order diversity (LOD) [25] and information theoretic scores (ITS) [26]. It is shown that the increase in the ensemble classifier accuracy can be explained with the information theoretic score. On the other hand KW-Variance diversity measure and information theoretic based low order diversity have similar behavior and the performance increase in the ensemble cannot be explained directly with these measures.

The rest of the thesis chapters are organized as follows:

• In Chapter 2, first classifier ensemble methods, namely Bagging, Boosting, Stacked Generalization, Mixture of Experts, Input Decimated Ensembles are summarized and combination methods are given. Next measures of diversity for classifier ensembles and mutual information based classifier ensemble analysis are given.

• Chapter 3 includes the Random Subspaces (RAS), proposed Relevant Random Subspaces (Rel-RAS) and minimum Redundancy and Maximum Relevance Random Subspace (mRMR-RAS) algorithms. Next theoretical analysis of the algorithms is presented. Experimental results on supervised learning are given in terms of accuracy and diversity.

• In Chapter 4, first semi-supervised learning algorithms are summarized. Then RASCO, Rel-RASCO and mRMR-RASCO algorithms are presented and experimental results are given.

• Chapter 5 concludes the thesis by discussing the outcomes and the possible future directions for the work.

2. CLASSIFIER ENSEMBLES AND DIVERSITY

2.1 Classifier Ensembles

During the last decade computational intelligence community started to benefit from different experts to reduce the probability of making mistake [27,28]. Kittler states that in statistical pattern recognition most of the progress has been on modeling probability density function, feature selection and classification context and describes the classifier ensembles as one of the exciting directions [29]. Besides, in pattern recognition, the models that deal with real-world problems have their own limitations and errors [30]. Classifier Ensembles aim to produce accurate recognition results by training several classifiers [31] and combining their outputs by managing the strengths and weaknesses of the classifiers [30]. In literature various terms have been used for the same notions in classifier combination [2], i.e. classifier ensembles have different names, such as, ensemble based systems, mixture of experts, classifier fusion, committee of classifiers, multiple classifier systems [28].

There are many reasons to build ensembles. Dietterich states statistical, computational and representational reasons to construct ensemble based systems [32]. Statistically, with sufficient data different classifiers can be obtained [32] and combining several classifiers may reduce the risk of making the wrong decision [28]. Computationally, when a classifier is stuck in a local optima it may not perform well or some classifiers, such as neural networks, may perform different based on the initial parameters. Hence combining separately trained classifiers may perform better than selecting the best network and eliminating the others. On the other hand, different classifiers trained on the same dataset may perform differently. In the feature space each classifier may have its own region that it performs best. In some applications different types of features (representation/description) can be obtained and different types of classifiers can be trained on each set of features. For example: in person identification, one can obtain face, voice and handwriting information. Also in neurological disorder diagnosis MRI scan, EEG recording, blood test results can be obtained [28]. In addition to different

representations, different training sets recorded at different times and using different features may also be available.

Different taxonomies of classifier ensembles have been suggested in the literature. Kuncheva [2] divides classifier ensemble framework into four parts: instance, feature, classifier and combination levels (see Figure 2.1). Lam [33] categorizes classifier combination methods into multiple, conditional, hierarchical and hybrid topologies. On the other hand, in a recent work, Rokach [34] presents a new taxonomy on classifier ensembles: inducer, combiner, diversity, size and members' dependency. Please refer to [2, 28, 34] for further information on classifier ensemble taxonomies.

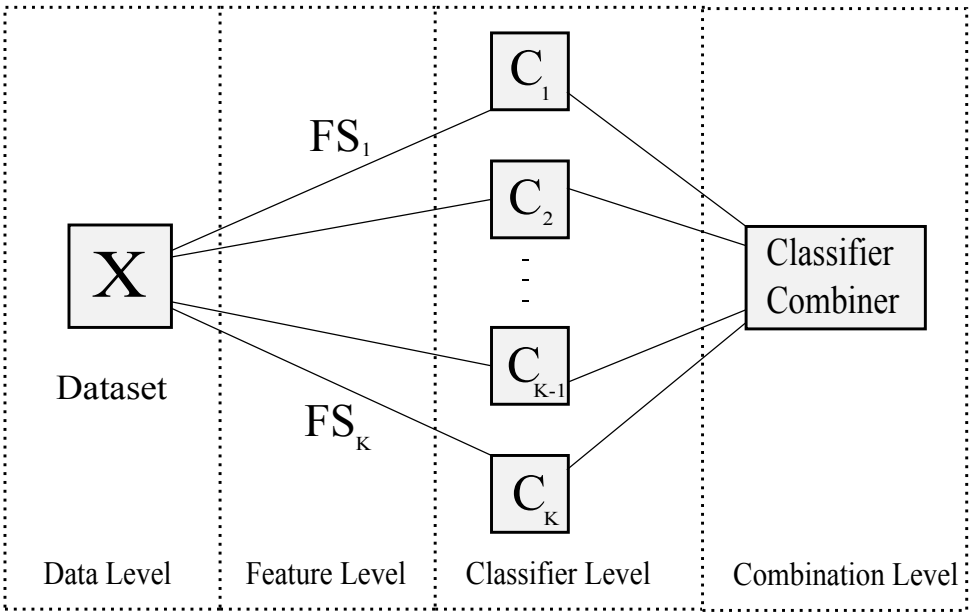


Figure 2.1: General framework of classifier ensembles [2].

In instance level classifier combination, different datasets are bootstrapped from a training dataset and different classifiers are trained. These techniques work well with unstable classifiers [35], such as decision trees, neural networks, where a small change in the dataset, may causes a major change in the hypothesis [32]. Well known ensemble algorithms in instance level combination category are; Bagging [36] and Boosting algorithms [35]. Feature level approach aims to reduce the dimensionality of feature vectors of the base learners in order to reduce the curse of dimensionality. Some of the feature level algorithms are RAS (Random Subspace Method) [17], Input Decimation Approach [37], and Mixture of Experts [2]. Details of the RAS algorithm will be given in the next chapter. In classifier level, different types of classifiers can be used. Classifier decision combination can be either a classifier selection or classifier

combination. In classifier selection a classifier is selected to give the final decision. Classifier ensemble members can be generated either in parallel or sequentially where subsequent classifiers are created based on the preceding classifiers [38]. The next sections summarize some of the well known classifier ensemble algorithms.

2.1.1 Bagging

Bootstrap Aggregation (Bagging) [36] generates multiple versions of a classifier by training individual classifiers on bootstrapped samples of the training set, using them as new learning sets. Each example in each data subset is selected randomly with replacement and each classifier is trained on the average on 63.2 percent of the entire training set [39]. The generated classifiers are aggregated by majority or weighted voting methods. Bagging performs well with unstable algorithms such as decision trees and multilayer perceptrons where small change in the training set creates a large difference in the classifier [39]. Pseudo code of Bagging is given in Algorithm 1.

Algorithm 1 Bagging Algorithm

// X =$(X_1, X_2, ..., X_n)$ be the training dataset with n samples.
// X_i: ith training instance $(i = 1, 2, ..., n)$ and $X_i = (x_{i1}, x_{i2}, ..., x_{id})$
// d: the dimensionality of training instance
Training:
for k = 1 to K **do**
  Take a bootstrap sample $\hat{X}^k$ from X
  Train classifier C_k using $\hat{X}^k$
end for
Testing:
Given a test instance t
Run $C_1 ... C_k$ on the input t
Choose the class with the maximum number of votes as the label of t

2.1.2 Boosting

In boosting methods, at each iteration learning algorithms use a different weighting or distribution for training. The probability of selecting an individual is adapted at each iteration based on the performance of previous classifiers. The weights of misclassified instances are increased at each iteration. Experimental results show that while boosting is sensitive to noise, bagging is effective with noisy data [35]. The most popular boosting algorithm is adaptive boosting (Adaboost) that keeps adding components until

a predetermined error rate on training dataset is reached [40, 41]. The Adaboost.M1 algorithm [28] for multi-class problems is given in Algorithm 2.

Algorithm 2 Adaboost Algorithm

// X =$(X_1, X_2, ..., X_n)$ be the training dataset with n instances.
// X_i: ith training instance $(i = 1, 2, ..., n)$ and $X_i = (x_{i1}, x_{i2}, ..., x_{id})$
// d: the dimensionality of training instances
// c: number of classes
// Initialize the probability of selecting ith instance: $D_1(i) = 1/n$, $i = 1, 2, ..., n$
Training:
for t = 1 to K **do**
  Select a training instance subset $\hat{X}^t$ drawn from the distribution D_t
  Train classifier C_t using $\hat{X}^t$
  Calculate, e_t, the error of C_t
  if e_t>05 **then**
    abort
  end if
  $\beta_t = e_t/(1 - e_t)$
  $Z_t = \sum_i D_t(i)$ // Normalization constant
  if $C_t(x_i) = l_i$ **then**
    $D_{t+1}(i) = D_t(i) \times \beta_t/Z_t$
  else
    $D_{t+1}(i) = D_t(i)/Z_t$
  end if
end for
Testing:
Given a test instance x
Run $C_1 ... C_K$ on input x
Obtain total vote for each class $j = 1, 2, ..., c$
$V_j = \sum_{t:C_t(x)=l_j} \log(1/\beta_t)$
Choose the class that receives the highest total vote

2.1.3 Mixture of experts

Mixture of Experts is also another layered classifier ensemble algorithm. In this algorithm in the second layer instead of a classifier there is a selector which determines the participation of the classifiers in the final decision. This algorithm was initially proposed for Neural Networks where each Neural Network is responsible for a portion of the feature space [2]. The outputs of each Neural Network are given to a gating network and the outputs of the gating network is the probability of each classifier to participate for decision. The selector uses these probabilities to give the outputs of the examples.

2.1.4 Stacked generalization

Stacked generalization is a layered algorithm that aims to find a mapping between ensemble classifier outputs and original class labels. Thus at the first level the ensemble classifiers receive the data as input and at the second layer the outputs of the classifiers in the first layer are given as inputs [42]. The algorithm works as follows: Specifically the training data is divided into K folds. Each first level classifier C_k is trained on the different $K-1$ fold of the training data. For each classifier the remaining one fold of the training data is used as a test set. The outputs of the classifiers and their true labels are used as an input for the second layer classifier [28]. The aim is to learn the classifiers that consistently classify instances correctly or incorrectly.

2.1.5 Input decimated ensembles

The aim of the Input Decimated Ensembles is to de-correlate the base classifiers by training them on different subsets of the input features, selected from the ones that are most correlated with a particular class label [37, 43]. In a c class problem, Input Decimation trains c classifiers, each of them corresponds to one class. For each classifier a user determined number of features, having the absolute correlation to the class label are selected. The objective is to get rid of the features that are not related to each class. In [37] Input Decimation results are evaluated over a synthetic dataset and multi-layer perceptrons are used as base classifiers and combination is achieved by averaging.

2.1.6 Classifier combination methods in classifier ensembles

The decisions of the ensemble of classifiers depend on the output of each classifier. The combination of the classifier outputs can be considered under the categories of: combination of abstract level outputs, combination of ranked lists and combination of continuous level outputs [31, 44]. Kuncheva adds one more type, the oracle level, where the output of a classifier for a given example is only known as correct or incorrect [2].

For further information we refer the Kuncheva's book on classifier combination [2] and [27, 28, 31, 45, 46].

2.1.6.1 Combinations of abstract level outputs

Combination of abstract level outputs consists of the combination methods that use the classifiers whose output is a unique class label. This combination scheme consists of majority vote, weighted majority vote, Bayesian formulation, a Dempster-Shafer theory of evidence, the Behavior-Knowledge Space method [31].

Voting methods:

Since the combination is achieved only on the outputs of the classifiers without training any combiner, majority voting is the simplest method to implement [2, 31]. The output class label of a data point is decided by the major class label obtained by different classifiers. The general majority vote is the special kind of weighted majority vote where each weight is equal. If the classifiers in the ensemble have different accuracies then giving more weights to the accurate classifiers may improve the ensemble accuracy. Weighting can be obtained using a genetic algorithm according to an objective function or the performance of the classifiers on the training dataset [31].

Bayesian combination rule:

Bayesian combination rule finds the weights of classifiers by using their performances on the training dataset. Therefore the confusion matrix of each classifier is used as an indicator for its performance. For a problem with c class possibilities and plus reject option, the confusion matrix size will be $c(c+1)$. Confusion matrices for all classifiers are calculated and based on these matrices using Bayesian formula *belief* values for each class are obtained. For any input sample, the class whose *belief* value is the highest is chosen. Formulation and detailed descriptions about the Bayesian combination can be found in [31].

Behavior-Knowledge space:

Bayesian method assumes the conditional independence of the decisions of the classifiers. Behavior-Knowledge Space method also finds the ensemble from the decisions of the classifiers and can be considered as a refinement of the Bayesian method without assuming conditional independence [31]. High order probabilities are computed from the frequencies on the training set. The algorithm keeps the output combination of the classifiers in the training dataset and creates a table from these combinations. During training, the output combinations of the classifiers and correct

labels are kept on the table. The output combinations are assigned to a class based on the maximum true class decision in the training set [47].

2.1.6.2 Combinations of ranked lists

Some classifiers may output order (ranking) of possible class labels [48]. Instead of the best guess of the classifiers they give a complete ranking of the possible classes. Borda count [28], is a ranked lists combination method used to determine the ranking of the experts without training. Variations of Borda count are used in many real life applications; such as European Song Contest (Eurovision), electing officers at certain university senate elections. The class label of the dataset can be obtained by considering all the ranks obtained from different classifiers.

2.1.6.3 Combinations of continuous outputs

These combination schemes deal with the classifiers that output confidence or distance values for each input sample which can usually be accepted as an estimate of the posterior probability of a particular class given an input instance [28]. Basic combination operators used in this scheme are: Maximum, minimum, mean, median, sum and product rules [49] [50].

2.2 Measures of Diversity for Classifier Ensembles

In many pattern recognition problems, it is difficult to obtain a classifier that has a perfect generalization performance. Classifier ensembles aims to train different classifiers and combine their outputs to perform better than a single classifier [28]. Intuitionally, if we have classifiers in an ensemble that make errors on different data points, it is likely to obtain an ensemble superior to a single classifier. If the classifiers in the ensemble make different errors it is probable that they will be corrected in the ensemble. Diversity of a classifier ensemble measures of how likely are classifiers to give different results on the same data point [2]. In general a good ensemble consists of the base classifiers that are as accurate and diverse as possible [38]. Classifier diversity can be achieved using different training datasets, training parameters, classifiers and feature subsets [28]. Most of the popular algorithms such as bagging and boosting provide diversity with generating datasets by re-sampling instances. Similarly with

different initialization parameters, number of layers and etc. Neural Networks may provide diversity in the ensemble. Another way of providing diversity is to use different types of base classifiers such as decision trees, support vector machines and etc.

In literature there are many works to explain the relationship between classifier diversity and accuracy [2, 51–53]. In order to explain this relationship pairwise and non-pairwise diversity measures are proposed [2, 52]. While pairwise diversity is computed between two classifiers, non-pairwise diversity considers the decision of the classifier ensembles. However there is no consensus on what a good measure of diversity should be [52, 53]. Although there are proven connections between diversity and accuracy, in real-world problems there are some doubts on using diversity measures to build classifier ensembles [54].

Commonly used pairwise and non-pairwise diversity measures are given in the following sections [2]:

2.2.1 Pairwise measures

Pairwise diversity measures are simple to compute and evaluated between two classifiers. For K classifiers $K(K-1)/2$ pairwise measures are computed and the ensemble diversity is obtained by averaging. The pairwise diversity measures are based on the joint output of two classifiers C_i and C_k as shown in Table 2.1 [52].

Table 2.1: The 2x2 relationship between classifiers with probabilities

	C_k correct (1)	C_k wrong (0)
C_i correct (1)	a	b
C_i wrong (0)	c	d
Total: a+b+c+d = 1		

The Q-statistics: The Q statistics for classifiers C_i and C_k gives positive values if instances are correctly classified by both classifiers and negative otherwise. It's calculated as follows:

$$Q_{i,k} = \frac{ad - bc}{ad + bc} \tag{2.1}$$

Q value varies between -1 and 1 and the maximum diversity is obtained for $Q = 0$.

The correlation coefficient (ρ): is defined as the correlation between the outputs of the classifiers:

$$\rho_{i,k} = \frac{ad - bc}{\sqrt{(a+b)(c+d)(a+c)(b+d)}} \tag{2.2}$$

If the classifiers are uncorrelated, ρ = 0, then maximum diversity is obtained.

The disagreement measure(D): is defined as the probability that the classifiers disagree:

$$D_{i,k} = b + c \tag{2.3}$$

The double-fault measure (DF): is defined as the probability that the classifiers are both incorrect:

$$DF_{i,k} = d \tag{2.4}$$

2.2.2 Non-pairwise measures

Non-pairwise diversity measures consider the decision of the classifiers in the ensemble. Some of the most commonly applied non-pairwise diversity measures are as follows [2, 25]:

Kohavi-Wolpert Variance (KW) : Kohavi and Wolpert derived a formula for the variability of the predicted class labels for a specific classifier model. Kohavi Wolpert variance diversity is derived from this formula [2]. Let $X = (X_1, X_2, ..., X_n)$ be the training dataset with n samples. X_i be the d dimensional ith training instance, $X_i = (x_{i1}, x_{i2}, ..., x_{id})$ and $(i = 1, 2, ..., n)$. Let $f(X_j)$ be the number of classifiers that correctly recognizes the X_j, among o total of K classifiers, the KW-variance is computed as follows:

$$kw = \frac{1}{nK^2} \sum_{j=1}^{n} f(X_j)(K - f(X_j)) \tag{2.5}$$

KW and disagreement measures are linearly related to each other [2].

The Entropy Measure (Ent): If half of the classifiers in the ensemble are correct and the rest are wrong then the highest diversity will be obtained. Let $y_i = [y_{1i}, y_{2i}, ..., y_{ni}]^T$ be n dimensional binary vector such that $y_{ji} = 1$ if the classifier C_i recognizes X_j

correctly and $y_{ji} = 0$ otherwise, $i = 1, 2, ..., K$. The highest diversity among classifiers are for a particular X_j is obtained by $\lfloor K/2 \rfloor$ of votes in y with the same value and the other part $K - \lfloor K/2 \rfloor$ with the alternative value. Thus Entropy measure is defined as follows:

$$Ent = \frac{1}{n} \sum_{j=1}^{n} \frac{1}{K - \lfloor K/2 \rfloor} \min \left(\sum_{i=1}^{K} y_{ji}, K - \sum_{i=1}^{K} y_{ji} \right) \quad \textbf{(2.6)}$$

Entropy value varies between 0 and 1. 0 indicates that there is no diversity between classifiers and 1 indicates their complete dependence.

Generalized Diversity(GD): Let Y be a random variable expressing the proportion of classifiers that fail on a randomly drawn object $x \in R^n$. Let p_i denote the probability that $Y = i/K$. $p(i)$ be the probability that i randomly chosen classifier will fail on randomly chosen x. The GD is defined as:

$$p(1) = \sum_{i=1}^{K} \frac{i}{K} p_i, \quad p(2) = \sum_{i=1}^{K} \frac{i-1}{K-1} p_i, \quad GD = 1 - \frac{p(2)}{p(1)} \quad \textbf{(2.7)}$$

GD varies between 0 and 1. Minimum diversity is obtained when $p(2) = p(1)$ and maximum diversity is achieved when $p(2) = 0$.

Coincident Failure Diversity (CFD): CFD is a modified version of GD and is calculated as:

$$CFD = \begin{cases} 0 & p_0 = 1 \\ \frac{1}{1-p_0} \sum_{i=1}^{K} \frac{K-i}{K-1} p_i & p_0 < 1 \end{cases} \quad \textbf{(2.8)}$$

The maximum value of CFD is 1 and it is achieved when all misclassifications are unique.

2.3 Information Theoretic Analysis of the Classifier Ensembles

Diversity measures introduced in this chapter have been used in many applications. Especially they have been used for classifier selection. However it is observed that maximizing diversity does not always result in successful classifiers [26]. There are also some studies showing that diversity measures are confusing and ineffective [2,54]. Therefore alternative attempts to analyze the classifier ensembles are emerging. Recently Brown in [25] and Meynet and Thiran in [26, 55] examine the classifier

ensembles in an information theoretic view. Information theory has been applied to many fields from communication to biology and machine learning [25]. It has also been used for feature extraction, selection and pattern classification [26].

Brown analyzed the classifier ensembles in an information theoretic view and expanded mutual information among classifiers in an ensemble into *accuracy* and *diversity* components [25]. On the other hand Meynet and Thiran analyzed the dependency between classifiers and their accuracies using information theoretic approach by considering classifiers trained on data coming from the same physical distribution [26, 55].

In this thesis first we will give Brown's information theoretic approach. Then we will relate it with Meynet and Thiran's approach.

Let X be the dataset, l represent the labels, and C be any classifier. The aim of any classification algorithm is to estimate the labels: $\hat{l} = C(X)$. Error of any classifier, $p(\hat{l} \neq l)$, is bounded by the following inequalities [25]:

$$\frac{H(l) - I(X;l) - 1}{\log(|l|)} \leq p(\hat{l} \neq l) \leq \frac{1}{2} H(l|X) \tag{2.9}$$

Where $H(l)$ is the entropy of l and $I(X;l)$ is the mutual information between X and l. Details of the entropy and mutual information are given in Appendix B. In order to increase the classification accuracy $H(l|X)$ should be minimized and $I(X;l)$ maximized. Similarly, in a classifier ensemble with a set of K classifiers, $S = \{C_1, C_2, ..., C_K\}$ (C_i, represents the output of the classifier and $i = 1, 2, ...K$), mutual information between classifier outputs and class labels, $I(C_{1:K};l)$, should be maximized. Shannon mutual information computes the dependency between variable pairs. In order to compute the dependencies between multiple variables, multivariate mutual information, *Interaction Information* can be used [25]. Then using Interaction Information the ensemble mutual information, $I(C_{1:K};l)$, can be expanded as follows:

$$I(C_{1:K};l) = \sum_{i=1}^{K} I(C_i;l) - \sum_{C \subseteq S, |C|=2..K} I(\{C\}) + \sum_{C \subseteq S, |C|=2..K} I(\{C\}|l) \tag{2.10}$$

In Equation **10**, the first term, $\sum_{i=1}^{K} I(C_i;l)$, is the *relevancy* of a classifier output to the class label. The second term is a subtractive term independent of the class labels l. It

is the interaction information among the possible subsets of the classifiers and referred as ensemble *redundancy*. The last term is an additive term, contains the class labels, and is referred to as *conditional redundancy*. In order to maximize Equation **10**, the second term should be minimized while the others are maximized. The summation is obtained over all possible subsets of classifiers and it can be splitted into low-order and high-order diversity terms as follows:

$$\begin{aligned} I(C_{1:K};l) = \sum_{i=1}^{K} I(C_i;l) &- \sum_{|C|=2} I(\{C\}) + \sum_{|C|=2} I(\{C\}\,|l) \\ &- \sum_{|C|=3} I(\{C\}) + \sum_{|C|=3} I(\{C\}\,|l) \\ &- \ldots \qquad\quad + \ldots \\ &- \sum_{|C|=K} I(\{C\}) + \sum_{|C|=K} I(\{C\}\,|l) \end{aligned} \tag{2.11}$$

This equation can be interpreted as:

$$\begin{aligned} I(C_{1:K};l) = \text{Individual Mutual Information} &+ \text{2-way diversity} \\ &+ \text{3-way diversity} \\ &+ \ldots\text{-way diversity} \\ &+ K\text{-way diversity} \end{aligned} \tag{2.12}$$

If the classifiers are statistically independent, then the diversity would be; $I(C_{1:K};l) = \sum_{i=1}^{K} I(C_i;l)$. However in real applications it is difficult to obtain independent classifiers. In [25] 3-way and above diversities are omitted and the ensemble mutual information is approximated using only pairwise interactions:

$$I(C_{1:K};l) \approx \sum_{i=1}^{K} I(C_i;l) - \sum_{j=1}^{K-1} \sum_{k=j+1}^{K} I(C_j;C_k) + \sum_{j=1}^{K-1} \sum_{k=j+1}^{K} I(C_j;C_k|l) \tag{2.13}$$

Similarly Meynet and Thiran also tried to measure the quality of classifier ensembles with information theoretic perspective [26, 55]. They aimed to design a global score that can be used in different classifier ensembles and avoid the limitations of the traditional diversity based techniques. Thus an empirical *information theoretic score* (ITS) is given to measure the goodness of K classifier ensembles combined by majority

voting and this score is also used to select optimal ensemble of classifiers. The proposed (ITS) is:

$$ITS = (1+ITA)^3(1+ITD) \quad \textbf{(2.14)}$$

where ITA is the information theoretic accuracy which is relevance term normalized by the number of classifiers, K, in Equation **13**:

$$ITA = \frac{1}{K}\sum_{i=1}^{K} I(C_i;l) \quad \textbf{(2.15)}$$

ITD is the information theoretic diversity which is the ratio between the number of pairwise classifiers $C(K,2)$ and diversity term in Equation **13**:

$$ITD = \frac{\binom{K}{2}}{\sum_{j=1}^{K-1}\sum_{k=j+1}^{K} I(C_j;C_k)} \quad \textbf{(2.16)}$$

While ITA aims to favour the most accurate classifiers, the second term in ITS aims to increase the diversity of an ensemble with same ITA. ITS was shown to outperform the diversity based selection techniques while selecting classifiers in an ensemble. Note that the proposed model of ITS is a choice and as will be shown in the experiments other similar modeling approaches can be used [26].

3. SUPERVISED LEARNING USING INFORMATIVE FEATURE SUBSPACES

In supervised learning a set of training data is available and classifiers that aim to minimize error on an unseen test data are designed using this a priori information [8]. In this chapter we assume that we are only given a training dataset and no unlabeled data is available. First we give related work on supervised learning with Random Subspaces (RAS). Then we introduce the Relevant Random Subspaces (Rel-RAS) and minimum Redundancy and Maximum Relevance Random Subspaces (mRMR-RAS) algorithms. Next these algorithms are analyzed using RM-characteristics of feature subspaces. Finally, experimental results on 5 real datasets, a synthetic dataset and a real dataset with added redundant and noisy features are given. In the experiments, diversity analyses of ensembles are given using both Kohavi Wolpert variance and information theoretic analysis.

3.1 Related Work

In the previous chapter we summarized the well known off-the-shelf classifier ensemble algorithms. Bagging and Boosting algorithms are well known classifier ensemble methods work on the instance space. However, these algorithms require large number of instances to perform well. If the number of features are much larger than the number of instances, algorithms that work on feature subspaces may perform better. In this section, we summarize the previous supervised feature ensemble algorithms. One of the most well known algorithms that trains classifiers on randomly selected feature subspaces is the Random Subspaces algorithm [17]. This algorithm is detailed in the next section.

There are also some algorithms that work both on instance and feature spaces. Random Forest [56] is a modified version of Bagging algorithm and it differs from Bagging in the construction of decision trees. For each node of the decision tree, features that split the node are selected from the best features among the randomly selected

feature subset. Rotation Forest [57] is also another algorithm that uses bootstrapped data. First the feature space is randomly divided into subsets and Principal Component Analysis (PCA) is applied on these subsets. The training dataset for a base classifier is obtained by rotating the original dataset using the PCA coefficients. Decision tree is used as a base classifier and Rotation Forest was shown to perform better than Bagging, Adaboost and Random Forest.

Genetic algorithms were also used for feature subset selection in classifier ensembles and they performed well [58]. Opitz [59] proposed a genetic algorithm based feature selection method for classifier ensembles. The algorithm creates the initial classifiers by selecting random feature subsets. Then feature subsets are updated by crossover and mutation operations. The fitness of each member is obtained using the classifier accuracy and diversity. The ensemble is constructed using the most fit individual classifiers. Oliveira et. al. [60] proposed 2-level hierarchical multi-objective genetic algorithm approach for ensemble creation. Where in the first level a set of good classifiers are generated by conducting feature selection and in the second level the best ensemble is searched among the classifiers generated in the previous level. However genetic algorithms need to have enough population size to be successful and their computational complexity is very high.

On the other hand, a number of studies investigated the use of feature selection methods in classifier ensembles. Vale et. al. [61] proposed a class based feature selection method to be used in the classifier ensembles. Important features corresponding to each class are selected and based on these features a classifier becomes responsible for each class. However in this method the system needs at least one classifier to correctly recognize each class. In [62], hill climbing, a genetic algorithm, forward sequential selection and backward sequential selection are considered for ensemble feature selection. These search strategies incorporate different diversity measures in the search of the best feature subsets and employ the same fitness function. It is shown that ensemble feature selection can be sensitive to the diversity criteria and the performance of the diversity measures depend on the data being processed.

In this thesis, the proposed algorithms differ from the ones in previous works in terms of feature subspace selection. Instead of the best features as in most of the previous works, the algorithms proposed in this thesis select features randomly. The probability

of selection can be either random (RAS), random based on relevance (Rel-RAS) or based on a randomized version of the minimum Redundancy and Maximum Relevance (mRMR) [7] feature selection algorithm (mRMR-RAS).

3.2 Random Subspaces (RAS)

The Random Subspace (RAS) method was proposed by Ho [17] to construct decision forest by combining multiple decision trees trained on randomly selected feature subspaces. The aim is to avoid overfitting of the decision trees while satisfying maximum accuracy. Later, in addition to decision trees, nearest neighbour classifiers [18], linear classifiers (Pseudo Fisher Linear Discriminant and Nearest Mean Classifier) [19] and Support Vector Machines [63, 64] are also used together with the RAS method and they were shown to perform better than a single classifier.

We assume that we are given a classification problem with c classes. Let X_i $(i = 1,2,...,n)$ be the d dimensional ith training sample, $X_i = (x_{i1}, x_{i2}, ..., x_{id})$, in the training dataset X $=(X_1, X_2, ..., X_n)$ with n samples. S_k be the randomly selected subspace with m $(m < d)$ features. The labels l are represented using 1-of-c coding. Let C_k be the classifier constructed using the training dataset $\hat{X}^k$ that are produced from randomly selected subspaces, S_k $(k = 1,2,...,K)$. In the RAS method, C_k classifiers $(k = 1,2,...,K)$ are constructed and then they are combined by simple majority voting to obtain ensemble classifier C_E. Let the decision of classifier C_k be $d_{k,j} \in \{0,1\}$, $k = (1,2,...,K)$ and $j = (1,2,...,c)$. $d_{k,j}$ is obtained using the decision of the kth classifier as follows:

$$d_{k,j} = \begin{cases} 1, & \text{The } k\text{th classifier chooses class } j \\ 0, & \text{otherwise} \end{cases} \tag{3.1}$$

Then the ensemble classifier C_E is:

$$C_E = \underset{j \in \{1,2,...,c\}}{\arg\max} \sum_{k=1}^{K} d_{k,j} \tag{3.2}$$

The RAS method is given in Algorithm 3. Since the feature subspaces are selected randomly, RAS has the advantage of systematically constructing classifier ensembles that are mutually independent to a certain extent [18]. If the number of instances is small compared to the number of features, one may face with the small sample size

problem (curse of dimensionality). In the RAS method the selected features for each subspace is smaller than the original feature space. The number of instances does not change. Therefore the RAS method may be able to produce feature subspaces that eliminate the curse of dimensionality problem. However, if the dataset has a large number of irrelevant features, RAS may select feature subspaces which do not contain (m)any relevant features. This may result in classifiers which perform poorly, resulting in poor ensemble performance. Therefore more intelligent feature subset selection methods should be used. In the next sections, the Rel-RAS and mRMR-RAS algorithms are proposed to remedy these problems.

Algorithm 3 RAS Algorithm

for k = 1 to K **do**
  $S_k \leftarrow$ Rand(m) //Select random subspaces $S_1 ... S_K$
  Project X to $\hat{X}^k$ using S_k
  Train classifier C_k using $\hat{X}^k$
end for
//Combine classifiers by majority voting:
C_E = MajorityVote(C_1,...,C_K):

3.3 Relevant Random Subspaces (Rel-RAS)

While the RAS method produces subspaces by randomly selecting features, the Rel-RAS selects each feature in the subset based on the relevance scores of the features obtained using the mutual information between feature and class labels. Note that any other method, that computes the correlation between features and labels and gives a probability of selection for each feature could also be used.

Training dataset X can be written in terms of feature vectors, F. Let F_j, $j = \{1,2,...,d\}$ denote the n dimensional feature vector of the jth feature and $F_j = \left(x_{1j}, x_{2j}, ..., x_{nj}\right)$. The relevance $Rel(F_j)$ of a feature F_j, i.e. the mutual information, $I(F_j, l)$, between F_j and the target classes l can be written as:

$$Rel(F_j) = I(F_j, l) = \sum_{i=1}^{n} \sum_{t=1}^{c} p(x_{i,j}, l_{i,t}) log \frac{p(x_{i,j}, l_{i,t})}{p(x_{i,j}) p(l_{i,t})} \tag{3.3}$$

where $x_{i,j}$ denotes the ith feature value of F_j and $l_{i,t}$ denotes the tth class label $(t = 1,2,...,c)$ for the ith training sample.

In order to be able to compute the probabilities in Equation **3**, if features are continuous valued, we first discretize them. For discretization we use 10 equal sized bins placed between the minimum and maximum value observed for the particular feature in the labeled training set. We approximate the probabilities by means of counting the samples that fall into each bin. The details of the discretization algorithm is given in Appendix A.

In the Rel-RAS algorithm K subspaces $(S_1,\ldots,S_K)$, each containing m $(m>0)$ features are created. Each feature subspace is produced using tournament selection [65] between pairs of individual features (i.e. tournament size is 2). The tournament selection is performed as follows: Two features are randomly selected from the set of all available features. Among these two features, the one with higher relevance is added to the subset of selected features. The selected feature is extracted from the set of available features and the procedure is repeated until the set of selected features contains the required number of features. Similar to RAS, in Rel-RAS also, a classifier is trained on each one of the feature subspaces $(S_1,...,S_K)$ and the final classifier is obtained by majority voting.

The main difference between RAS and Rel-RAS is the feature subspace selection. The goal of Rel-RAS' selection scheme is to select random feature subspaces which are as relevant as possible to the class labels. While RAS selects feature subspaces according to a uniform distribution on features, Rel-RAS uses feature probabilities proportional to relevance scores. Using probability of selection proportional to relevance scores ensures that more informative features are selected. Especially for large d, when there is a large number of irrelevant features and a small number of relevant features, in each selected subspace, RAS may select feature subspaces which does not contain any relevant features. This may result in classifiers which perform very poorly, resulting in poor ensemble performance. As the feature subspaces selected contain more features, RAS can select relevant features, however, the larger the subspaces the longer it takes to train and test each classifier.The Rel-RAS algorithm is given in Algorithm 4. The experimental results show that generally Rel-RAS results in better classifiers than RAS algorithm.

In a related work [43], input decimated ensembles, instead of mutual information based relevance scores authors used correlation to select subspaces. The features in the

subspaces consisted of the top most relevant features for discrimination of each class from the rest of the classes. However in the input decimated ensembles the number of the classifiers is limited to the number of classes. On the other hand, selecting the best features may not always give the best classifier. Our approach enables selection of as many random subspaces as needed and the number of classifiers is not limited. We also enable classifier diversity by selecting features randomly.

Algorithm 4 Rel-RAS Algorithm

XD = Discretize(X)
V = Relevance(XD,l) //Mutual Information between features and labels l
//Select subspaces $S_1 \ldots S_k$
for k = 1 to K **do**
  $S_k \leftarrow$ Tournament(V,m)
  Project X to $\hat{X}^k$ using S_k
  Train classifier C_k using $\hat{X}^k$
end for
//Combine classifiers by majority voting:
C_E = MajorityVote(C_1,...,C_K):

3.4 Minimum Redundancy and Maximum Relevance Random Subspaces (mRMR-RAS)

Rel-RAS algorithm selects each feature based on the relevance score between features and class labels. However the redundancy of the features in each feature subspace is not concerned. On the other hand most of the powerful feature selection algorithms consider redundancy between features in order to improve the classification performance by selecting the relevant and non-redundant best features [7, 66]. We also propose mRMR-RAS (minimum Redundancy and Maximum Relevance Random Subspaces) algorithm that considers both the relevance and redundancy in each feature subspace. During the computations we modified mRMR (minimum Redundancy and Maximum Relevance) [7] feature selection scheme. mRMR is a feature selection method which tries to find an ordering of features based on their relevance to the class label. mRMR also aims to select the next feature as uncorrelated as possible with the current subspace of selected features. Mutual information is used as a measure of feature-feature or feature-label similarity.

Let S be the feature subspace that mRMR seeks, the redundancy of S can be described using the within mutual information, W, of S:

$$W = \frac{1}{|S|^2} \sum_{F_i, F_j \in S} I(F_i, F_j) \tag{3.4}$$

Where $I(F_i, F_j)$ is the mutual information between feature vector F_i and feature vector F_j. $|S|$, is the size of the feature subspace S. In order to measure the relevance of features to the target class, again mutual information is used. Let $I(F_i, l)$ denote the mutual information between feature F_i and the target classes l. V, the relevance of S, is computed as:

$$V = \frac{1}{|S|} \sum_{F_i \in S} I(F_i, l) \tag{3.5}$$

Feature selection tries to choose an S with as small W and as large V as possible. So that the selected features are as relevant and as non-redundant as possible. The mRMR method achieves both goals by maximizing either $(V - W)$ which is called MID (Mutual Information Distance) or V/W which is called MIQ (Mutual Information Quotient). MID is used in our computations.

mRMR-RAS algorithm, selects the first feature using the Relevance scores, V, as a probability distribution. Then using redundancy the scores, W, MID scores are calculated and $V - W$ are used as the probability of selecting the next feature. By adding randomness we are able to create diverse, relevant and non-redundant feature subspaces, therefore we try to obtain diverse enough and accurate classifiers. Pseudo code of the proposed algorithm is given in Algorithm 5.

3.5 Accuracy Analysis of the Subspace Selection Algorithms

In this section, we aim to explain why we expect our feature subspace selection methods, Rel-RAS and mRMR-RAS to perform better than random subspace selection RAS. The accuracy analysis of Rel-RAS and mRMR-RAS algorithms will be performed using the RM (Recursively More) characteristic property of feature spaces [7]. Let S_1 and S_2 be two subspaces with m features. S_1 is more *characteristic*, if the classification error, e_1 on S_1 obtained by classifier C is less than the classification error,

Algorithm 5 mRMR-RAS Algorithm

```
XD = Discretize(X)
V = Relevance(XD,l) //Mutual Information between features and labels l
W = Redundancy(XD) // Mutual Information between features
//Select random subspaces S_1...S_k
for k = 1 to K do
  for i = 1 to m do
    if i = =1 then
      S_k(i) ← Tournament(V,1)
    else
      S_k(i) ← Tournament(V − W,1)
    end if
    Project X to X̂^k using S_k
    Train classifier C_k using X̂^k
  end for
end for
//Combine classifiers by majority voting:
C_E = MajorityVote(C_1,...,C_K):
```

e_2 on S_2 obtained by classifier C. Let a series of subsets of S_1 obtained by a feature selection algorithm be:

$$S_1^1 \subset S_1^2 \subset ... \subset S_1^k \subset ... \subset S_1^{m-1} \subset S_1^m = S_1 \tag{3.6}$$

and similarly subsets of S_2 be:

$$S_2^1 \subset S_2^2 \subset ... \subset S_2^k \subset ... \subset S_2^{m-1} \subset S_2^m = S_2 \tag{3.7}$$

S_1 is *Recursively More characteristic* (RM characteristic) than S_2, if $\forall k \ (1 \leq k \leq m)$ the classification error $e_1^k < e_2^k$. However in most cases it is difficult to obtain $e_1^k < e_2^k$, $\forall k$. Let $\rho \ (0 \leq \rho \leq 1)$ be a confidence score that gives the percentage of k values that satisfy $e_1^k < e_2^k$. When $\rho = 0.9$, S_1 is said to be *approximately* RM-characteristic [7]. For the case of Rel-RAS, mRMR-RAS and RAS, let $\bar{e}_{Rel-RAS}$, $\bar{e}_{mRMR-RAS}$ and $\bar{e}_{RAS}$ be the mean of the individual classification errors for Rel-RAS, mRMR-RAS and RAS algorithms, respectively. The mean individual classification error, $\bar{e}$, for any of the algorithm can be computed as follows:

$$\bar{e} = \frac{1}{K} \sum_{k=1}^{K} e^k \tag{3.8}$$

Where e^k is the classification error obtained by the classifier C_k trained on the kth subspace ($k = 1, 2, ..., K$) obtained from a subspace selection algorithm. We

experimentally show that $\bar{e}_{Rel-RAS} < \bar{e}_{RAS}$ and $\bar{e}_{mRMR-RAS} < \bar{e}_{RAS}$, i.e. mean of the individual classification errors for Rel-RAS and mRMR-RAS are smaller than that of RAS for different subspace sizes (See the mean individual classifier accuracies in experimental results in the next section).

3.6 Experimental Results

In this section, we present the experimental results obtained using ensembles of classifiers with RAS, Rel-RAS, mRMR-RAS and single classifiers. First, results on 5 different real datasets: Audio Genre, Optdigits, Classic-3, Isolated Letter Speech and MFeat and one synthetic dataset are presented. Then results on Audio Genre dataset appended with different redundant features are given. Detailed descriptions of the datasets are given in Appendix C. For each dataset, experimental results of Rel-RAS, mRMR-RAS and RAS are obtained on 10 different random runs. At each random run, the whole dataset is divided equally into a training partition and a test partition. Training set is further splitted into unlabeled training set and μ portion of the rest of the training data is used as the labeled training set. In order to compare the supervised and semi-supervised learning results the same data splitting is applied for both learning schemes. Note that unlabeled training set is only used in semi-supervised learning experiments, given in the next chapter and the μ is defined as follows:

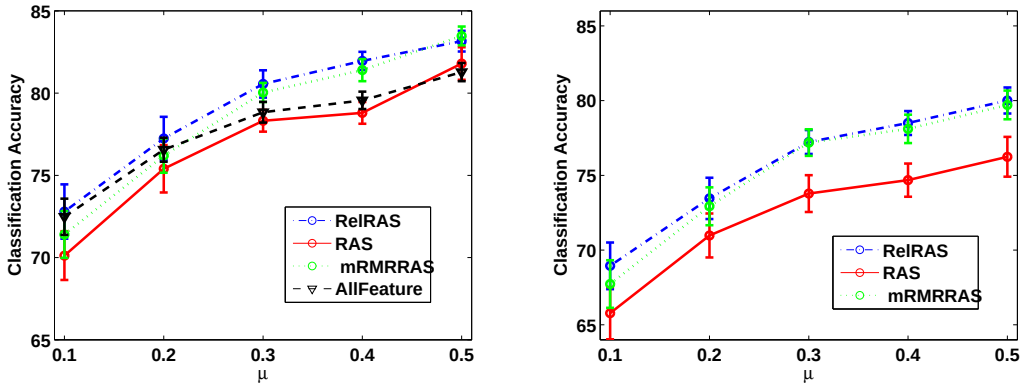
$$\mu = \frac{\text{\#labeled training set used to train classifier}}{\text{\# labeled training set}} \tag{3.9}$$

First, we investigate the effect of the μ and experimental results are given for different values of μ on Audio Genre dataset. The number of selected features, m, is 25 for RAS, Rel-RAS and mRMR-RAS algorithms. Then in order to evaluate the classification accuracies under small number of training datasets and small number of classifier ensembles, the μ is fixed to 0.3 and the K is selected as 5 and 25. The mean ensemble and individual classification accuracies and their standard error bars are given in the figures obtained for Audio Genre dataset. The standard error bars for all results are too low and we give the mean ensemble classification accuracies for all datasets. On the other hand supervised learning results without classifier ensembles, single classifiers, are also given. These results are represented as "Allfeature" in the figures.

Implementation details of classifiers used are as follows: PRTools [67] implementation of KNN (k-nearest neighbor) and LDC (linear Bayes normal classifier) classifiers, Weka J48 [68] implementation of decision tree classifier and Libsvm [69] implementation of Support Vector Machines (SVM) are used as base classifier in the algorithms. The KNN classifier implementation in PrTools uses the value of 3. The LDC classifier [9] computes the linear classifier between the classes, assuming the same class covariance matrix for all the classes. Unregularized class covariance matrix is used in the experiments. The J48 decision tree implementation is used with the default parameters. Linear kernel is used in SVM.

3.6.1 Real data results

Audio Genre Dataset: Mean ensemble and mean individual classification accuracies for KNN classifier with respect to μ are given in Figure 3.1(a) and Figure 3.1(b), respectively. Similarly, mean ensemble and mean individual classification accuracies for LDC, decision tree and SVM classifiers are given in Figure 3.2, Figure 3.3 and Figure 3.4 respectively.

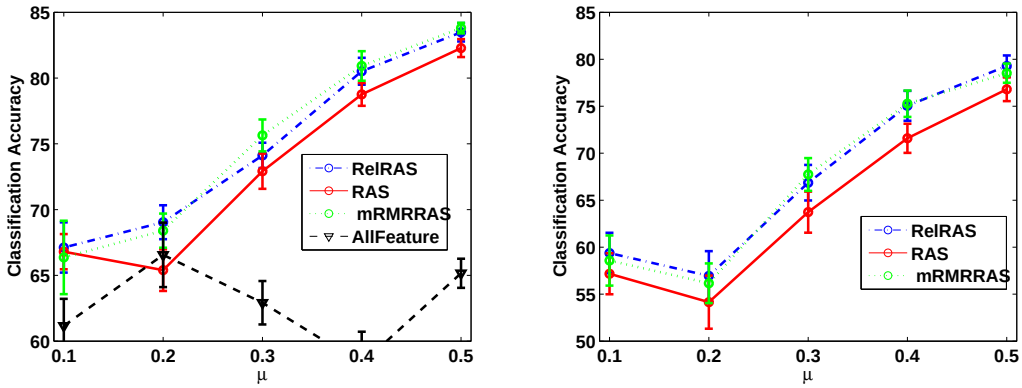


(a) Ensemble, KNN (b) Individual, KNN

Figure 3.1: Mean ensemble and individual test accuracies on Audio Genre dataset obtained by mRMR-RAS, Rel-RAS, RAS and single classifier with respect to μ for $K = 5$, $m = 25$ and classifier = KNN.

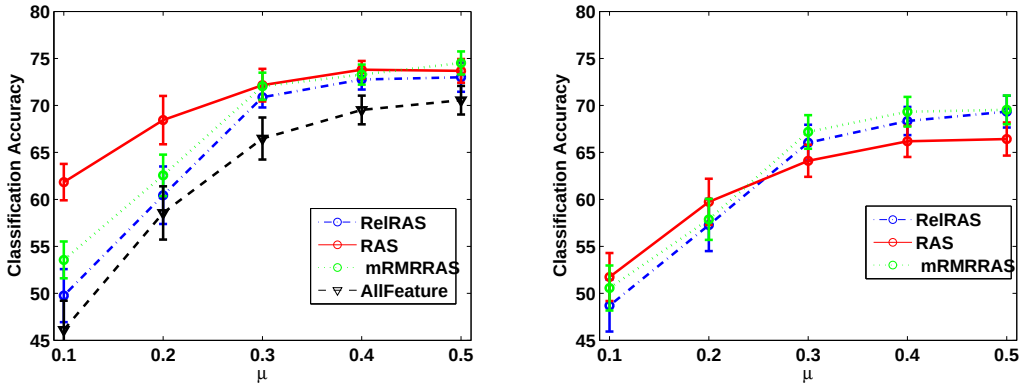
When KNN, LDC and SVM classifiers are used, both proposed algorithms outperform the RAS algorithm. When decision tree is used, RAS algorithm performs better than Rel-RAS and mRMR-RAS. However mean ensemble classification accuracies with decision tree are less than classification accuracies obtained with KNN, LDC and SVM classifiers for different μ. On the other hand except for SVM, single classifier does not perform better than ensemble algorithms. However when small amount of training samples are used single SVM performs slightly better than Rel-RAS and mRMR-RAS

algorithms. But still proposed algorithms perform better than RAS when $\mu = 0.1$ and SVM classifier is used as base classifier. Note that increase μ increases the accuracies of the proposed algorithms and when $\mu > 0.3$ and SVM classifier is used, proposed algorithms outperforms the single SVM and RAS algorithm.



(a) Ensemble, LDC (b) Individual, LDC

Figure 3.2: Mean ensemble and individual test accuracies on Audio Genre dataset obtained by mRMR-RAS, Rel-RAS, RAS and single classifier with respect to μ for $K = 5$, $m = 25$ and classifier = LDC.

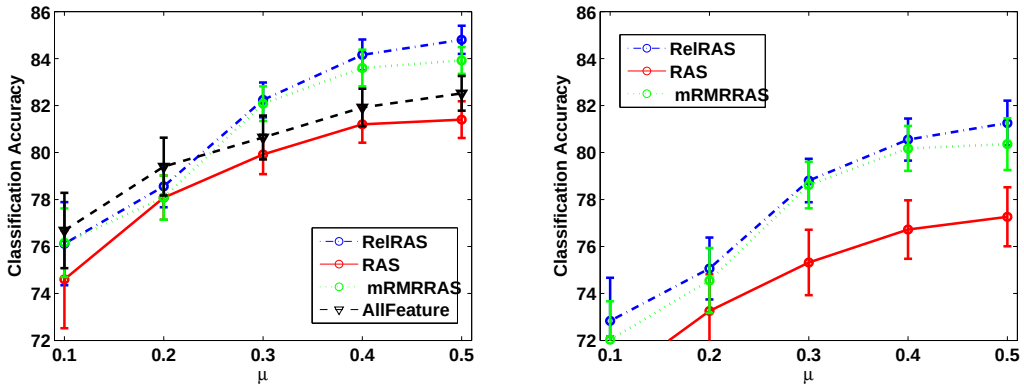


(a) Ensemble, J48 (b) Individual, J48

Figure 3.3: Mean ensemble and individual test accuracies on Audio Genre dataset obtained by mRMR-RAS, Rel-RAS, RAS and single classifier with respect to μ for $K = 5$, $m = 25$ and classifier = J48.

Mean individual classification accuracies of the algorithms show that the proposed algorithms create RM characteristic feature subspaces than RAS algorithm except for decision tree when $\mu < 0.3$. RM characteristic feature subspace also translates into better ensemble accuracy.

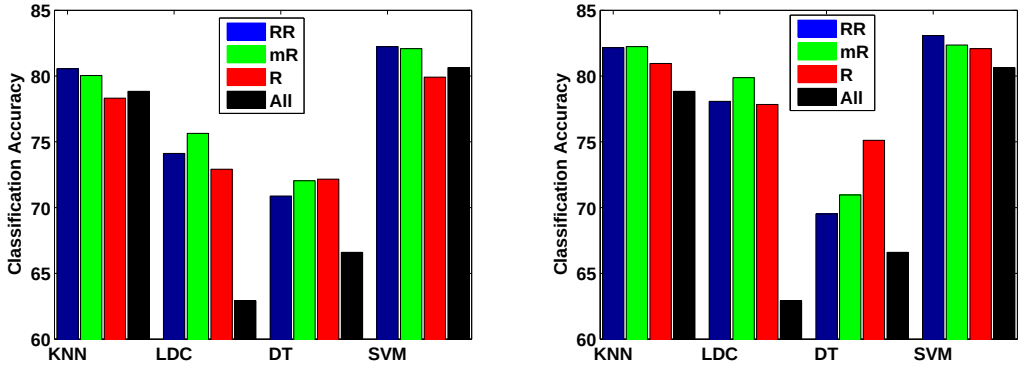
Experimental results show that increase in the number of training samples increases the ensemble accuracy for all algorithms and the proposed algorithms outperform single classifiers and RAS algorithm. In order to evaluate the performance of the algorithms with small number of instances, classification accuracies are also obtained when $\mu =$



(a) Ensemble, SVM (b) Individual, SVM

Figure 3.4: Mean ensemble and individual test accuracies on Audio Genre dataset obtained by mRMR-RAS, Rel-RAS, RAS and single classifier with respect to μ for $K = 5$, $m = 25$ and classifier = SVM.

0.3 and K = 5, 25. Classification accuracies on Audio Genre dataset with respect to different classifiers and different algorithms are given for μ = 0.3 and K = 5, 25 in Figure 3.5. In the figures, the RR, mR, R and All represent the Rel-RAS, mRMR-RAS, RAS and single classifier results respectively. In Figure 3.5, except for decision tree, the proposed algorithms outperform the RAS algorithm and single classifier. The best classification accuracy is obtained by Rel-RAS algorithm with SVM classifier.

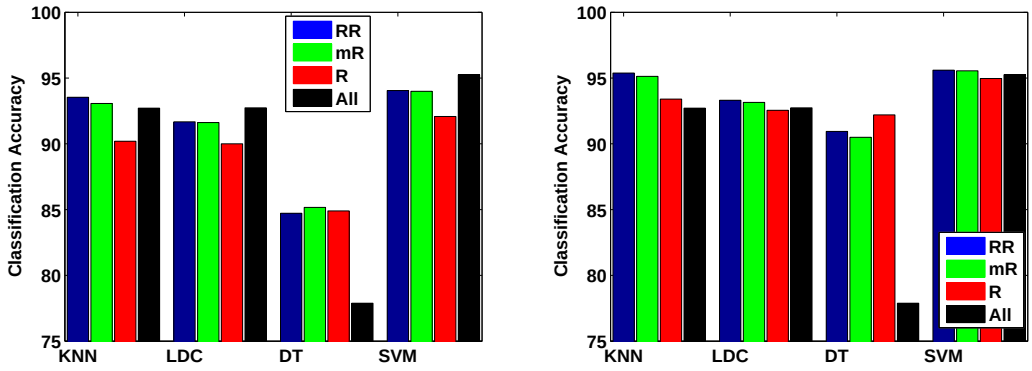


(a) $K = 5$ (b) $K = 25$

Figure 3.5: Mean ensemble test accuracies on Audio Genre dataset obtained by Rel-RAS (RR), mRMR-RAS (mR), RAS (R) and single classifier using All features (All) for $m = 25$.

UCI Optdigits dataset: Mean ensemble test accuracies on Optdigits dataset obtained by mRMR-RAS, Rel-RAS, RAS and All Features with respect to different classifiers are given in Figure 3.6 for K=5 and K=25. When K=5 classifiers are used, the single classifiers perform as good as ensemble learning algorithms. When K=25 classifiers are used, the proposed algorithms perform better than RAS and single classifiers with KNN, LDC and SVM classifiers. When $\mu = 0.3$, the number of instances and attributes

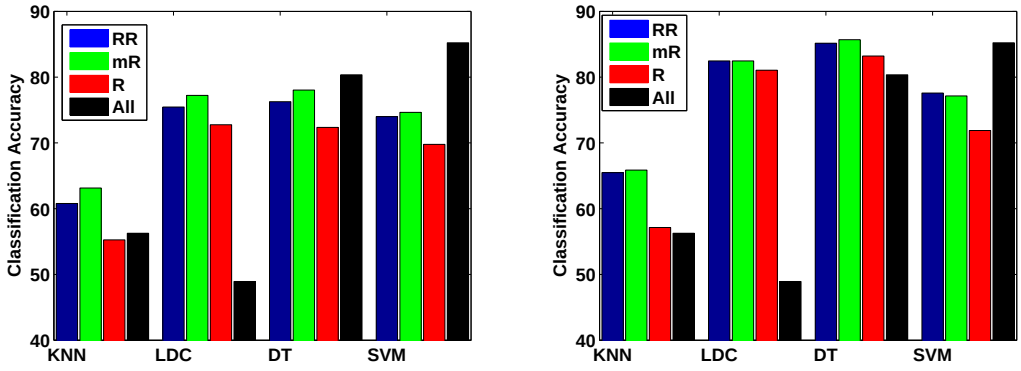
are 425 and 64 respectively. Depending on the classifier used, the number of training instances in the dataset is enough for a single classifier to perform well.



(a) $K = 5$ (b) $K = 25$

Figure 3.6: Mean ensemble test accuracies on Optdigits dataset obtained by Rel-RAS (RR), mRMR-RAS (mR), RAS (R) and single classifier using All features (All) for $m = 25$.

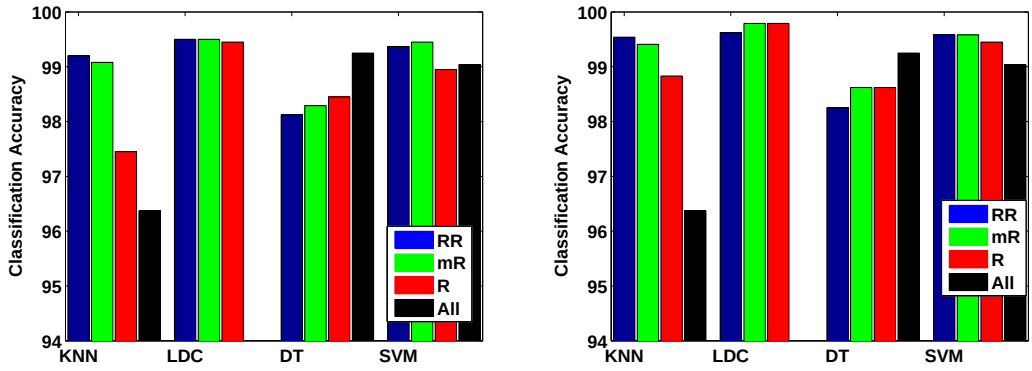
Classic-3 dataset: In Figure 3.7, the mean ensemble test accuracies on Classic-3 dataset obtained by mRMR-RAS, Rel-RAS, RAS and All Features with respect to different classifiers are given for K=5 and K=25. The best classification accuracy is obtained by mRMR-RAS algorithm with decision tree for K=25 classifiers. Due to the sparsity of features in this dataset, any subspace of features may not perform well. On the other hand, m is also another parameter that effects the performance of the algorithms. The effect of the m parameter on the datasets is given in the next section. Note that the proposed algorithms perform better than the RAS and single classifier when decision tree is used as a base classifier.



(a) $K = 5$ (b) $K = 25$

Figure 3.7: Mean ensemble test accuracies on Classic-3 dataset obtained by Rel-RAS (RR), mRMR-RAS (mR), RAS (R) and single classifier using All features (All) for $m = 25$.

UCI Isolated Letter Speech dataset: The mean ensemble test accuracies on Isolated Letter Speech dataset obtained by mRMR-RAS, Rel-RAS, RAS and All Features with respect to different classifiers are given in Figure 3.8. The proposed algorithms outperform both RAS and single classifier when KNN, LDC and SVM classifiers are used. When the decision tree is used, the RAS algorithm performs better than the proposed algorithms. Additionally, the single LDC classifier performs less than 50 %. Note that when $\mu = 0.3$, the number of instances and attributes are 36 and 617 respectively. Therefore LDC classifier is effected by the small sample size problem. When we increase the number of instances in the training set to 240, the mean test classification accuracy of single LDC increases to 80 %.

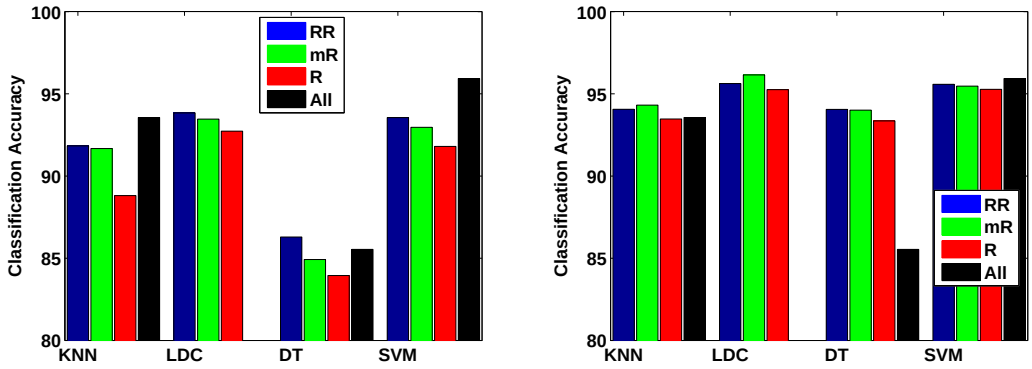


(a) $K = 5$ (b) $K = 25$

Figure 3.8: Mean ensemble test accuracies on Isolated Letter Speech dataset obtained by Rel-RAS (RR), mRMR-RAS (mR), RAS (R) and single classifier using All features (All) for $m = 25$.

MFeat dataset: The mean ensemble test accuracies on MFeat dataset obtained by mRMR-RAS, Rel-RAS, RAS and All Features with respect to different classifiers are given in Figure 3.9. The best classification accuracy is obtained by single SVM classifier. Additionally, mRMR-RAS algorithm with LDC classifier for K = 25 performs as good as single SVM classifier. Note that the proposed algorithms outperform RAS algorithm. On the other hand single decision tree and LDC classifiers do not perform well on MFeat dataset. When $\mu = 0.3$, the number of instances and attributes are 150 and 649 respectively. Therefore LDC classifier is affected by small sample size problem.

The significance of the experiments is also evaluated with t-test. We have obtained p values for 10-fold cross validation accuracies of RAS, Rel-RAS and mRMR-RAS algorithms. The significance values, when $K = 25$ subspaces are selected for the



(a) $K = 5$ (b) $K = 25$

Figure 3.9: Mean ensemble test accuracies on Mfeat dataset obtained by Rel-RAS (RR), mRMR-RAS (mR), RAS (R) and single classifier using All features (All) for $m = 25$.

algorithms are given in Table 3.1 and Table 3.2. Details of the t-test can be found in Appendix H.

Table 3.1: t-test p values of RAS and Rel-RAS algorithms for each dataset, K=25, μ = 0.3 and m=25.

Classifier	audio	optdigits	classic-3	isolet	mfeat
KNN	0.15	0.00	0.00	0.01	0.15
LDC	0.85	0.01	0.32	0.18	0.31
J48	0.00	0.00	0.22	0.31	0.18
SVM	0.31	0.00	0.00	0.48	0.35

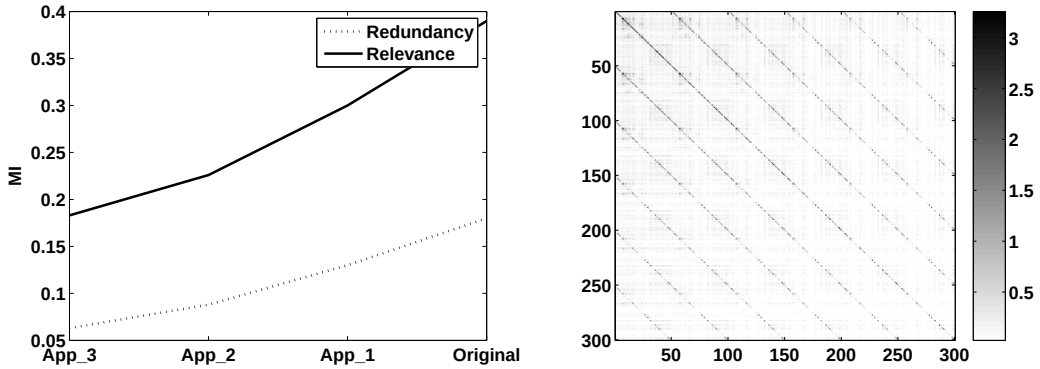
Table 3.2: t-test p values of RAS and mRMR-RAS algorithms for each dataset, K=25, μ = 0.3 and m=25.

Classifier	audio	optdigits	classic-3	isolet	mfeat
KNN	0.15	0.00	0.00	0.05	0.03
LDC	0.15	0.02	0.34	1	0.00
J48	0.04	0.00	0.13	1	0.00
SVM	0.78	0.02	0.00	0.533	0.6

According to Table 3.1, except for the Audio genre dataset, with 90% probability, generally Rel-RAS ensemble accuracy is better than that of RAS. In Table 3.2 p values for the 10-fold cross validation accuracies of RAS and mRMR-RAS algorithms are given. Similar p values obtained between RAS and Rel-RAS algorithms are generally valid between RAS and mRMR-RAS algorithms. Except for isolet dataset, with 90% probability, generally mRMR-RAS ensemble accuracy is better than that of RAS.

3.6.2 Robustness to redundant features

In these experiments we evaluate algorithms' robustness to redundant features. Real datasets used in our experimental results are carefully obtained and they do not have redundant features. Audio Genre dataset has the highest average feature relevance to class labels in the experimental datasets. Therefore the feature space in this dataset is appended with different powers of the original features in order to obtain redundancy. Three datasets, App_1, App_2 and App_3, are generated with different powers of the original features. Dataset App_1 represents the case where the orginal feature space $[x]$ is appended with 2nd and 3rd powers of the original features $[x^2 x^3]$. The new feature space in App_1 dataset contains 150 features. Similarly App_2 dataset represents the case where the original feature space $[x]$ is appended with 2nd, 3rd,..., 5th powers of the original features $[x^2 x^3 x^4 x^5]$. The total number of features in App_2 dataset is 300. The last toy dataset App_3, where the original feature space $[x]$ is appended with 2nd, 3rd,..., 8th powers of the original features $[x^2, \ldots, x^7 x^8]$, has 450 features.



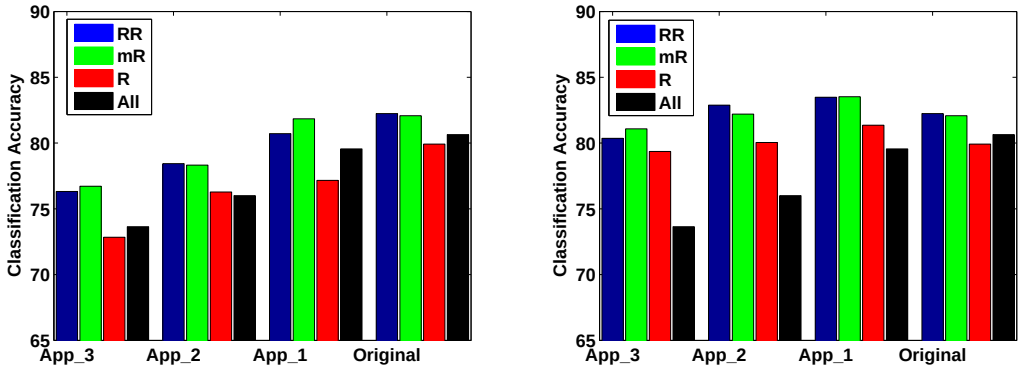
(a) Relevance and redundancy (b) Pairwise redundancy between features

Figure 3.10: a) Relevance, redundancy analysis and b) redundancy map of Audio Genre dataset appended with redundant features.

In Figure 3.10(a) the mean relevance and the mean redundancy values in the datasets are given. We see that increase in the appended features decreases the mean redundancy in the dataset. This is because of the increase of the number of features and their possible pairwise combinations. Although the mean redundancy values in the datasets decrease the mean relevance values in the datasets are also decreased. Therefore datasets start to have less relevant features when we append different powers of the features to the original feature space. Figure 3.10(b) reports the pairwise redundancy between features of the App_2 dataset. The diagonal elements of the

appended features have the highest mutual information. These results show that there is a strong mutual information between each feature and features obtained by it's powers.

In Figure 3.11(a) and Figure 3.11(b) the mean classification accuracies obtained using SVM classifier on Audio Genre dataset appended with redundant features are given for $K = 5$ and $K = 25$ classifiers, respectively. The proposed algorithms outperform both RAS and single classifier. On the other hand, increase in the number of classifiers in the ensemble increases the classification accuracy. Figure 3.11 shows that, even the redundancy of the dataset is low (App_1 dataset), the proposed algorithms outperform both RAS and single SVM classifier. Note that increase in the redundant features decreases the single SVM's and classification performance of the ensembles.

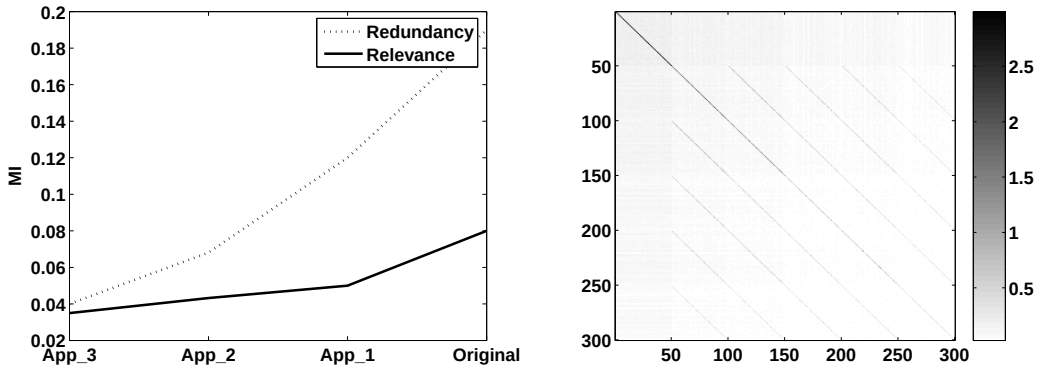


(a) $K = 5$ (b) $K = 25$

Figure 3.11: Mean ensemble test accuracies on Audio Genre dataset appended with redundant features obtained by Rel-RAS (RR), mRMR-RAS (mR), RAS (R) and single classifier using All features (All) for $\mu = 0.3$, $m = 25$ and classifier = SVM.

3.6.3 Synthetic data results

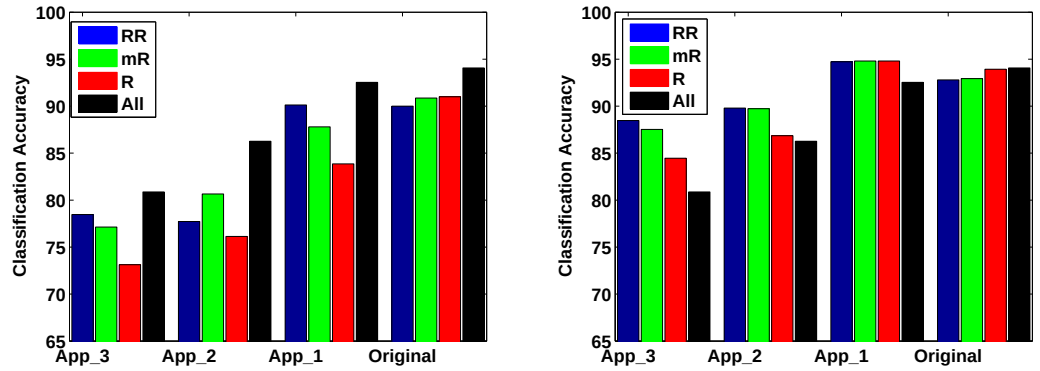
The classification accuracies of the algorithms are also evaluated with a synthetic two class dataset. The dataset is generated from Gaussian distributions with a covariance matrix 10 at diagonal and mean -1 for one class and 1 for the other class. The total number of features and instances are chosen to be 50 and 300 respectively. In the experiments in order to obtain redundant features, the feature space is appended with different powers of the original features. As in the previous experiments, App_1 represents the case where the original feature space $[x]$ is appended with 2nd and 3rd powers of the features $[x^2 x^3]$. App_2 and App_3 datasets are obtained using the same way used to obtain redundant features in Audio Genre dataset.



(a) Relevance and Redundancy (b) Redundancy Map

Figure 3.12: a) Relevance, redundancy analysis and b) redundancy map of synthetic dataset appended with redundant features.

The mean relevance and the mean redundancy values in the synthetic datasets are given in Figure 3.12(a). As in the Audio Genre dataset appended with redundant features results, in the synthetic dataset results we see that increase in the appended features decreases the mean redundancy and the mean relevance in the dataset. In Figure 3.12(b) the pairwise redundancy between features of the App_2 dataset is given. The diagonal elements of the appended features have the highest mutual information. These results show that there is a strong mutual information between different powers of appended features.



(a) $K = 5$ (b) $K = 25$

Figure 3.13: Mean ensemble test accuracies on synthetic dataset appended with redundant features obtained by Rel-RAS (RR), mRMR-RAS (mR), RAS (R) and single classifier using All features (All) for $\mu = 0.3$, $m = 25$ and classifier = SVM.

In Figure 3.13(a) and Figure 3.13(b) mean classification accuracies obtained using SVM classifier on synthetic dataset appended with redundant features are given for $K = 5$ and $K = 25$ classifiers, respectively. From the figures it can be seen that increase in the number of classifiers in the ensemble increases the classification accuracy.

Single SVM classifier performs better than the ensemble algorithms when there is no redundancy in the features. When $K = 25$ classifiers are used the proposed algorithms outperform both RAS and single SVM classifier even the redundancy is low (App_1 dataset). The features in the synthetic dataset are uncorrelated and single SVM performs better than ensemble algorithms when $K = 5$ are selected.

3.6.4 Classifier diversity and information theoretic analysis of the algorithms in supervised learning

Classifier diversity is suspected to affect the ensemble accuracy and there have been efforts to explain the relationship between classifier diversity and ensemble accuracy [2]. In our experiments, classifier diversities are measured on the test dataset using the Kohavi Wolpert variance diversity measure [2] given in Equation **5**. KW-variance and most of the diversity measures only consider the outputs of the classifiers and there are some doubts about using these diversity measures. In order to analyze the classification performance, two mutual information based accuracy and diversity analysis are given in Section 2.3. The first one is Brown's [25] information theory based low order diversity and the second one is Meynet's ITS [26]. In our experiments in addition to KW-variance we first used the Brown's low order diversity and examined the ensemble mutual information using Equation **13**. Next we also analyzed the ensemble accuracies with ITS. Except for the Classic-3 dataset we found that there is a direct relationship between the ITS given in Equation **14** and classification accuracy. However as stated in [26] the model choice for ITS can be changed. Therefore to capture the relationship between ensemble diversity and accuracy in all datasets, without changing the order, Equation **14** is modified as follows:

$$ITS = (A + ITA)^3 (B + ITD) \qquad \textbf{(3.10)}$$

Where A and B are the constant terms for ITA and ITD respectively. We experimentally found A and B, 0.12 and 0.08, respectively.

In Figure 3.14 KW-variance, low order diversity and ITS analysis of Audio Genre dataset with KNN, LDC, decision tree and SVM classifiers are given. We see that KW-variance and LOD have the same tendency and the proposed algorithms are less diverse than RAS algorithm. On the other hand, the best classification accuracy is

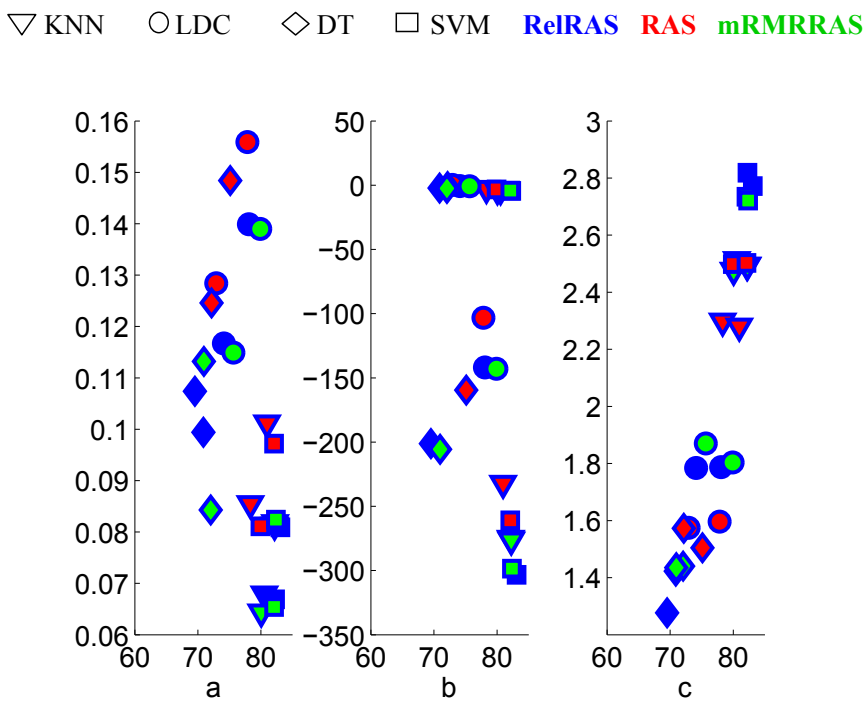


Figure 3.14: Classification accuracy versus diversity on Audio Genre dataset obtained by mRMR-RAS, Rel-RAS and RAS for $\mu = 0.3$, $K = 5,25$ and $m = 25$ a)KW-variance b) LOD c) ITS.

obtained with Rel-RAS using SVM classifier and it has the highest ITS. Similarly, KW-variance, low order diversity and ITS analysis of Optdigits, Classic-3, Isolated and MFeat datasets with KNN, LDC, decision tree and SVM classifiers are given in Figure 3.15, Figure 3.16, Figure 3.17 and Figure 3.18, respectively.

We found out that the KW-variance classifier diversity with Rel-RAS and mRMR-RAS algorithms are slightly less than the classifier diversity with RAS. Also a direct relationship between increase in the ensemble accuracies and ensemble mutual information could not found with low order diversity measure. But it can be seen that KW-variance and mutual information based low order diversity measure have the same tendency for all datasets. These results show that, classifiers combined in Rel-RAS and mRMR-RAS algorithms more agree on class labels of test data than RAS algorithm. Even though the KW-variance diversity of RAS is better than Rel-RAS and mRMR-RAS, generally ensemble accuracy of Rel-RAS and mRMR-RAS are better, which may be due to the fact that the individual classifier accuracies are better (RM Characteristic). Besides in order to express the relationship between classification accuracy and low order diversity, 3-way and more diversity should be used.

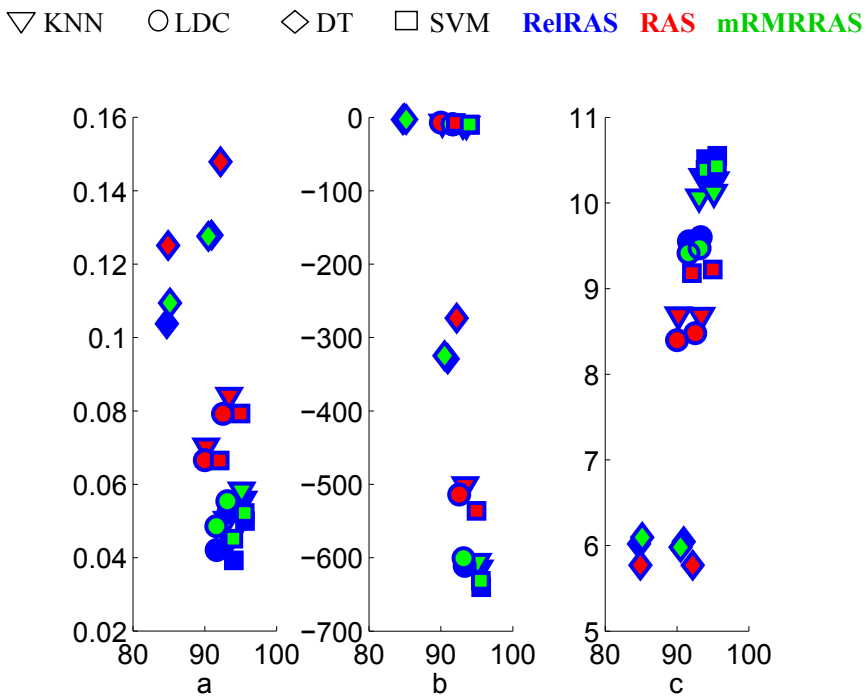


Figure 3.15: Classification accuracy versus diversity on Optdigits dataset obtained by mRMR-RAS, Rel-RAS and RAS for $\mu = 0.3$, $K = 5, 25$ and $m = 25$ a)KW-variance b) LOD c) ITS.

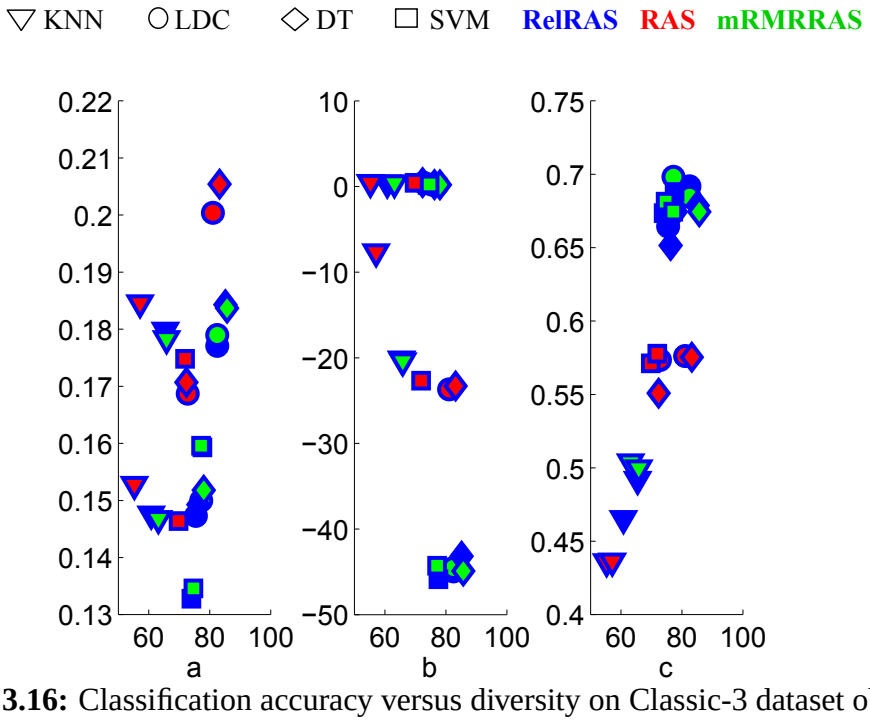


Figure 3.16: Classification accuracy versus diversity on Classic-3 dataset obtained by mRMR-RAS, Rel-RAS and RAS for $\mu = 0.3$, $K = 5, 25$ and $m = 25$ a)KW-variance b) LOD c) ITS.

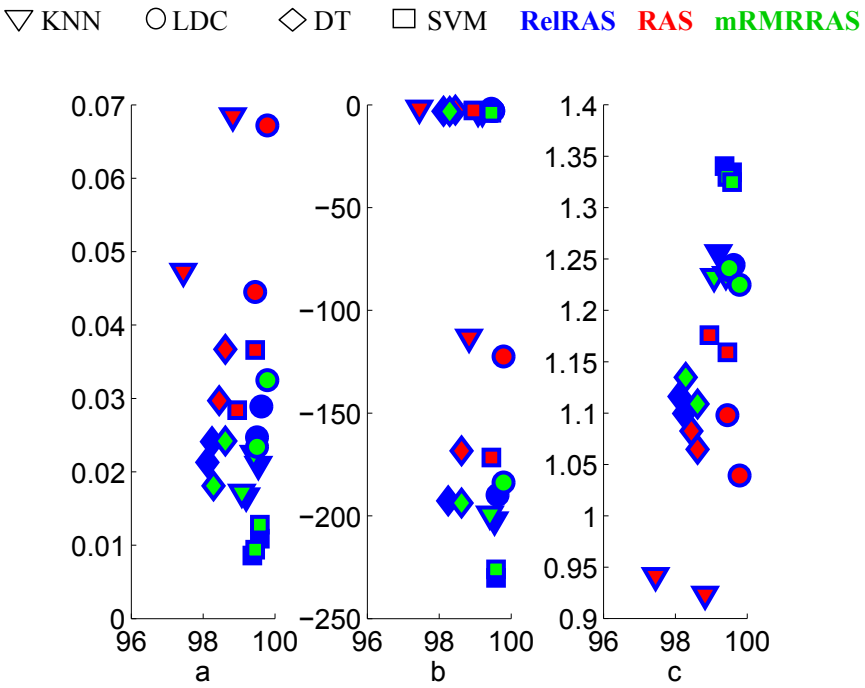


Figure 3.17: Classification accuracy versus diversity on Isolated dataset obtained by mRMR-RAS, Rel-RAS and RAS for $\mu = 0.3$, $K = 5,25$ and $m = 25$ a)KW-variance b) LOD c) ITS.

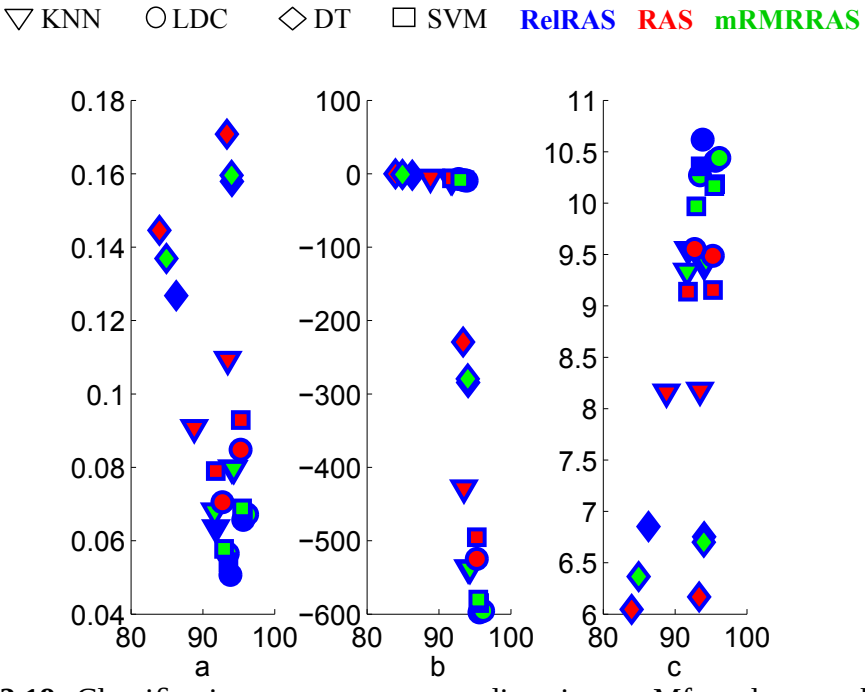


Figure 3.18: Classification accuracy versus diversity on Mfeat dataset obtained by mRMR-RAS, Rel-RAS and RAS for $\mu = 0.3$, $K = 5,25$ and $m = 25$ a)KW-variance b) LOD c) ITS.

3.7 Discussion

In this chapter, the Rel-RAS and mRMR-RAS algorithms which use more informative feature subspaces for classifier ensembles are introduced. The classification accuracies of RAS, Rel-RAS, mRMR-RAS and single classifiers are compared on 5 real datasets: Audio Genre, Optdigits, Classic-3, Isolated letter speech, Mfeat and one synthetic dataset. Besides feature space of the Audio Genre dataset is increased with different powers of the original features in order to obtain redundant features and classification accuracies of the algorithms are also evaluated using these features. In the experiments KNN, LDC, decision tree and SVM are used as base classifier.

Experimental results on real datasets show that the proposed algorithms generally outperform both RAS and single classifier when KNN and LDC classifiers are used. Except for the Classic-3 dataset ensemble algorithms give good classification accuracies when KNN classifier is used. KNN classifier uses the Euclidian distance to find the nearest neighbours. The computed distances are affected from the sparsity of the Classic-3 dataset. Therefore on Classic-3 dataset, KNN classifier is less accurate than the other classifiers. Additionally, single LDC classifier only performs well on Optdigits dataset. In the experiments each dataset use different number of training samples and features: Audio Genre dataset has 45 instances and 50 attributes, Optdigits dataset has 435 instances and 64 attributes, Classic-3 dataset has 228 instances and 273 attributes, Isolet dataset has 38 instances and 617 attributes and Mfeat dataset has 160 instances and 649 attributes. We see that except for Optdigits dataset, the number of training instances is less than the number of attributes in the training samples. Therefore single LDC classifiers are affected from small sample size problem on real datasets. On the other hand, when decision tree is used the proposed algorithms perform better than RAS and single classifier on Classic-3 and Mfeat datasets. Generally decision tree classifier performs worse than the other classifiers except for the sparse Classic-3 dataset. One possible reason of these results is that decision tree uses attributes to distinguish the instances. However the other algorithms use the instances to determine the classification boundaries. The pruning in the decision tree can potentially collapse the leaves that belong to minority classes. Therefore the classification performance may degrade depending to the confidence factor of the pruning. However in the experimental results we didn't change the

pruning parameter and we used J.48 with it's default pruning parameter. When SVM classifier is used, the proposed algorithms perform better than RAS and single SVM on Audio Genre, Optdigits and Isolated letter speech dataset. On the other datasets single SVM performs better than ensemble learning algorithms. Previously Sun and Zhang in [70] showed that single SVM classifier performed better than RAS in 3 of the different 6 datasets. Also they found that single classifier performed worse than RAS algorithm on all datasets when 1-NN and decision trees are used. Similarly in [71], single SVM performed better than RAS algorithm for EEG signal classification in 5 of the 9 datasets. When synthetic and Audio Genre dataset's feature spaces are increased with different powers of the original features we found that the proposed algorithms outperform both single SVM and RAS algorithm. Bertoni et. al. [64] found that different number of m may lead to different results and for low dimension of feature space ensemble may perform less than single SVM. We also analyzed the ensemble algorithms on Classic-3 dataset where single SVM outperforms the Rel-RAS, mRMR-RAS and RAS algorithms. The classification accuracies of the Rel-RAS, mRMR-RAS, RAS and single SVM classifier versus m are given in Figure 3.19. As stated in [64], we also show that increase in the number of selected features, m, also increases the classification accuracy. The proposed algorithms outperform both RAS and single SVM when $m > 50$ on classic-3 dataset.

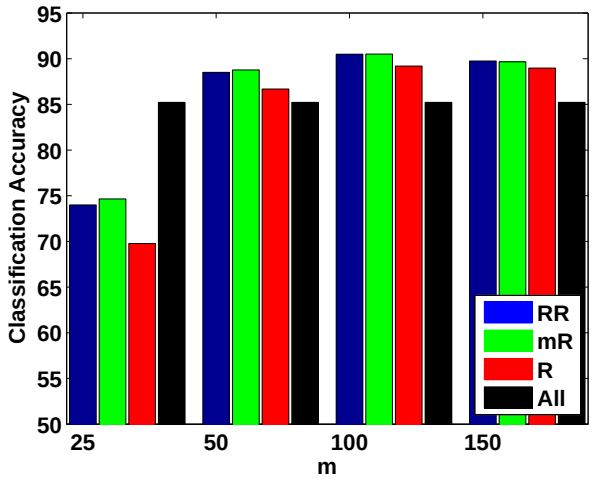


Figure 3.19: Classification accuracy versus m on Classic-3 dataset obtained by Rel-RAS (RR), mRMR-RAS (mR), RAS (R) and single classifier using All features (All) for $\mu = 0.3$, $K = 25$ and Classifier = SVM.

Note that in the ensemble algorithms m needs to be chosen so that individual classifiers have accuracies of more than 50 % For a certain value of m, (where average classifier

accuracy is more than 50 %) as K increases up to a certain value of K, say K^*, the ensemble accuracy increases. It stabilizes for values of K larger than K^*.

In the experiments KW-variance, low order diversity and ITS of the algorithms against classification accuracies are also analyzed. From the experiments we found that KW-variance and low order diversity have the same tendency. Classifiers combined in the proposed algorithms more agree on class labels of test data than RAS algorithm. Then the Rel-RAS and mRMR-RAS algorithms are less diverse than RAS algorithms in terms of KW-variance and low order diversity. On the other hand, ITS is also found to be useful to explain the classification performance of the ensemble algorithms and the proposed methods are generally shown to have higher ITS than RAS.

4. SEMI-SUPERVISED LEARNING USING INFORMATIVE FEATURE SUBSPACES

Unlabeled data have become abundant in many different fields ranging from bioinformatics to web mining, where obtaining the inputs for data points is cheap; however, labeling them is time, money and effort consuming. For example, in speech recognition, recording huge amount of audio does not cost a lot. However, labeling it requires someone to listen and type. Similarly, billions of web pages can be obtained from web servers. However, classifying these web pages into classes is a time consuming and difficult task. Similar situations are valid for remote sensing, face recognition, medical imaging, content based image retrieval [72] and intrusion detection in computer networks [13].

With the availability of unlabeled data and difficulty of obtaining labels, semi-supervised learning methods have gained great importance. On the other hand, in some applications data samples obtained from various sources may be represented in different multiple ways (or views), for example, web pages can be represented using both text information from the web page and text information from the other linked web pages [15]. Generally, when multiple feature views are available, they are concatenated to form the whole feature space. However, this may sometimes be problematic, because the concatenated features may lack physical meaning or may have redundancies [16]. These different views can also be used for training multiple classifiers. Co-training algorithm [15] is an iterative algorithm, proposed to train classifiers on different feature splits and it aims to achieve better classification error by producing classifiers that compensate for each others' classification error. Under certain assumptions, starting with a weak classifier, Co-training algorithm can learn from unlabeled data. The first assumption, *compatibility*, means that the target function over each feature set predicts the same label. The second assumption is, given the class of the instance, the feature sets are *conditionally independent* [15]. It is,

however, difficult for real datasets to satisfy compatibility and especially conditional independence.

Recently, a multi-view Co-training algorithm, RASCO (Random Subspace Method for Co-training) [23], which obtains different feature splits using random subspace method was proposed and shown to result in smaller errors than the traditional Co-training and Tri-training [22] algorithm. RASCO uses random feature splits in order to train different classifiers. The unlabeled data samples are labeled and added to the training set based on the combination of decisions of the classifiers trained on different feature splits. However, if there are many irrelevant features, RASCO may often end up choosing subspaces of features not suitable for good classification.

Instead of totally random feature subspaces, we propose to use Rel-RAS (Relevant Random Subspaces) and mRMR-RAS (minimum Redundancy and Maximum Relevance Random Subspaces) algorithms for Co-training. These algorithms will be detailed in this section. The first proposed algorithm, Rel-RASCO (Relevant Random Subspaces for Co-training) [73, 74], produces relevant random subspaces using relevance scores of features which are obtained using the mutual information between features and class labels. In order to also maintain randomness, each feature for a subspace is selected based on probabilities proportional to relevance scores of features. The second algorithm, mRMR-RASCO (minimum Redundancy and Maximum Relevance Random Subspaces for Co-training) [73], aims to produce random feature subsets that are relevant and non-redundant as possible. In our applications we modified the mRMR feature selection algorithm to produce relevant and non-redundant subspaces. Experimental results on five real and one synthetic datasets show that the proposed algorithms outperform RASCO and traditional Co-training. The work presented in [75] is related to our work in terms of using relevant feature subspaces instead of random ones. However, they use a genetic algorithm to obtain the relevant feature subspaces and do not consider unlabeled data.

The rest of the section is organized as follows. In Section 4.1 literature summary on Co-training style algorithms are given. In Section 4.2 and Section 4.3 Co-training algorithm and RASCO algorithms are given, respectively. Section 4.4 and Section 4.5 provide the details of the proposed algorithms, Rel-RASCO and mRMR-RASCO,

respectively. In section 4.6 the experimental results obtained on different datasets are provided. Section 4.7 concludes the chapter.

4.1 Related Work

Semi-supervised learning methods use unlabeled data in addition to the labeled data for better classification [10, 12]. According to the feature spaces used, semi-supervised learning (SSL) algorithms can be divided into single-view and multi-view algorithms. One of the most successful single view learning algorithms is the Expectation Maximization algorithm which estimates the parameters of a generative model [76]. On the other hand, Self-Training algorithm trains classifier on a single view and it adds unlabeled data incrementally into the labeled dataset [77]. Single-view SSL approaches use either multiple same or multiple different classification algorithms. Statistical Co-learning [78] and Democratic Co-learning [72] are examples of SSL algorithms which train different classification schemes on single-view and do ensemble. On the other hand Tri-training algorithm [22] and Co-training by Committee [79] are single-view SSL that use multiple same classification schemes. Co-training is one of the most well-known multi-view SSL algorithms [15]. Some of the other multi-view SSL algorithms are Co-EM [77] and RASCO [23].

Co-training algorithm has been shown to be successful [15]. However compatibility and independence are strong assumptions of Co-training and many real datasets can not satisfy these assumptions. Therefore, many extensions of Co-training have been proposed in the literature to remedy this problem. In [77], Co-EM algorithm, which incorporates Expectation Maximization into Co-training, was introduced. Instead of assigning each unlabeled data point to a class, Co-EM assigns them to each class with a probability. At each iteration one classifier assigns weighted class values to be used by the other classifier in the next iteration. Co-EM was shown to be less sensitive to independence of classifiers and performed slightly better than Co-training on a text classification problem.

Yan and Naphade proposed semi-supervised cross feature learning to tackle with the strong assumptions of Co-training [80]. They initially train two classifiers two label unlabeled data. Then another two classifiers are trained on the new labeled dataset for

weighted combination. They also extended their work to multiple views. However the classifiers for ensemble is duplicated with the number of views.

Recently Zhou and Li proposed an ensemble method, Co-Forest, that uses random forests in Co-training paradigm [81]. Co-Forest uses bootstrap sample data from training set and trains random trees. At each iteration each random tree is reconstructed by newly selected examples for its concomitant ensemble. Similarly, in [82] a Co-training algorithm is evaluated by multiple classifiers on bootstrapped training examples. Each classifier is trained on the whole feature space and unlabeled data are exploited using multiple classifier systems. Another similar application, Co-training by Committee, is given by Hady and Schwenker in [79]. Co-training by Committee is evaluated using three successful ensemble learning algorithms: Bagging, Adaboost and random subspace method. The committee, i.e. the classifier ensemble, is constructed by using one of these three algorithms and is named as CoBag (Co-training with Bagging), CoAdaBoost (Co-training with AdaBoost) and CoRSM (Co-training with Random Subspace Method). CoBag and CoAdaBoost algorithms work on single feature view and construct the different classifiers by bootstrapping on the training dataset. J48 decision tree was used as the base classifier and CoAdaBoost was generally shown to perform better than CoBag and CoRSM. Experimental results were obtained on different UCI datasets [83] which at most have 60 features.

It should be noted that extensions of Co-training that require bootstrapping may need a lot of labeled samples in order to be successful. For high dimensional datasets, the classifiers trained on small bootstrapped data samples using single feature view may face the "large p, small n problem" [84] (p is the dimensionality and n is the number of data points).

In [78] supervised learning is enhanced with unlabeled data without assuming two compatible and independent feature views. The only requirement for the proposed Co-training algorithm in [78] is that, each hypothesis partitions the input space into a set of classes with equal sizes. Instead of two different feature views, two different supervised algorithms, ID3 and HOODG, are used on the labeled dataset. During the iterations, each classifier labels the unlabeled data points to be used as a labeled example in the next iteration for the other classifier. When a classifier labels an

example, the algorithm uses a statistical test which may require enough amounts of labeled samples.

Democratic Co-learning [72], extends the work of [78] and instead of multiple views, it uses multiple classifiers. As different learning algorithms have different inductive bias, Democratic Co-learning does not need two independent and redundant feature sets. Statistical confidence interval and majority voting is used to decide on the unlabeled data points that are to be added to the labeled dataset. In [22] Zhou and Li proposed Tri-training algorithm where three classifiers are used for Co-training without requiring sufficient and redundant features. The algorithm trains classifiers on bootstrapped data samples and does not require any feature splits.

Random subspace methods [17] are one of the successful methods used for producing an ensemble of classifiers. RASCO algorithm combines the ideas of Co-training and random subspace methods. Instead of using two feature subspaces, it generates a number of subspaces. The labeled dataset is projected onto those random subspaces and a classifier is trained using each feature projection. The intuition behind this is that each classifier can complement another one. RASCO has been shown to perform better than Co-training and Tri-training methods on three different datasets in [23]. The datasets used in [23] have at most 34 features. Another similar approach to RASCO, that uses support vector machines, was proposed to be used for content based image retrieval [85]. Later the work in [85] was extended by using bagging and random subspace method in the same framework in [86].

In many high dimensional datasets, features could be correlated or there may be irrelevant features. When there are a lot of correlated or irrelevant features RASCO may select these features and performance of each individual classifier may decrease. This drawback can be avoided by selecting features which are more relevant, which implies that a more intelligent selection algorithm than random selection could be used. In the next subsections we give the details of the Rel-RASCO and mRMR-RASCO algorithms that use more intelligent feature subset selection algorithms for Co-training.

In the context of multi-view Co-training, feature selection was also used to reduce the input space dimensionality and make computation faster. In [87], an algorithm that maximizes the independence between two feature sets was used in Co-training with

Radial Basis Function neural networks. Similarly, correlation based feature selection was used in [88]. Also in [89], a wrapper approach with forward feature selection in Co-training for predicting emotions with spoken dialogue data was used and it was shown that if a good set of features are selected, Co-training can be highly effective.

4.2 Co-Training

The Co-training algorithm works on two feature subsets which are referred as views and it is assumed that two different views are available [15]. The overall feature set F is the concatenation of different views (In Co-training there are two feature subsets S_1 and S_2).

We assume that we are given a classification problem with c classes. L_i $(i = 1,2,...,n)$ be the d dimensional ith labeled training sample, $L_i = (x_{i1}, x_{i2}, ..., x_{id})$, in the labeled training dataset L, $L = (L_1, L_2, ..., L_n)$ with n samples. There is also an unlabeled dataset U, $U = (U_{n+1}, U_{n+2}, ..., U_{n+r})$ which consists of inputs only where $U_j = (x_{j1}, x_{j2}, ..., x_{jd})$ and $(j = n+1, n+2, ..., n+r)$.

The Co-training algorithm starts with a set of labeled data L and unlabeled data U. It creates a pool of examples U' by choosing u examples randomly from U. The algorithm iterates a specified number of times and does the following: By using L it trains classifiers C_1 and C_2 that use only the S_1 and S_2 portion of the feature space respectively. C_1 and C_2 label examples from U' and select the most surely classified single example from each class. (In [15] the number of added examples for each class depends on the class sizes. We assume that class sizes are similar and a single example for each class is added.) Each classifier adds self-labeled examples to L. Then the algorithm randomly chooses examples from U to replenish U'. Two classifiers, C_1 and C_2, predict class labels for data samples. At each iteration, the samples from U' for which a classifier is sure about that sample above a threshold are selected. This process is continued until the number of data samples in U' are less than a number of data samples threshold. Afterwards the predictions are combined. Most of the previous studies combined the predictions by multiplying their class probability scores together and then renormalizing them. Previously, we proposed to use and adaptive Bayesian classifier combination for Co-training [24] and it performed slightly better than the product combination.

The pseudo code of the Co-training algorithm is given in Algorithm 6.

Algorithm 6 Co-training Algorithm

U' = Select u random examples from U
for i = 1 to I **do**
  for j = 1 to 2 **do**
    Project L to $\hat{L}^j$ using S_j
    Train classifier C_j using $\hat{L}^j$
    Classify U' by C_j
    Select the most surely classified example on U'
    Remove this example from U' and add to L
  end for
end for
Combine C_1 and C_2

4.3 Random Subspaces for Co-training (RASCO)

Random subspace method for Co-training is an iterative semi-supervised classification scheme that uses ensembles of classifiers constructed on randomly generated feature subspaces. It was proposed by Wang et al in [23] and was also used by Hady et al [79] and compared with CoBag, CoAdaBoost algorithms that use bootstrapped data. The RASCO algorithm is inspired from the random subspaces given by Ho [17], in which decision trees are constructed on the feature subsets selected randomly. RASCO algorithm uses the RAS algorithm in semi-supervised learning framework.

Let d be the dimension of original feature space and m be the dimension of each feature subset. The algorithm selects K random subspaces each with m features. A classifier C_k is trained on the labeled training set $\hat{L}_k$ obtained from random selected subset S_k. Then unlabeled dataset U is labeled by majority voting of the classifiers. For each class one most surely classified example from unlabeled data is added to the L. The algorithm terminates after a number of iterations. The pseudo-code of the RASCO algorithm is given in algorithm 7.

As stated previously in RAS algorithm, if there are many irrelevant or correlated features in the dataset RASCO also may select these features and performance of each individual classifier may decrease. For supervised case we proposed to use Rel-RAS and mRMR-RAS algorithms to remedy this problem. Similarly for semi-supervised

Algorithm 7 RASCO Algorithm

for k = 1 to K **do**
  $S_k \leftarrow$ Rand(m) //Select random subspaces $S_1 ... S_k$
  Project L to $\hat{L}_k$ using S_k
  Train classifier C_k using $\hat{L}_k$
end for
//Combine classifiers:
//Define C_k as $d_{k,j} \in \{0,1\}$
for i = 1 to I **do**
  //Combine classifiers by majority voting:
  C_E = MajorityVote(C_1,...,C_K):
  Label examples on U by using C_E
  Select one most surely classified example from U for each class, add them to L.
end for

case we propose the Rel-RASCO and mRMR-RASCO algorithms to remedy this problem.

4.4 Relevant Random Subspace Method for Co-training (Rel-RASCO)

Rel-RASCO algorithm uses the same subspace selection method with Rel-RAS algorithm [74] given in Section 3.3. When producing each feature subspace, Rel-RASCO selects each feature based on its relevance score which is obtained using mutual information between the feature and the class labels.

We create K subspaces $S_1, \ldots, S_K$, each containing $m > 0$ features using the relevance values between features and class labels. Similar to RASCO, in Rel-RASCO also, a classifier is trained on each one of the feature subspaces $S_1, \ldots, S_K$ and the final classifier is obtained by majority voting. At each iteration of Co-training, one most surely classified example from U for each class is added to L. The Rel-RASCO algorithm is given in Algorithm 8.

4.5 Minimum Redundancy and Maximum Relevance Random Subspace Method for Co-training (mRMR-RASCO)

Rel-RASCO algorithm selects feature subsets using the relevance scores obtained between features and class labels. The redundancy of the features in each feature subset is not concerned. In supervised learning scenario this problem is considered with mRMR-RAS algorithm. In semi-supervised case we also propose mRMR-RASCO

Algorithm 8 Rel-RASCO Algorithm

Discretize(L)
Rel = Relevance(L,l) //Mutual Information between features and labels l
//Select random subspaces $S_1 ... S_k$
for k = 1 to K **do**
  $S_k \leftarrow$ Tournament(Rel,m)
  Project L to $\hat{L}_k$ using S_k
  Train classifier C_k using $\hat{L}_k$
end for
//Combine classifiers:
//Define C_k as $d_{k,j} \in \{0,1\}$
for i = 1 to I **do**
  //Combine classifiers by majority voting:
  C_E = MajorityVote(C_1,...,C_K):
  Label examples on U by using C_E
  Select one most surely classified example from U for each class, add them to L.
end for

(minimum Redundancy and Maximum Relevance Random Subspace Method for Co-training) algorithm considers both the relevance and redundancy in each feature subset. mRMR-RASCO algorithm uses the same method with mRMR-RAS algorithm for subset generation.

mRMR-RAS uses, W, redundancy between features in a subset and, V, relevance between features and class labels. In mRMR-RASCO, the first feature is selected using the Relevance, V, as a probability distribution. Then redundancy scores, W, are calculated and $V - W$ are used as the probability of selecting the next feature. Detailed description of the subspace selection in mRMR-RAS is given in Section 3.4. By adding randomness we are able to create diverse, relevant and non-redundant feature subsets, therefore Co-training has diverse enough and accurate classifiers. K subspaces $S_1, \ldots, S_K$, each containing $m > 0$ features are generated using the relevance and redundancy sores. Similar to RASCO and Rel-RASCO, in mRMR-RASCO also, a classifier is trained on each one of the feature subspaces $S_1, ..., S_K$ and the final classifier is obtained by majority voting. At each iteration of Co-training, one most surely classified example from U for each class is added to L. Pseudo code of the proposed algorithm is given in Algorithm 9.

Algorithm 9 mRMR-RASCO Algorithm

Discretize(L)
V = Relevance(L,l) //Mutual Information between features and labels l
W = Redundancy(L) // Mutual Information between features
//Select random subspaces $S_1 \ldots S_k$
for k = 1 to K **do**
  for i = 1 to m **do**
    if i = =1 **then**
      $S_k(i) \leftarrow$ Tournament(V,1)
    else
      $S_k(i) \leftarrow$ Tournament($V - W$,1)
    end if
  end for
  Project L to $\hat{L}_k$ using S_k
  Train classifier C_k using $\hat{L}_k$
end for
//Combine classifiers:
//Define C_k as $d_{k,j} \in \{0,1\}$
for i = 1 to I **do**
  //Combine classifiers by majority voting:
  C_E = MajorityVote(C_1,...,C_K):
  Label examples on U by using C_E
  Select one most surely classified example from U for each class, add them to L.
end for

4.6 Experimental Results

In this section, we present the experimental results comparing performances of Rel-RASCO, mRMR-RASCO, RASCO and Co-training. First, results on 5 different real datasets: Audio Genre, Optdigits, Classic-3, Isolated Letter Speech (Isolet) and MFeat are presented. Then results on Audio Genre and synthetic datasets appended with different redundant features are given. Detailed descriptions about the datasets are given in Appendix C. Besides classifier diversity and information theoretic analysis of the algorithms are also presented.

4.6.1 Real data results

Experimental results are obtained on 5 different datasets: 'Optdigits' (Optical Recognition of Handwritten Digits), 'MFeat' (Multiple Features) and 'Isolet' (Isolated Letter Speech) datasets from the UCI machine learning repository [83], 'Classic-3' text

dataset from [90] and the 'Audio Genre' dataset of [91]. Table C.1 shows the number of features, instances and classes of the features for all 5 datasets.

For each dataset, experimental results for Rel-RASCO, mRMR-RASCO and RASCO are obtained on 10 different random runs. At each random run, the whole dataset is splitted equally into a training partition and a test partition. Training set is further splitted into unlabeled training set and μ portion of the rest of the training data is used as the labeled training set.

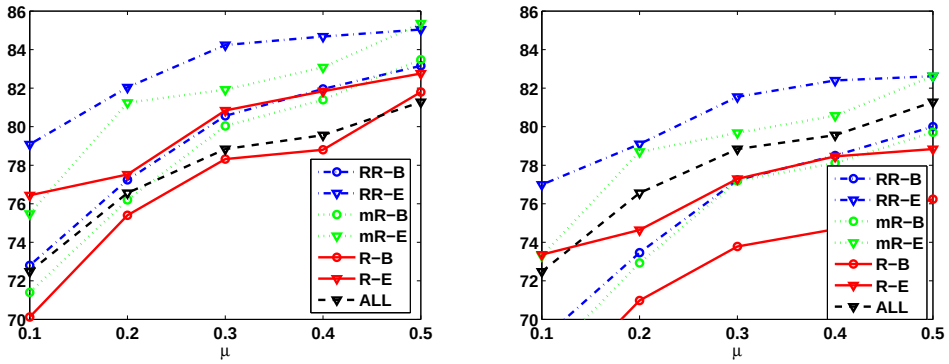
In supervised learning experiments, we see that increase the number of training dataset (μ parameter) also increases the classification accuracy. In semi-supervised learning experiments we also did the same experiments by increasing the number of training samples in the dataset. Experiments are reported for different number of subspaces, K = 5, 25.

In the figures RelRASCO-B, RASCO-B, mRMR-RASCO-B and RelRASCO-E, RASCO-E, mRMR-RASCO-E represent the Rel-RASCO, RASCO and mRMR-RASCO results at the beginning and end of Co-training respectively. First we report the averages of the ensemble accuracies and averages of the individual classifier accuracies of Audio Genre dataset with respect to μ. Standard errors of the results depend to the base classifier used. However they are generally around 2% and in order to keep the figures readable standard error bars are not given. On the other hand, unlabeled data degrade the classification accuracies of self-training when $\mu = 0.3$. Therefore they are not given in the figures. Co-training results are less than RASCO and the other algorithms therefore we don't give them in the figures.

Audio Genre Dataset: The 5 least confused genres of Tzanetakis dataset [91], Classical, Hiphop, Jazz, Pop and Reggae, each with 100 samples, are used. Two different sets of audio features are computed. First, timbral, rhytmic content and pitch content features yielding 30 features are extracted using the Marsyas Toolbox [91]. Next, 20 features covering temporal and spectral properties are extracted using the Databionic Music Miner framework [92].

Mean ensemble classification accuracies and mean individual classification accuracies at the beginning and end of Co-training with respect to different values of μ for KNN classifier are given in Figure 4.1(a) and Figure 4.1(b) respectively. In the figures, RR-B

and RR-E represent the classification accuracy of Rel-RASCO at the beginning and Rel-RASCO at the end of Co-training, respectively. Similarly mR-B, mR-E, R-B and R-E represent the classification accuracies of the mRMR-RASCO (mR) and RASCO (R) at the beginning (B) and at the end (E) of the Co-training. ALL represents the single classifier performance on supervised learning. We see that the proposed algorithms outperform both RASCO and single classifier. Increase in the μ also increases the classification accuracies of the algorithms. Note that, ensemble algorithms benefit from unlabeled data and they perform better than the individual classifiers.

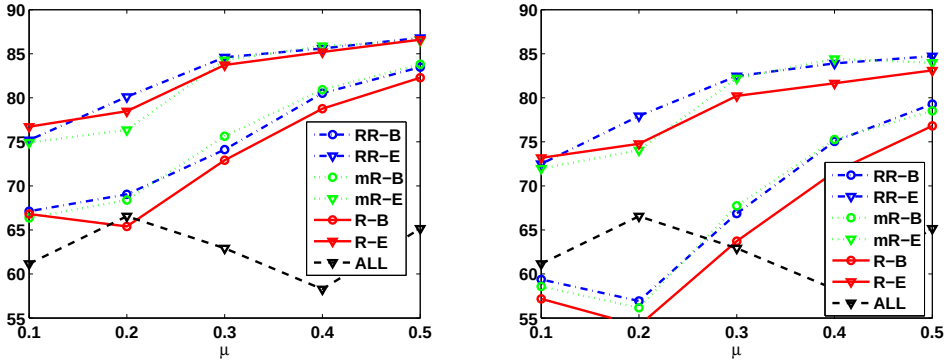


(a) Ensemble, KNN (b) Individual, KNN

Figure 4.1: Mean ensemble and individual test accuracies on Audio Genre dataset obtained by mRMR-RASCO, Rel-RASCO and RASCO with respect to μ for $m = 25$, classifier = KNN.

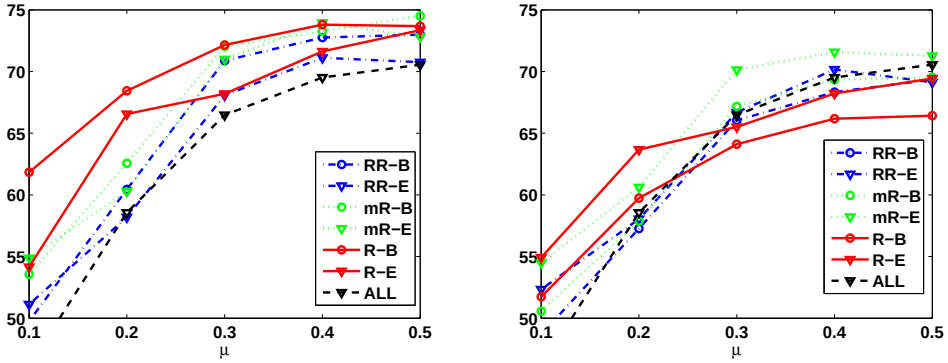
In Figure 4.2(a) and Figure 4.2(b) mean ensemble and mean individual classification performances on Audio Genre dataset with LDC classifier are given, respectively. At the beginning of the Co-training the proposed algorithms perform better than RASCO. On the other hand at the end of Co-training the proposed algorithms perform slightly better than RASCO. Note that, Figure 4.2(b) shows that the proposed algorithms are more RM characteristic than RASCO.

In Figure 4.3(a) and Figure 4.4(a) mean ensemble classification accuracies with decision tree and SVM classifiers are given. RASCO performs better than the proposed algorithms when decision tree is used. The proposed algorithms select more relevant features than random selection. Therefore similar features may be used during tree production and proposed methods may perform less than the RASCO. However when SVM classifier is used, the Rel-RASCO and mRMR-RASCO perform better than RASCO and single classifier.



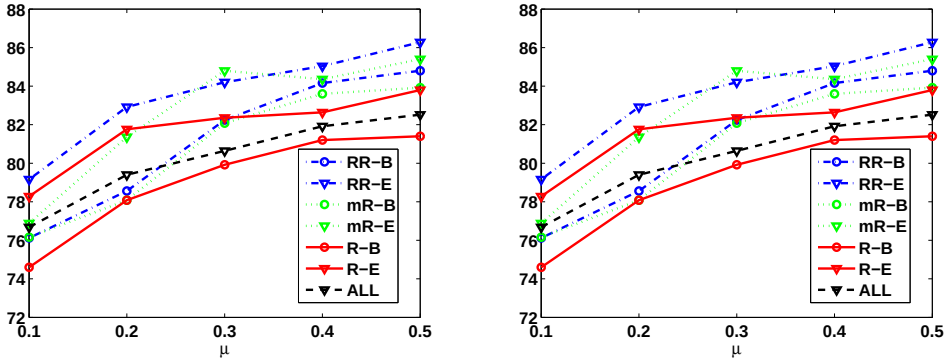
(a) Ensemble, LDC (b) Individual, LDC

Figure 4.2: Mean ensemble and individual test accuracies on Audio Genre dataset obtained by mRMR-RASCO, Rel-RASCO and RASCO with respect to *mu* for $m = 25$, classifier = LDC.



(a) Ensemble, J48 (b) Individual, J48

Figure 4.3: Mean ensemble and individual test accuracies on Audio Genre dataset obtained by mRMR-RASCO, Rel-RASCO and RASCO with respect to *mu* for $m = 25$, classifier = J48.



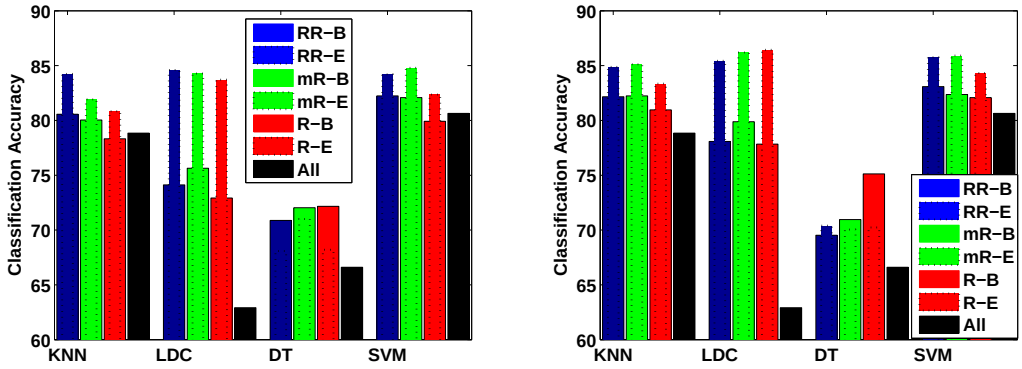
(a) Ensemble, SVM (b) Individual, SVM

Figure 4.4: Mean ensemble and individual test accuracies on Audio Genre dataset obtained by mRMR-RASCO, Rel-RASCO and RASCO with respect to *mu* for $m = 25$, classifier = SVM.

Supervised learning experimental results showed that increase in the number of training samples increases the ensemble accuracy for all algorithms and the proposed algorithms outperform single classifiers and RAS algorithm. Similar results are also obtained on semi-supervised learning case. The classification accuracies of

the algorithms are evaluated with small number of instances and small number of classifiers with fixing $\mu = 0.3$ and K = 5, 25.

In Figure 4.5 mean ensemble classification accuracies of Audio Genre dataset at the beginning and at the end of the Co-training with different classifiers are given for K= 5 and 25, respectively. We see that Rel-RASCO and mRMR-RASCO perform better than RASCO at the beginning and at the end of Co-training. On the other hand when KNN, LDC and SVM classifiers are used the algorithms benefit from unlabeled data.

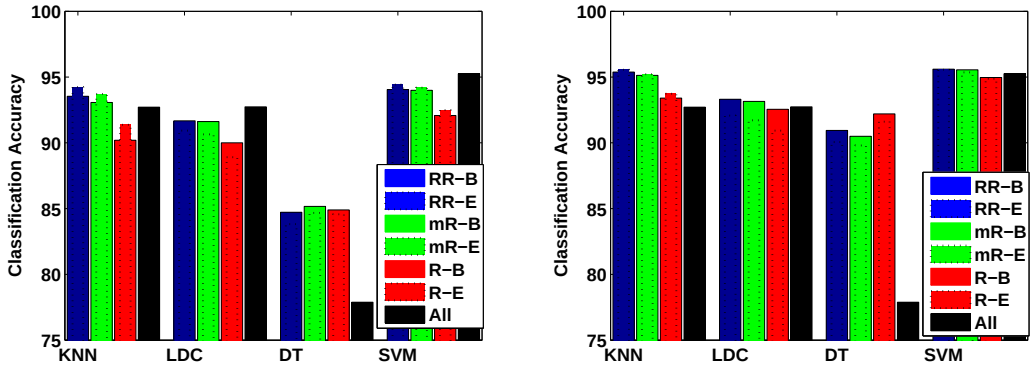


(a) $K = 5$ (b) $K = 25$

Figure 4.5: Mean ensemble test accuracies on Audio Genre dataset, at the beginning (-B) and end (-E) of Co-training, obtained by Rel-RASCO (RR), mRMR-RASCO (mR), RASCO (R) and single classifier using all features (All) for $m = 25$.

UCI Optdigits dataset: The mean ensemble classification accuracies of Optdigists dataset with different classifiers are given for K=5 and 25 classifiers in Figure 4.6. We see that the proposed algorithms benefit from unlabeled data when KNN classifier is used. Semi-supervised ensemble learning algorithms do not benefit from unlabeled data when the LDC and decision tree classifiers are used as base classifier. On the other hand ensemble algorithms benefit from unlabeled data when SVM classifier is used as base classifier for K=5.

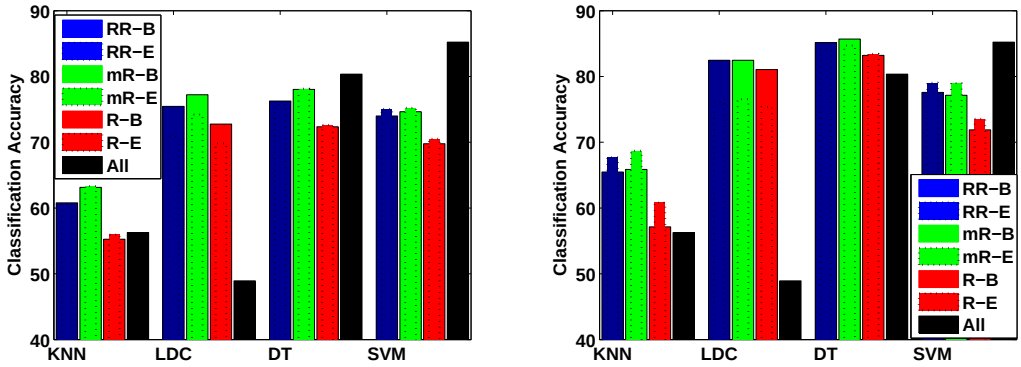
Classic-3 dataset: Term Frequencies of words are used as features and they are obtained using Term-to-Matrix generator (TMG) Matlab Toolbox [93]. Mean ensemble classification accuracies of Classic-3 dataset for different classifiers are given in Figure 4.7. We see that single SVM and single decision tree performs better than the ensemble methods when K = 5. When decision tree is used as base classifier for K = 25 the proposed algorithms perform better than RASCO and single classifier.



(a) $K = 5$ (b) $K = 25$

Figure 4.6: Mean ensemble test accuracies on Optdigits dataset, at the beginning (-B) and end (-E) of Co-training, obtained by Rel-RASCO (RR), mRMR-RASCO (mR), RASCO (R) and single classifier using all features (All) for $m = 25$.

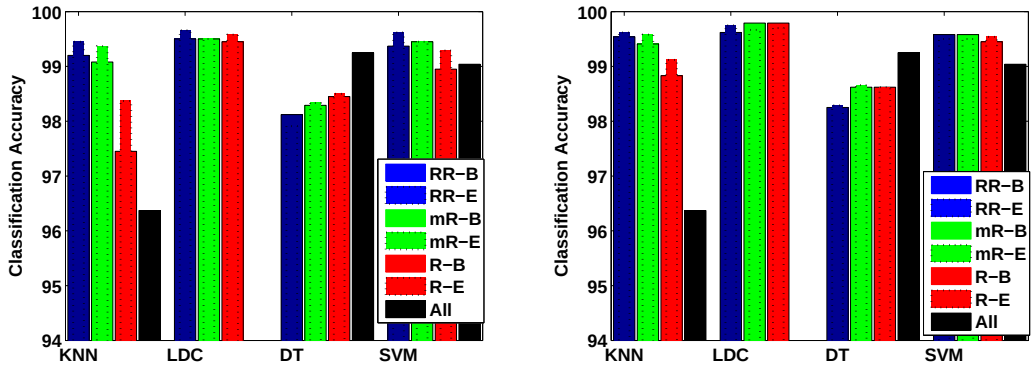
UCI Isolated Letter Speech dataset: A high dimensional dataset with 617 features and 480 instances from B and C letters are used in this experiment. In Figure 4.8 the mean ensemble classification accuracies of Isolet dataset with different classifiers are given for K= 5 and 25. When KNN and LDC are used the algorithms may benefit from unlabeled data. On the other hand the proposed algorithms generally perform better than the RASCO and single classifier.



(a) $K = 5$ (b) $K = 25$

Figure 4.7: Mean ensemble test accuracies on Classic-3 dataset, at the beginning (-B) and end (-E) of Co-training, obtained by Rel-RASCO (RR), mRMR-RASCO (mR), RASCO (R) and single classifier using all features (All) for $m = 25$.

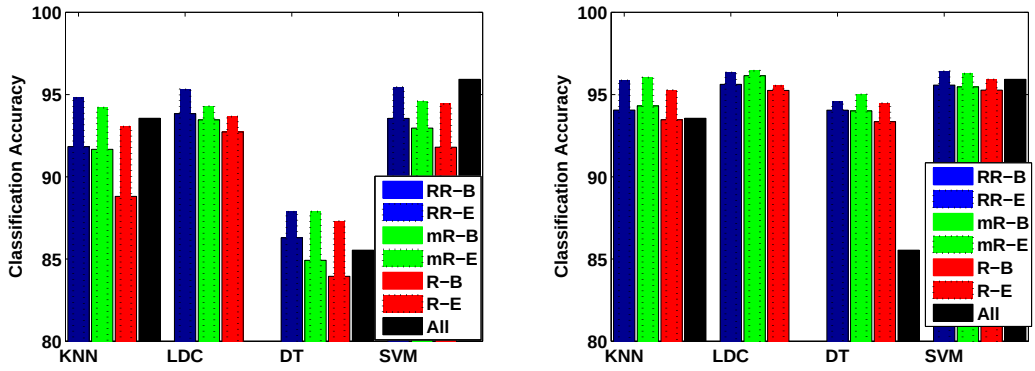
MFeat dataset: Mfeat dataset is also a high dimensional dataset with 649 features. In Figure 4.9 the mean ensemble classification accuracies of Mfeat dataset with different classifiers are given for K= 5 and 25. We see that algorithms benefit from unlabeled data and the best classification accuracy at the end of the Co-training is obtained with Rel-RASCO algorithm using SVM classifier.



(a) $K = 5$ (b) $K = 25$

Figure 4.8: Mean ensemble test accuracies on Isolated Letter Speech dataset, at the beginning (-B) and end (-E) of Co-training, obtained by Rel-RASCO (RR), mRMR-RASCO (mR), RASCO (R) and single classifier using all features (All) for $m = 25$.

We also obtained the p values using t-test for the 10-fold cross validation accuracies of RASCO, Rel-RASCO and mRMR-RASCO algorithms at the beginning and end of Co-training when $K = 25$ subsets are used (Table 4.1). Details of the t-test are given in Appendix H. According to Table 4.1, when KNN classifier is used with 90% probability, at the end of Co-training Rel-RASCO ensemble accuracy is better than that of RASCO. We think that the performance increase obtained by Rel-RASCO is related to a number of factors, including the number of features in the dataset, their average relevance, the number of samples available and also the size and number of feature subspaces used.



(a) $K = 5$ (b) $K = 25$

Figure 4.9: Mean ensemble test accuracies on Mfeat dataset, at the beginning (-B) and end (-E) of Co-training, obtained by Rel-RASCO (RR), mRMR-RASCO (mR), RASCO (R) and single classifier using all features (All) for $m = 25$.

When there are many features as in Mfeat and Isolet or the features are not so relevant as in Classic-3 and Optdigits, Rel-RASCO has advantage over RASCO. Rel-RASCO's performance is significantly better than RASCO's performance at the

end of Co-training with SVM classifier except for Isolet dataset. However LDC and decision tree results are not as significant as SVM results at the end of Co-training.

Table 4.1: t-test *p* values of RASCO and Rel-RASCO at the beginning and at the end of the algorithms for each dataset, K=25, m=25.

Classifier	audio	optdigits	classic-3	Isolet	mfeat
KNN (Beg)	0.15	0.00	0.00	0.01	0.15
KNN (End)	0.06	0.00	0.00	0.02	0.05
LDC (Beg)	0.85	0.01	0.32	0.18	0.31
LDC (End)	0.31	0.00	0.59	0.77	0.01
J48 (Beg)	0.00	0.00	0.22	0.31	0.18
J48 (End)	0.94	0.00	0.56	0.27	0.9
SVM (Beg)	0.31	0.00	0.00	0.48	0.35
SVM (End)	0.063	0.00	0.00	0.79	0.04

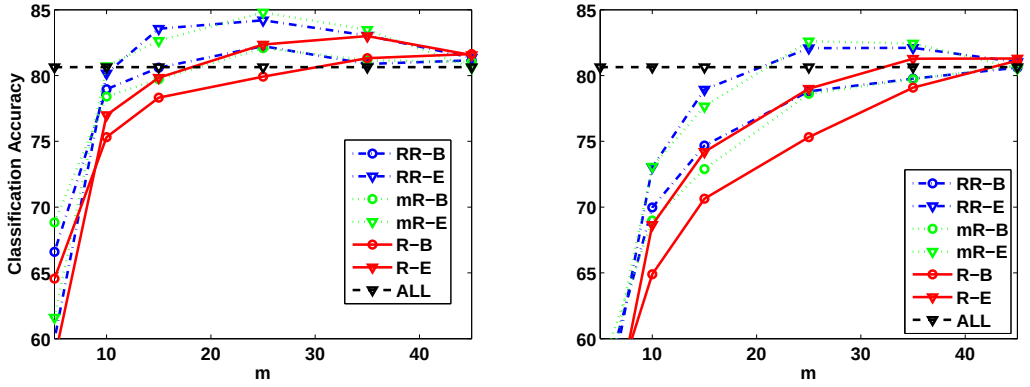
In Table 4.2 *p* values for the 10-fold cross validation accuracies of RASCO and mRMR-RASCO algorithms at the beginning and end of Co-training are given. We see the similar results obtained between RASCO and Rel-RASCO. With 90% probability, at the end of Co-training mRMR-RASCO ensemble accuracy is better than that of RASCO when KNN and SVM classifiers are used. On the other hand LDC and decision tree results are not as significant as KNN and SVM results at the end of Co-training.

Table 4.2: t-test *p* values of RASCO and mRMR-RASCO at the beginning and at the end of the algorithms for each dataset, K=25, m=25.

Classifier	audio	optdigits	classic-3	isolet	mfeat
KNN (Beg)	0.15	0.00	0.00	0.05	0.03
KNN (End)	0.08	0.00	0.00	0.05	0.01
LDC (Beg)	0.15	0.02	0.34	1	0.00
LDC (End)	0.82	0.01	0.46	0.76	0.00
J48 (Beg)	0.04	0.00	0.13	1	0.00
J48 (End)	0.91	0.00	0.3	0.87	0.19
SVM (Beg)	0.78	0.02	0.00	0.533	0.6
SVM (End)	0.02	0.01	0.00	0.85	0.18

Next the effect of the parameter *m*, which is the number of features selected, is evaluated on the Audio Genre dataset.In Section 3.7 we have shown that increase in the *m* also increases the classification accuracy. In Figure 4.10 Audio Genre dataset accuracies with SVM classifier are given for K=5 and μ = 0.3. Figure 4.10(a) shows the ensemble classification accuracy with respect to *m*. Rel-RASCO and mRMR-RASCO outperform both RASCO and single classifier when $m > 10$. The best classification

accuracy is obtained when m = 25. Previously [23] has given the best m simply as $m = d/2$ for RASCO. However for high dimensional datasets, increase in the m also increases the complexity of the algorithm. Figures 4.10(b) shows the mean classification accuracies of individual classifiers. Single SVM classifier performs better than the mean individual classification accuracies of the ensemble algorithms when m<10.



(a) Ensemble, μ=0.3 (b) Individual,μ=0.3

Figure 4.10: Mean ensemble and individual classifier test accuracies on Audio Genre dataset at the beginning (-B) and end (-E) of Co-training, obtained by Rel-RASCO (RR), mRMR-RASCO (mR), RASCO (R) and single classifier using all features (All), with respect to m for K=5 and classifier = SVM.

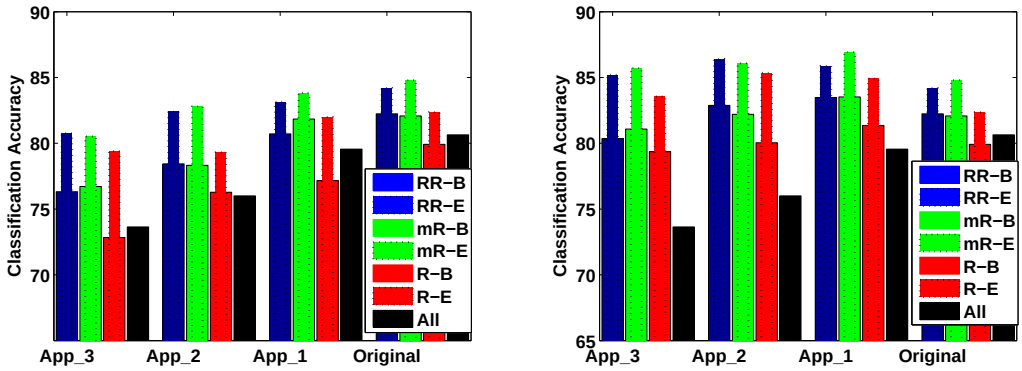
As a general guideline, m should not be too large to overfit the training data and it should not be too small to result in too weak classifiers. As the number of feature subspaces and hence classifiers increase, the same ensemble accuracy can be achieved using smaller size feature subspaces. The number of features used by Rel-RASCO should be at least as much as the number of features that results in a good accuracy when feature selection is performed on all the available data. It is possible to determine this lower bound using a fast feature selection algorithm such as mRMR [94]. The value of m could also be selected using a model selection method such as cross-validation, however this could be a time-intensive task.

4.6.2 Robustness to redundant features

In supervised learning experiments we evaluate the robustness of the algorithms with redundant features and we show that the proposed subspace selection algorithms outperform RAS and single classifier. In order to evaluate algorithms' robustness to redundant features in semi-supervised learning, again Audio Genre dataset's feature

space is appended with different powers of the original features. Real dataset experiments show that proposed algorithms outperform the RASCO with KNN, LDC and decision tree. On the other hand the best classification accuracies are generally obtained with SVM and we see that proposed algorithms with SVM generally perform better than RASCO.

Three datasets, App_1, App_2 and App_3, generated in supervised learning experiments are also used in this experiment (Please see Section 3.6.2 for details of the datasets). In Figure 4.11 the mean classification accuracies obtained on Audio Genre dataset appended with redundant features are given for SVM classifier at the beginning and at the end of the algorithms for $K = 5$ and $K = 25$. It can be seen from the figure that, the proposed algorithms outperform the RASCO and single classifier at the beginning and at end of the algorithms. Besides all of the algorithms benefit from unlabeled data and the proposed algorithms perform better than RASCO algorithm at the end of Co-training and single classifier.



(a) $K = 5$ (b) $K = 25$

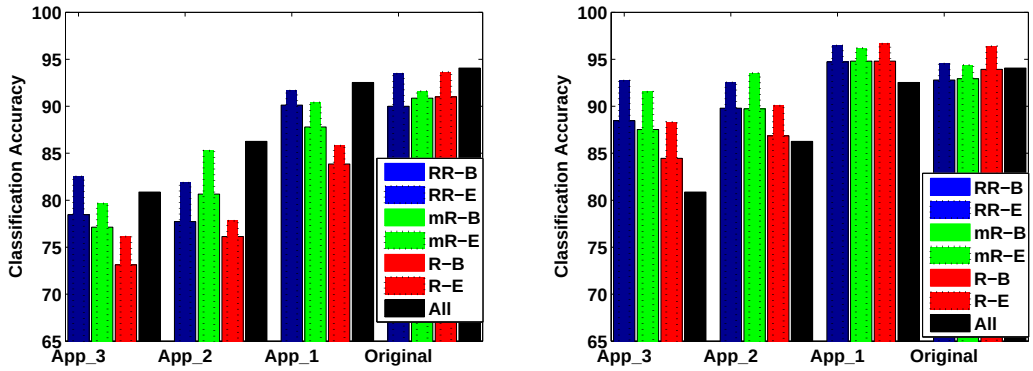
Figure 4.11: Mean ensemble test accuracies on Audio Genre dataset appended with redundant features, at the beginning (-B) and end (-E) of Co-training, obtained by Rel-RASCO (RR), mRMR-RASCO (mR), RASCO (R) and single classifier using all features (All), for $\mu = 0.3$, $m = 25$ and classifier = SVM.

4.6.3 Synthetic data results

Classification accuracies of the RASCO, Rel-RASCO and mRMR-RASCO algorithms are also evaluated with a synthetic two class dataset. The dataset is generated from Gaussian distributions with a covariance matrix 10 at diagonal and mean -1 for one class and 1 for the other class. The total number of features is chosen to be 50. Three synthetic datasets, App_1, App_2 and App_3, generated in supervised learning

experiments are also used in this experiment (Please see Section 3.6.3 for details of the datasets).

The mean classification accuracies at the beginning and at the end of the algorithms obtained using SVM classifier on synthetic dataset appended with redundant features are given for $K = 5$ and $K = 25$ classifiers in Figure 4.12(a) and Figure 4.12(b), respectively.



(a) $K = 5$ (b) $K = 25$

Figure 4.12: Mean ensemble test accuracies on synthetic dataset appended with redundant features, at the beginning (-B) and end (-E) of Co-training, obtained by Rel-RASCO (RR), mRMR-RASCO (mR), RASCO (R) and single classifier using all features (All), for $\mu = 0.3$, $m = 25$, classifier = SVM.

It can be seen from Figure 4.12(a) that the single SVM classifier performs better than the ensemble algorithms for $K = 5$. However the proposed algorithms perform better than the single SVM and RASCO when $K = 25$ classifiers are used as shown in Figure 4.12(b). Note that the original dataset has uncorrelated features. Therefore the single SVM performs slightly better than the ensemble algorithms at the beginning of the Co-training on original synthetic dataset. On the other hand, ensemble algorithms benefit from unlabeled data and we see that the proposed algorithms outperform single SVM and RASCO when the datasets are too redundant (Please see App_2 and App_3 datasets results in Figure 4.12(b)).

4.6.4 Classifier diversity and information theoretic analysis of the algorithms in semi-supervised learning

In Section 3.6.4 classifier diversities and information theoretic analysis of the algorithms have shown that, although the proposed algorithms are less diverse than the RAS algorithm in terms of KW-variance and LOD, they perform better than RAS. On

the other hand we found that the classification accuracy of the ensemble methods can be analyzed with ITS. Similar experiments and analysis given in supervised learning scenarios are also done for semi-supervised learning scenarios.

In figures KW-Variance, LOD and ITS versus classification accuracies are given for all datasets at the end of the Co-training. Note that KW-Variance, LOD and ITS versus classification accuracies at the beginning of the algorithms are given in the previous chapter. Figures are obtained using the classification accuracies and diversities of different number of classifiers (K =5, 25) in the ensembles.

In Figure 4.13 classification accuracy versus diversity analysis on Audio Genre dataset at the end of the Co-training is given. Rel-RASCO and mRMR-RASCO algorithms are less diverse than RASCO algorithm at the end of Co-training in terms of KW-variance. Also at the end of Co-training KW-variances of the algorithms decrease. In figure it is shown that LOD has a similar tendency with KW-variance. On the other hand the proposed algorithms have higher ITS than the RASCO algorithm at the end of Co-training. Note that the proposed algorithms have the best ITS with SVM at the end of Co-training.

In Figure 4.14, Figure 4.15, Figure 4.16 and Figure 4.17 classification accuracy versus diversity analysis on Optdigits, Classic-3, Isolet and Mfeat datasets at the end of the Co-training are given, respectively. Similar results obtained with the Audio Genre dataset are obtained for Optdigits, Classic-3, Isolet and Mfeat datasets and KW-variance and LOD of the proposed algorithms are less than RASCO. Increasing the ITS of the algorithms also increases the classification accuracy and the best ITS at the end of the algorithms are obtained with the proposed algorithms. Generally KW-variances of the algorithms at the end of Co-training are less than the KW-variances at the beginning of Co-training. Even though the KW-variance diversity of RASCO is better than Rel-RASCO and mRMR-RASCO, generally ensemble accuracy of Rel-RASCO and mRMR-RASCO are better, which may be due to the fact that the individual classifier accuracies are better (RM Characteristic of the proposed algorithms). Besides in order to express the relationship between classification accuracy and low order diversity, 3-way and more diversity should be used.

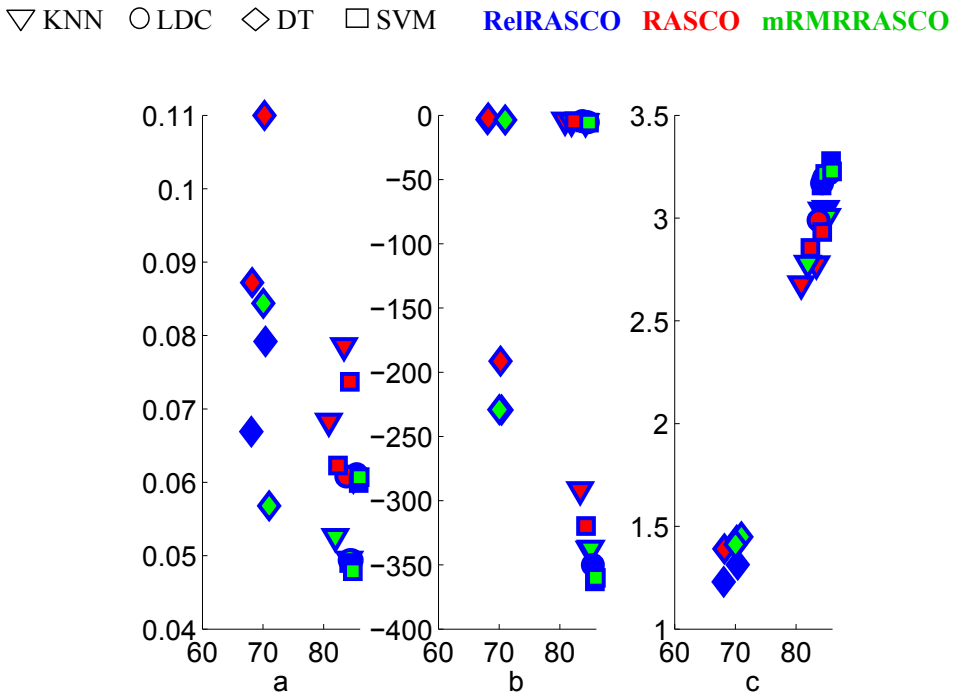


Figure 4.13: Classification accuracy versus diversity on Audio Genre dataset obtained by mRMR-RASCO, Rel-RASCO and RASCO (End of the algorithms) for $\mu = 0.3$, $m = 25$ a)KW-variance b) LOD c) ITS.

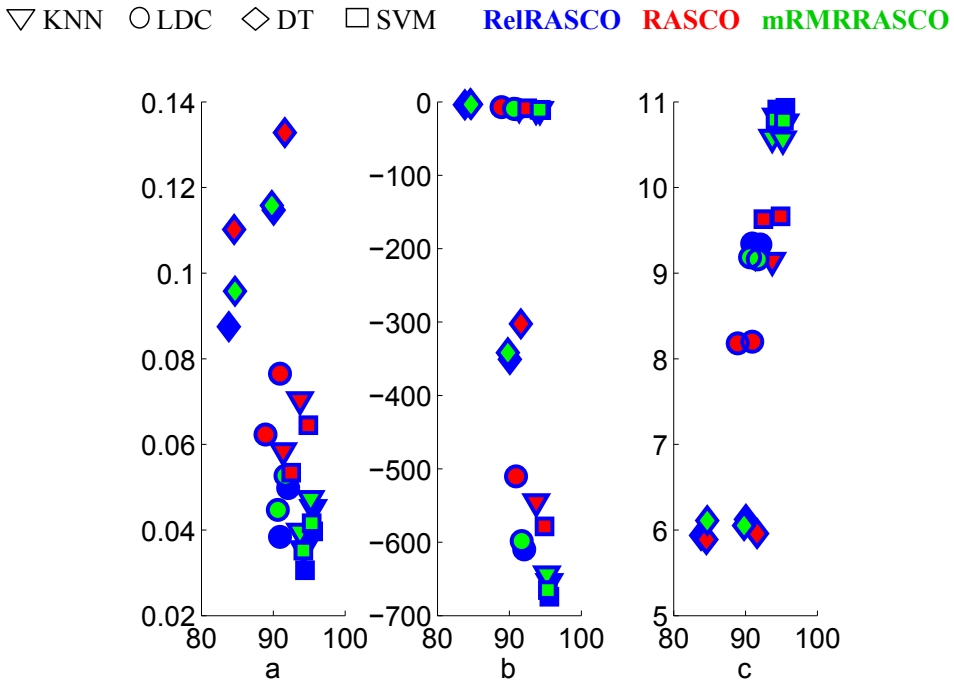


Figure 4.14: Classification accuracy versus diversity on Optdigits dataset obtained by mRMR-RASCO, Rel-RASCO and RASCO (End of the algorithms) for $\mu = 0.3$, $m = 25$ a)KW-variance b) LOD c) ITS.

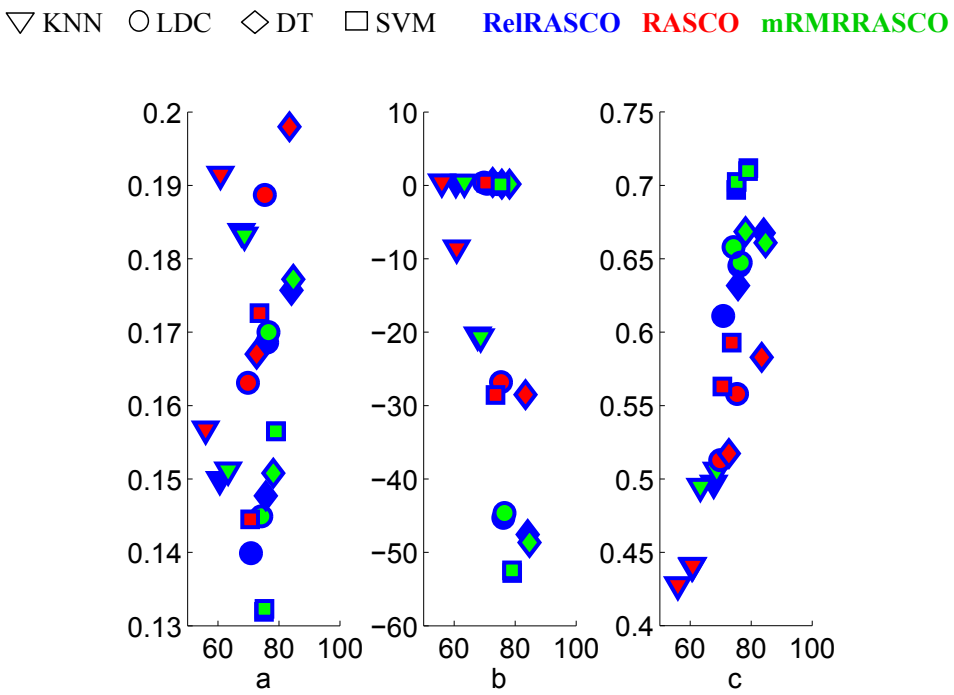


Figure 4.15: Classification accuracy versus diversity on Classic-3 dataset obtained by mRMR-RASCO, Rel-RASCO and RASCO (End of the algorithms) for $\mu = 0.3$, $m = 25$ a)KW-variance b) LOD c) ITS.

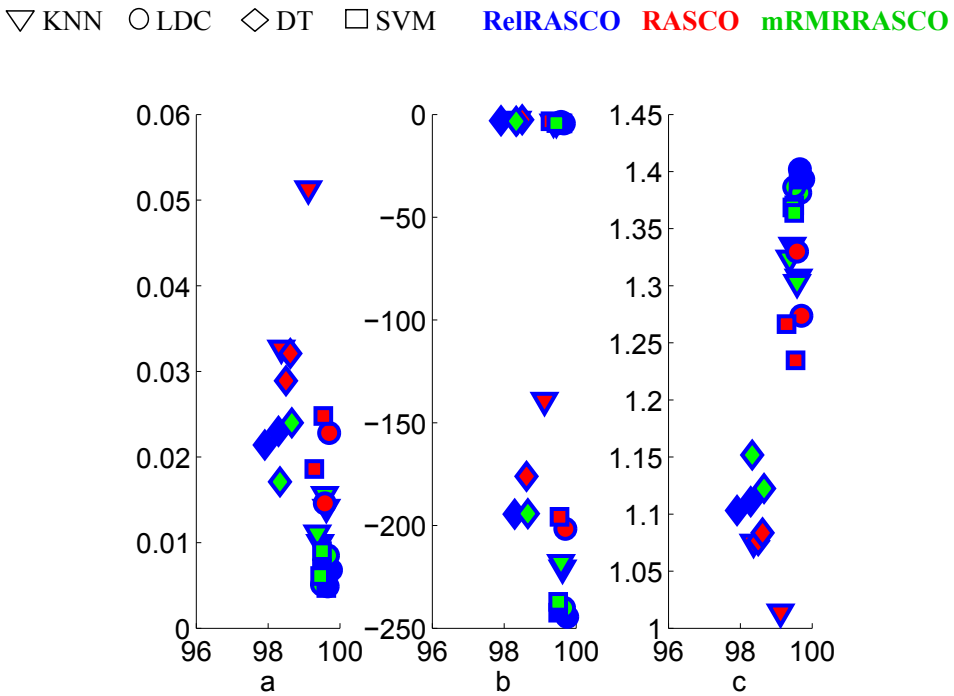


Figure 4.16: Classification accuracy versus diversity on Isolet dataset obtained by mRMR-RASCO, Rel-RASCO and RASCO (End of the algorithms) for $\mu = 0.3$, $m = 25$ a)KW-variance b) LOD c) ITS.

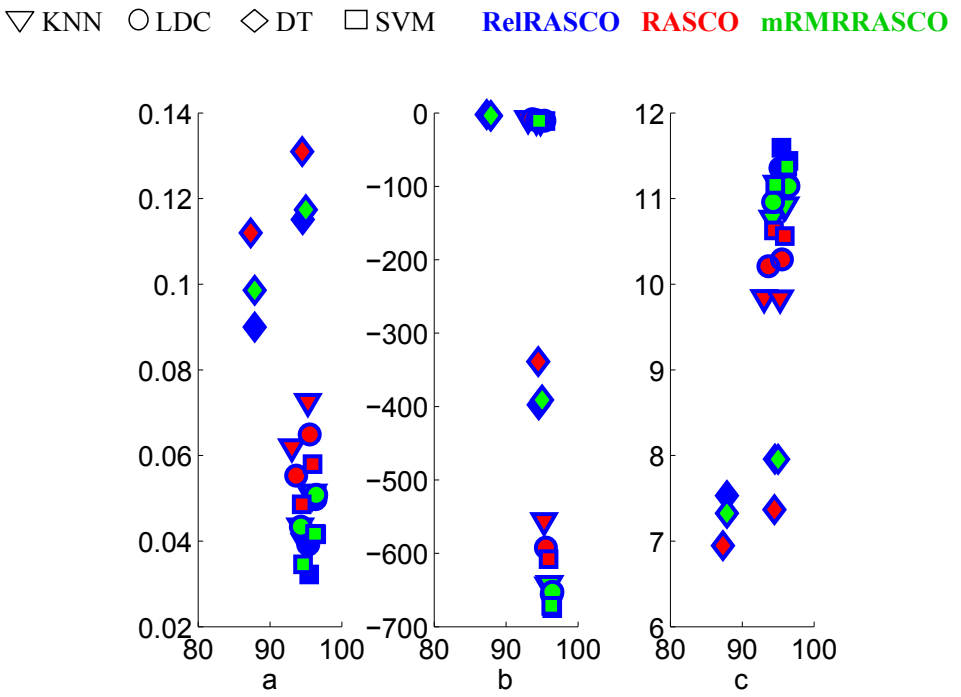


Figure 4.17: Classification accuracy versus diversity on Mfeat dataset obtained by mRMR-RASCO, Rel-RASCO and RASCO (End of the algorithms) for $\mu = 0.3$, $m = 25$ a)KW-variance b) LOD c) ITS.

4.7 Discussion

In this chapter, the Rel-RASCO and mRMR-RASCO algorithms which use more informative feature subspaces for Co-training are introduced. Classification accuracies of RASCO, Rel-RASCO and mRMR-RASCO on 5 real datasets: Audio Genre, Optdigits, Classic-3, Isolated letter speech, Mfeat and one synthetic dataset with redundant features are obtained. Besides classification accuracies of the algorithms on Audio Genre dataset appended with redundant features are also investigated. We showed that, at the beginning of Co-training, before unlabeled data are used, classifier ensembles of the proposed algorithms have in general better accuracies than RASCO. When unlabeled data are labeled iteratively, the ensemble accuracy of Rel-RASCO and mRMR-RASCO are still better than RASCO or single classifier. Generally mRMR-RASCO and Rel-RASCO perform significantly better than RASCO or single classifier when there are many irrelevant features. As the number of classifiers in the ensemble increase, especially at the end of Co-training, the ensemble accuracy of RASCO approaches the ensemble accuracy of Rel-RASCO. Additionally mean individual classification accuracies show that the Rel-RASCO and mRMR-RASCO algorithms are more RM-characteristic than the RASCO algorithm.

The diversity analysis of the algorithms are also obtained for KNN, LDC, decision tree and SVM classifiers with different number of classifiers (K) in the ensemble. KW-variance, LOD and ITS are given for all datasets at the end of the Co-training. Although Rel-RASCO and mRMR-RASCO have less KW-variance and LOD than RASCO algorithm, they generally perform better classification accuracy than RASCO. In the experiments we also found that the KW-variance and LOD decrease at the end of the algorithms. Besides Rel-RASCO and mRMR-RASCO algorithms are shown to have higher ITS than RASCO.

4.7.1 The effect of unlabeled data

Do unlabeled data improve the classification performance? There have been many studies trying to find an answer to this question [95, 96]. Some studies for example [12] showed that unlabeled data may help to increase the classification accuracy. On the other hand Cozman and Cohen [95] showed that unlabeled data can degrade the classification performance if the model assumption and the data distribution do not match. This result was obtained by generative classifiers on an artificial dataset that has dependent features. The percentage of unlabeled data among the training set was fixed and Naive Bayes classifier was used. Besides Tian et. al. [97] showed that if the model assumption does not hold, the performance of unlabeled data is affected by the complexity of the classifier. They considered semi-supervised learning problems where the labeled and unlabeled data do not come from the same distribution and analyzed the effect of unlabeled data on content based image retrieval problem. It is shown that unlabeled data help if both labeled and unlabeled data come from the same distribution. Otherwise depending on the difference between labeled and unlabeled data, more unlabeled data may decrease the performance. In [98] Co-training and EM algorithm degraded the classification performance of text categorization task. Catal and Diri [99] also analyzed unlabeled data effect on software fault prediction problem and they showed that unlabeled data may decrease the performance of software fault prediction problem.

In our experiments, unlabeled data generally increases the classification performance of Rel-RASCO, mRMR-RASCO and RASCO algorithms. In the experiments it is observed that KNN and SVM classifiers always benefit from unlabeled data. On the

other hand depending on the dataset, unlabeled data generally improves classification accuracy of the algorithms with LDC and decision tree classifiers. Performance decrease of the algorithms depends to some factors such as: model assumption, base classifier used in the algorithms and overlearning of the classifiers that makes them to select incorrect examples from unlabeled dataset. LDC and decision tree classifiers on Optdigits dataset do not benefit from unlabeled data. Similarly Classic-3 dataset is too sparse and LDC classifier may not generate accurate model parameters.

5. CONCLUSION AND FUTURE WORK

The advent of the technology enables us to access all kinds of data easily from many fields of science and industry. It is so common to obtain vast of image, audio and video files or any type of measurements for the surveillance systems, medical applications and military target recognition and so on. Internet is also another source of data for many applications such as social network analysis. This phenomenon brings unlimited pattern recognition problems from many domains, with huge amount of data and features. Generally one can either train a single classifier with/without feature selection/extraction. However it is still time, money and effort consuming to label these datasets. Therefore training one classifier alone may be useless due to small amount of instances compared to the number of features (curse of dimensionality) [3]. On the other hand feature selection/extraction may not always improve the classification accuracy. Additionally, in some applications different types of sensors or measurement methods can be used to acquire the data samples. Thus features can be represented in multiple views and concatenation to form the whole feature space may sometimes be problematic. Therefore, instead of training one classifier with/without selection/extraction, alternative methods such as; ensemble of classifiers could be used.

In this thesis we focused on feature subspace selection methods for classifier ensembles and proposed two novel feature subspace selection methods. The proposed methods are evaluated under both supervised and semi-supervised learning scenarios. In supervised learning the proposed algorithms are compared with Random Subspaces (RAS) algorithm that randomly selects the feature subspaces used in the ensembles. In semi-supervised learning the algorithms are compared with RASCO (Random Subspace Method for Co-training) algorithms. In high dimensional feature spaces if there are many irrelevant features and redundancy, it is possible to obtain diverse but inaccurate classifiers with the RAS and RASCO algorithms. The subspace selection methods proposed in this thesis are also aimed to remedy these problems. The first method is used in Rel-RAS and Rel-RASCO algorithms where Rel-RAS is the

relevant random subspace method for supervised learning and Rel-RASCO is the relevant random subspace method for Co-training. The second method modifies the mRMR (minimum Redundancy Maximum Relevance) feature selection algorithm and is used in the mRMR-RAS and mRMR-RASCO algorithms where mRMR-RAS is the minimum redundancy maximum relevance random subspace method for supervised learning and mRMR-RASCO is the minimum redundancy maximum relevance random subspace method for Co-training.

The superiority of the proposed methods are given with the experiments on five real and synthetic datasets with KNN, LDC, decision tree and SVM classifiers based on the accuracy achieved. We found out that in supervised learning Rel-RAS and mRMR-RAS algorithms outperform the RAS algorithm and single classifiers when KNN, LDC and decision tree are used. On the other hand single SVM also performs as good as the ensemble methods. However, when the dataset has redundant features, the proposed algorithms outperform both RAS and single SVM classifier. Besides in semi-supervised learning Rel-RASCO and mRMR-RASCO algorithms generally outperform the RASCO algorithm and single classifier at the beginning and at the end of the Co-training. These results are explained with the RM-characteristics of feature subspaces in terms of mean accuracies of the individual classifiers. The proposed algorithms provide feature subsets agree on the class labels more than RAS and RASCO. This also tends the classifiers to be less diverse. Diversity analysis of the classifiers is obtained using, non-pairwise diversity measure, Kohavi Wolpert (KW) Variance. Besides information theoretic based low order diversity (LOD) and information theoretic scores (ITS) of the classifier diversities are evaluated. KW-variance and LOD results show that the proposed algorithms produce less diverse classifier ensembles than the ensembles generated with RAS and RASCO. On the other hand the superiority of an ensemble algorithm can be explained with the information theoretic score (ITS) [26] and it is shown that there is a relationship between ITS and ensemble classifier accuracy. Unlike the RAS and RASCO, the proposed algorithms have high ITS on both supervised and semi-supervised learning scenarios.

This work can be extended in different steps. Analysis of the algorithms are obtained with RM-characteristics of feature subspaces, KW-Variance diversity measure, information theoretic based low order diversity and information theoretic scores.

Previously there have been some attempts [100] to understand the behavior of Co-training in terms of PAC analysis (Probably Approximately Correct). PAC analysis of the classifier ensembles can be a way to extend this work. Bias-Variance decomposition [101] is also another analysis that can be applied to classifier ensembles. The proposed algorithms only select features based on a probability distribution. However the algorithms do not consider redundancy between the feature subspaces. It seems to be an open issue and a new subspace selection method that considers both relevance and redundancy between feature subsets may produce good classifier ensembles and may also increase the classification accuracy. On the other hand in this thesis ITS is only used to explain the superiority of the classifier ensembles. It can also be used as a classifier selection criteria in ensembles as shown in [26].

REFERENCES

[1] **Krishnapuram, B.**, **Williams, D.**, **Xue, Y.**, **Carin, L.**, **Figueiredo, M.A.T. and Hartemink, A.J.**, 2005. Active Learning of Features and Labels, *Proc.of the Workshop on Learning With Multiple Views at the 22nd International Conference on Machine Learning*, pp. 43–50.

[2] **Kuncheva, L.I.,** 2004. *Combining Pattern Classifiers: Methods and Algorithms*, Wiley-Interscience.

[3] **Skurichina, M. and Duin, R.P.W.,** 2005. Combining Feature Subsets in Feature Selection, *6th International Workshop on Multiple Classifier Systems*, Seaside, CA, USA, June 13-15.

[4] **Guyon, I. and Elisseeff, A.,** 2003. An Introduction to Variable and Feature Selection, *Journal of Machine Learning Research*, **3**, pp. 1157–1182.

[5] **Zongker, D. and Jain, A.K.,** 1996. Algorithms for Feature Selection: An Evaluation, *Proc. of the International Conference on Pattern Recognition*, Vienna, Austria, August 25-29.

[6] **Wu, S. and Flach, P.A.,** 2002. Feature Selection with Labelled and Unlabelled data, *ECML/PKDD'02 workshop on Integration and Collaboration Aspects of Data Mining, Decision Support and Meta-Learning*, University of Helsinki, August 19-23, pp. 156–167.

[7] **Peng, H.**, **Long, F. and Ding, C.**, 2005. Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**, pp. 1226 – 1238.

[8] **Theodoridis, S. and Koutroumbas, K.**, 2003. *Pattern Recognition*, Academic Press.

[9] **Duda, R.**, **Hart, P. and Stork, D.**, 2001. *Pattern Classification*,Wiley Interscience.

[10] **Seeger, M.**, 2002. Learning with Labeled And Unlabeled Data, *Technical Report*, University of Edinburgh, Edinburgh, UK.

[11] **Zhou, Z.H. and Li, M.**, 2005. Semi-Supervised Regression with Co-Training, *Proc. of the International Joint Conference on Artificial Intelligence*, Edinburgh, UK, pp. 908–916.

[12] **Chapelle, O.**, **Scholkopf, B. and Zien, A.**, 2006. *Semi-Supervised Learning*, MIT Press.

[13] **Roli, F.**, 2005. Semi-supervised Multiple Classifier Systems: Background and Research Directions, *Proc. of the 6th International Workshop on Multiple Classifier Systems*, Seaside, CA, USA, pp. 1–11.

[14] **Novak, B.**, 2004. Use of Unlabeled Data in Supervised Machine Learning, *Proc. of the SIKDD at Multiconference IS*, Ljubljana, Slovenia, October 12-15.

[15] **Blum, A. and Mitchell, T.**, 1998. Combining Labeled and Unlabeled Data with Co-training, *Proc. of the 11th Annual Conference on Computational Learning Theory (COLT '98),* Madison, Wisconsin, USA, pp. 92–100.

[16] **Xu, Y.**, **Zhang, C. and Yang, J.**, 2005. Semi-Supervised Classification of Musical Genre Using Multi-View Features, *Proc. of the International Computer Music Conference (ICMC 2005)*, Barcelona, Spain.

[17] **Ho, T.K.**, 1998. The Random Subspace Method for Constructing Decision Forests, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**(8), pp. 832–844.

[18] **Ho, T.K.**, 1998. Nearest Neighbors in Random Subspaces, vol. **1451**, *Lecture Notes in Computer Science*, pp. 640–648.

[19] **Skurichina, M. and Duin, R.P.W.**, 2002. Bagging, Boosting and the Random Subspace Method for Linear Classifiers, *Pattern Analysis and Applications*, **5**(2), pp. 121–135.

[20] **Blum, A.L. and Langley, P.**, 1997. Selection of Relevant Features and Examples in Machine Learning, *Artificial Intelligence*, **97**(1-2), pp. 245–271.

[21] **Gulgezen, G.**, **Cataltepe, Z. and Yu, L.**, 2009. Stable and Accurate Feature Selection, *ECML PKDD European Conference on Machine Learning and Knowledge Discovery in Databases*, Bled, Slovenia, September 7-11, 2009, pp. 455–468.

[22] **Zhou, Z.H. and Li, M.**, 2005. Tri-Training: Exploiting Unlabeled Data Using Three Classifiers, *IEEE Transactions on Knowledge and Data Engineering*, **17**(11), pp. 1529–1541.

[23] **Wang, J.**, **Luo, S.W. and Zeng, X.H.**, 2008. A Random Subspace Method for Co-training, *Proc. of the International Joint Conference on Neural Networks(IJCNN 2008)*, Hong Kong, China, pp. 195–200.

[24] **Yaslan, Y. and Cataltepe, Z.**, 2008. Co-Training with Adaptive Bayesian Classifier Combination, *Proc. of the International Symposium on Computer and Information Sciences (ISCIS2008)*, Istanbul, Turkey, October 27-29, pp. 1–4.

[25] **Brown, G.**, 2009. An Information Theoretic Perspective on Multiple Classifier Systems, *8th International Workshop on Multiple Classifier Systems*, Reykjavik, Iceland, June 10-12, pp. 344–353.

[26] **Meynet, J. and Thiran, J.P.**, 2010. Information Theoretic Combination of Pattern Classifiers, *Pattern Recognition*, **43**, pp. 3412–3421.

[27] **Polikar, R.**, 2009. Ensemble Learning, *Scholarpedia*, **4**(1), pp. 2776.

[28] **Polikar, R.**, 2006. Ensemble Based Systems in Decision Making, *IEEE Circuits and Systems Magazine*, **3**, pp. 21–45.

[29] **Kittler, J.**, 2000. A Framework for Classifier Fusion: Is It Still Needed?, *Proc. of the SSPR: SSPR, International Workshop on Advances in Structural and Syntactical Pattern Recognition,* Alicante, Spain, August 30-September 1, pp. 45–56.

[30] **Brown, G.**, 2010. Ensemble Learning, *Encyclopedia of Machine Learning*, Springer Press.

[31] **Suen, C.Y. and Lam, L.**, 2000. Multiple Classifier Combination Methodologies for Different Output Levels, *Proc. of the 1st International Workshop on Multiple Classifier Systems*, Cagliari, Italy, June 21-23, pp. 52–66.

[32] **Dietterich, T.G.**, 2000. Ensemble Methods in Machine Learning, *Proc. of the 1st International Workshop on Multiple Classifier Systems*, Cagliari, Italy, June 21-23, pp. 1–15.

[33] **Lam, L.**, 2000. Classifier Combinations: Implementations and Theoretical Issues, *Proc. of the 1st International Workshop on Multiple Classifier Systems*, Cagliari, Italy, June 21-23, pp. 77–86.

[34] **Rokach, L.**, 2009. Taxonomy for Characterizing Ensemble Methods in Classification Tasks: A Review and Annotated Bibliography, *Computational Statistics & Data Analysis*, **53**(12), pp. 4046–4072.

[35] **Valentini, G. and Masulli, F.**, 2002. Ensembles of Learning Machines, *Lecture Notes in Computer Science*, **2486**, pp. 3–20.

[36] **Breiman, L.**, 1996. Bagging Predictors, *Machine Learning*, **24**, pp. 123–140.

[37] **Oza, N.C. and Tumer, K.**, 2001. Input Decimation Ensembles: Decorrelation through Dimensionality Reduction, *Proc. of the International Workshop on Multiple Classifier Systems*, Cambridge, UK, July 2-4, pp. 238–247.

[38] **Li, S.Z. and Jain, A.K.**, 2009. *Encyclopedia of Biometrics*, Springer US.

[39] **Alpaydin, E.**, 2004. *Introduction to Machine Learning*, The MIT Press.

[40] **Freund, Y. and Schapire, R.E.**, 1997. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting, *JCSS: Journal of Computer and System Sciences*, **55**, pp. 119–139.

[41] **Raetsch, G.**, **Onoda, T. and Mueller, K.R.**, 2001. Soft Margins for AdaBoost, *Machine Learning*, **42**(3), pp. 287-320.

[42] **Wolpert, D.H.**, 1992. Stacked Generalization, *Neural Networks*, **5**(2), pp. 241–260.

[43] **Tumer, K. and Oza, N.C.**, 2003. Input Decimation Ensembles, *Pattern Analysis and Applications*, **6**, pp. 65–77.

[44] **Xu, L.**, **Krzyzak, A. and Suen, C.Y.**, 1992. Methods of Combining Multiple Classifiers and Their Applications to Handwriting Recognition, *IEEE Transactions on Systems,Man and Cybernetics*, **22**(3), pp. 418–435.

[45] **Shipp, C.A. and Kuncheva, L.I.**, 2002. Relationships Between Combination Methods and Measures of Diversity in Combining Classifiers, *Information Fusion*, **(3)**, pp. 135–148.

[46] **Fumera, G. and Roli, F.**, 2005. A Theoretical and Experimental Analysis of Linear Combiners for Multiple Classifier Systems, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, June, **27**(6), pp. 942–956.

[47] **Huang, Y.S. and Suen, C.Y.**, 1995. A Method of Combining Multiple Experts for the Recognition of Unconstrained Handwritten Numerals, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **17**(1), pp. s90–94.

[48] **Ho, T.K.**, **Hull, J.J. and Srihari, S.N.**, 1994. Decision Combination in Multiple Classifier Systems, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **16**(1), pp. 66–75.

[49] **Kittler, J.**, **Hatef, M.**, **Duin, R.P.W. and Matas, J.**, 1998. On Combining Classifiers, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**(3), pp. 226–239.

[50] **Kittler, J.**, 1998. Combining Classifiers: A Theoretical Framework, *Pattern Analysis and Applications*, **1**(1), pp. 18–27.

[51] **Brown, G.**, **Wyatt, J.L.**, **Harris, R. and Yao, X.**, 2005. Diversity Creation Methods: A Survey and Categorisation, *Information Fusion*, **6**, pp. 5–20.

[52] **Kuncheva, L.**, **Skurichina, M. and Duin, R.P.W.**, 2002. An Experimental Study on Diversity for Bagging and Boosting with Linear Classifiers, *Information Fusion*, **3**(4), pp. 245–258.

[53] **Windeatt, T.**, 2005. Diversity Measures for Multiple Classifier System Analysis and Design, *Information Fusion*, **6**(1), pp. 21–36.

[54] **Kuncheva, L.I. and Whitaker, C.J.**, 2003. Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy, *Machine Learning*, **51**(2), pp. 181–207.

[55] **Meynet, J. and Thiran, J.P.**, 2007. Information Theoretic Combination of Classifiers with Application to AdaBoost, *7th International Workshop on Multiple Classifier Systems*, Prague, Czech Republic, May 23-25, pp. 171–179.

[56] **Breiman, L.**, 2001. Random Forests, *Machine Learning*, **45**(1), pp. 5-32.

[57] **Rodriguez, J.J.**, **Kuncheva, L.I. and Alonso, C.J.**, 2006. Rotation Forest: A New Classifier Ensemble Method, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **28**(10), pp. 1619–1630.

[58] **Guerra-Salcedo, C. and Whitley, D.**, 1999. Feature selection mechanisms for ensemble creation: a genetic search perspective, *Proc. of Data Mining with Evolutionary Algorithms: Research Directions*, Orlando, Florida, July 18, pp. 13–17.

[59] **Opitz, D.W.,** 1999. Feature Selection for Ensembles, *Proc. of the 16th National Conference on Artificial Intelligence and 11th Conference on Innovative Applications of Artificial Intelligence AAAI/IAAI,* Orlando, Florida, USA, pp. 379–384.

[60] **Oliveira, L.S.**, **Sabourin, R.**, **Bortolozzi, F. and Suen, C.Y.**, 2003. Feature Selection for Ensembles: A Hierarchical Multi-Objective Genetic Algorithm Approach, *Proc. of the 7th International Conference on Document Analysis and Recognition*, Edinburgh, UK, August 3-6, pp. 676–680.

[61] **Vale, K.M.O.**, **Dias, F.G.**, **Canuto, A.M.P. and Souto, M.C.P.**, 2008. A Class-Based Feature Selection Method for Ensemble Systems, *8th International Conference on Hybrid Intelligent Systems*, Barcelona, Spain, September 10-12, pp. 596–601.

[62] **Tsymbal, A.**, **Pechenizkiy, M. and Cunningham, P.**, 2005. Diversity in Search Strategies for Ensemble Feature Selection, *Information Fusion*, **6**(1), pp. 83–98.

[63] **Tao, D. and Tang, X.**, 2004. Random Sampling Based SVM for Relevance Feedback Image Retrieval, *Proc. of the Computer Vision and Pattern Recognition Conference*, Washington, USA, pp. 647–652.

[64] **Bertoni, A.**, **Folgieri, R. and Valentini, G.**, 2005. Bio-Molecular Cancer Prediction with Random Subspace Ensembles of Support Vector Machines, *Neurocomputing*, **63**, pp. 535–539.

[65] **Goldberg, D.E. and Deb, K.**, 1991. A Comparative Analysis of Selection Schemes Used in Genetic Algorithms, *Proc. of the Foundations of Genetic Algorithms*, pp. 69–93.

[66] **Brown, G.**, 2009. A New Perspective for Information Theoretic Feature Selection, *Proc. of the 12th International Conference on Artificial Intelligence and Statistics*, Florida, USA, April 16-18.

[67] **Duin, R.P.W.**, 2000. PRTOOLS – Version 3.0 A Matlab Toolbox for Pattern Recognition, *Proc. of SPIE*.

[68] **Hall, M.**, **Frank, E.**, **Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I.H.**, 2009: The Weka Data Mining Software: An Update, *SIGKDD Explorations*, Vol. **11**(1) pp. 10-18.

[69] **Chang, C.C. and Lin, C.J.**, 2001. LIBSVM. a library for support vector machines, Retrieved June 10, 2010, from http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[70] **Sun, S. and Zhang, C.**, 2007. Subspace Ensembles for Classification, *Physica A*, **385**, pp. 199–207.

[71] **Sun, S.**, **Zhang, C. and Zhang, D.**, 2007. An Experimental Evaluation of Ensemble Methods for EEG Signal Classification, *Pattern Recognition Letters*, **28**(15), pp. 2157–2163.

[72] **Zhou, Y. and Goldman, S.**, 2004. Democratic Co-Learning, *Proc. of the 16th IEEE International Conference on Tools with Artificial Intelligence,* Boca Raton, Florida, USA, November 15-17, pp. 594–602.

[73] **Yaslan, Y. and Cataltepe, Z.**, 2009. Random Relevant and Non-Redundant Feature Subspaces for Co-training, *Proc. of the 10th International Conference on Intelligent Data Engineering and Automated Learning*, Burgos, Spain, September 23-26, pp. 679–686.

[74] **Yaslan, Y. and Cataltepe, Z.**, 2010. Co-training with Relevant Random Subspaces, *Neurocomputing*, **73**, pp. 1652–1661.

[75] **Cunningham, P. and Carney, J.**, 2000. Diversity versus Quality in Classification Ensembles Based on Feature Selection, *Proc. of the 11th European Conference on Machine Learning*, Barcelona, Spain, May 31-June 2, pp. 109–116.

[76] **Dempster, A.P.**, **Laird, N.M. and Rubin, D.B.**, 1977. Maximum Likelihood from Incomplete Data via the EM Algorithm, *Journal of the Royal Statistical Society. Series B(Methodological)*, **39**, pp. 1–38.

[77] **Nigam, K. and Ghani, R.**, 2000. Analyzing the Effectiveness and Applicability of Co-training, *Proc. of the 9th International Conference on Information and Knowledge Management*, McLean, USA, November 6-11, pp. 86–93.

[78] **Goldman, S. and Zhou, Y.**, 2000. Enhancing Supervised Learning with Unlabeled Data, *Proc. of the 17th IEEE International Conference on Machine Learning*, Stanford, CA, USA, June 29-July 2, pp. 327–334.

[79] **Hady, M.F.A. and Schwenker, F.**, 2008. Co-Training by Committee: A New Semi-Supervised Learning Framework, *Proc. of the ICDM Workshop on Foundations of Data Mining,* Pisa, Italy, December 15-19, pp. 563–572.

[80] **Yan, R. and Naphade, M.R.**, 2005. Semi-Supervised Cross Feature Learning for Semantic Concept Detection in Videos, *Proc. of the Computer Vision and Pattern Recognition Conference*, San Diego, CA, USA, June 20-26, pp. 657–663.

[81] **Li, M. and Zhou, Z.H.**, 2007. Improve Computer-Aided DiagnosisWith Machine Learning Techniques Using Undiagnosed Samples, *IEEE Transactions on Systems, Man and Cybernetics, Part A*, **37**(6), pp. 1088–1098.

[82] **Didaci, L. and Roli, F.**, 2006. Using Co-training and Self-training in Semi-supervised Multiple Classifier Systems, *Lecture Notes in Computer Science*, Vol. **4109**, pp. 522–530.

[83] **Asuncion, A. and Newman, D.J.**, 2007. UCI Machine Learning Repository, Retrieved February 13, 2010, from http://www.ics.uci.edu/ ~mlearn/MLRepository.html.

[84] **West, M.**, 2003. Bayesian Factor Regression Models in the "Large p, Small n" Paradigm, *Bayesian Statistics*, pp. 723–732.

[85] **Li, J.**, **Allinson, N.**, **Tao, D. and Li, X.**, 2006. Multitraining Support Vector Machine for Image Retrieval, *IEEE Transactions on Image Processing*, **15**, pp. 3597–3601.

[86] **Tao, D.**, **Tang, X.**, **Li, X. and Wu, X.**, 2006. Asymmetric Bagging and Random Subspace for Support Vector Machines-Based Relevance Feedback in Image Retrieval, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **28**, pp. 1088–1099.

[87] **Feger, F. and Koprinska, I.**, 2006. Co-training Using RBF Nets and Different Feature Splits, *Proc. of the International Joint Conference on Neural Networks*, Vancouver, BC, Canada, July 16-21, pp. 1878–1885.

[88] **Kiritchenko, S. and Matwin, S.**, 2001. Email Classification with Co-Training, *Proc. of the Conference of the Centre for Advanced Studies on Collaborative Research*, Toronto, Ontario, Canada, November 5-7.

[89] **Maeireizo, B.**, **Litman, D. and Hwa, R.**, 2004. Co-Training For Predicting Emotions With Spoken Dialogue Data, *Proc. of the 42nd Annual Meeting of the Association for Computational Linguistics,* Barcelona, Spain, July 21-26 pp. 203-206.

[90] **Boley, D.**, **Gini, M.**, **Gross, R.**, **Han, E.**, **Hastings, K.**, **Karypis, G.**, **Kumar, V.**, **Mobasher, B. and Moore., J.**, 1999. Partitioning-Based Clustering for Web Document Categorization, *Decision Support Systems*, **27**, pp. 329–341.

[91] **Tzanetakis, G. and Cook, P.**, 2002. Musical Genre Classification of Audio Signals, *IEEE Transactions on Speech and Audio Processing*, **10**(5), pp. 293–302.

[92] **Moerchen, F.**, **Ultsch, A.**, **Thies, M. and Loehken, I.**, 2006. Modelling Timbre Distance with Temporal Statistics from Polyphonic Music, *IEEE Transactions on Audio, Speech and Language Processing*, **14**, pp. 81–90.

[93] **Zeimpekis, D. and Gallopoulos, E.**, 2005. TMG: A MATLAB toolbox for generating term-document matrices from text collections, *Proc. of the Grouping Multidimensional Data: Recent Advances in Clustering*, Springer-Verlag, Heidelberg, pp. 187–210.

[94] **Ding, C. and Peng, H.**, 2003. Minimum Redundancy Feature Selection from Microarray Gene Expression Data, *Proc. of the Computational Systems Bioinformatics*, Stanford, CA, USA, August 11-14, pp. 523–528.

[95] **Cozman, F.G. and Cohen, I.**, 2002. Unlabeled Data Can Degrade Classification Performance of Generative Classifiers, *Proc. of the 15th International Florida Artificial Intelligence Research Society Conference*, Florida, USA, May 14-16, pp. 327–331.

[96] **Li, T. and Ogihara, M.**, 2005. Semisupervised Learning from Different Information Sources, *Knowledge and Information Systems*, **7**, pp. 289–309.

[97] **Tian, Q.**, **Yu, J.**, **Xue, Q. and Sebe, N.**, 2004. A new analysis of the value of unlabeled data in semi-supervised learning for image retrieval, *Proc. of the IEEE International Conference on Multimedia and Expo*, Taipei, Taiwan, June 27-30, pp. 1019–1022.

[98] **Ghani, R.**, 2002. Combining Labeled and Unlabeled Data for MultiClass Text Categorization, *Proc. of the International Conference on Machine Learning* , Sydney, Australia, July 8-12, pp. 187-194.

[99] **Catal, C. and Diri, B.**, 2009. Unlabelled Extra Data Do Not Always Mean Extra Performance for Semi-supervised Fault Prediction, *Expert Systems*, **26**, pp. 458–471.

[100] **Wang, W. and Zhou, Z.H.**, 2007. Analyzing Co-training Style Algorithms, ECML, Springer, *Lecture Notes in Computer Science*, Vol. **4701,** pp. 454–465.

[101] **Domingos, P.**, 2000. A Unified Bias-Variance Decomposition for Zero-One and Squared Loss, *Proc. of the 7th Conference on Artificial Intelligence and 12th Conference on Innovative Applications of Artificial Intelligence*,Austin, Texas, USA, July 30-August 3, pp. 564–569.

[102] **Yaslan, Y. and Cataltepe, Z.**, 2009. Audio Genre Classification with Semi Supervised Feature Ensemble Learning, *Proc. of the MML Workshop at ECML/PKDD*, Bled Slovenia, September 7-11.

[103] **Liu, C. and Wechsler, H.**, 2000. Robust Coding Schemes for Indexing and Retrieval from Large Face Databases, *IEEE Transactions on Image Processing*, **9**(1), pp. 132–137.

[104] **Webb, A.**, 2002. *Statistical Pattern Recognition*, John Wiley & Sons, New York.

[105] **Quinlan, J.R.**, 1996. Improved use of continuous attributes in C4.5, *Journal of Artificial Intelligence Research*, **4**, pp. 77–90.

[106] **Vapnik, N.V.**, 2000. *The Nature of Statistical Learning Theory*, Springer-Verlag, New York.

[107] **Joachims, T.**, 2002. *Learning to Classify Text using Support Vector Machines*, Kluwer Academic Publisher.

[108] **Burges, C.J.C.**, 1998. A Tutorial on Support Vector Machines for Pattern Recognition, *Data Mining and Knowledge Discovery*, **2**(4), pp. 121–167.

APPENDICES

APPENDIX A. Feature Discretization

Feature discretization is used when the features are continuous in the Rel-RAS, RelNR-RAS, Rel-RASCO and RelNR-RASCO algorithms. In order to compute the mutual information in these algorithms we first discretize the features into 10 bins.

Let F_k, $k = \{1, 2, ..., d\}$ denote the n dimensional feature vector for the kth feature in the dataset and $F_k = \{x_{1k}, x_{2k}, ..., x_{nk}\}$. The feature discretization algorithm is given below.

Algorithm 10 Feature Discretization

```
// b: Number of bins
// F_k: feature vector to be discretized
// DF: Discretized feature vector
// n: Number of features
disc = [(-floor(b/2)):(floor(b/2))];
mn = min(F_k), mx = max(F_k)
binwidth = (mn - mx)/b
E = mn + binwidth * (0:b);
E(1) = -inf, E(end) = inf;
for i = 1 to n do
   for j = 1 to b do
      if F_k(i) >= E(j) AND F_k(i) < E(j+1) then
         DF(i) = disc(j)
      end if
   end for
end for
```

APPENDIX B. Basics of Information Theory

The uncertainty present in a distribution of a random variable X, can be measured by entropy, $H(X)$, and is denoted as follows [25]:

$$H(X) = -\sum_{i=1}^{|X|} p(x_i)\log(p(x_i)) \tag{B.1}$$

An estimate of the probability distribution is obtained using frequency counts. Therefore $p(x_i) = \frac{\#x_i}{N}$, where $\#x_i$ is the number of observations on x_i and N is the number of total observations. The entropy is maximal if all events are equally likely. Using the rules of probability theory, the conditional entropy of X given Y can be written as follows:

$$H(X|Y) = \sum_{j=1}^{|Y|}\sum_{i=1}^{|X|} p(x_i|y_j)\log(p(x_i|y_j)) \tag{B.2}$$

The mutual information between X and Y, $I(X;Y)$, is the difference between the uncertainty present in the distribution of X and uncertainty remained in X after Y occured:

$$I(X;Y) = H(X) - H(X|Y) \tag{B.3}$$

$I(X;Y)$ can be expanded as follows:

$$I(X;Y) = \sum_{j=1}^{|Y|}\sum_{i=1}^{|X|} p(x_i,y_j)\log(\frac{p(x_i,y_j)}{p(x_i)p(y_j)}) \tag{B.4}$$

The information shared between X_1 and X_2 after Y occured is the conditional mutual information, $I(X_1;X_2|Y)$ and can be written as follows:

$$I(X_1;X_2|Y) = H(X_1|Y) - H(X_1|X_2,Y) \tag{B.5}$$

$$I(X_1;X_2|Y) = \sum_{k=1}^{|Y|} p(y_k)\sum_{j=1}^{|X_2|}\sum_{i=1}^{|X_1|} p(x_i,x_j|y_k)\log(\frac{p(x_i,x_j|y_k)}{p(x_i|y_k)p(y_j|y_k)}) \tag{B.6}$$

APPENDIX C. Datasets

In this thesis, experimental results are obtained on 5 different real datasets from different application areas: 'OptDigits' (Optical Recognition of Handwritten Digits), 'MFeat' (Multiple Features) and 'Isolet' (Isolated Letter Speech) datasets from the UCI machine learning repository [83], 'Classic-3' text dataset from [90] and the 'Audio Genre' dataset of [91]. Table C.1 shows the number of features, instances and classes of the features for all 5 datasets.

Table C.1: Real Datasets

Dataset	# features	# instances	# classes
Audio Genre	50	500	5
OptDigits	64	5620	10
Classic-3	273	3000	3
Isolet	617	480	2
MFeat	649	2000	10

Audio Genre Dataset: The 5 least confused genres of Tzanetakis dataset [91], Classical, Hiphop, Jazz, Pop and Reggae, each with 100 samples, are used [102]. Two different sets of audio features are computed. First, timbral, rhytmic content and pitch content features yielding 30 features are extracted using the Marsyas Toolbox [91]. Timbral features are generally used for music-speech discrimination and speech recognition. They differentiate mixture of sounds with the same or similar rhythmic content. Rhythmic content features characterize the movement of music signals over time and contain such information as the regularity of the rhythm, the beat, the tempo, and the time signature. The melody and harmony information about the music signal is obtained by pitch detection techniques. Next, 20 features covering temporal and spectral properties are extracted using the Databionic Music Miner framework [92].

UCI Optdigits Dataset: Optical Recognition of Handwritten Digits Dataset (optdigits) contains 64 features with 10 classes. Features are extracted from normalized bitmaps of handwritten digits from a preprinted form. Images are 32×32 bitmaps and they are divided into nonoverlapping blocks of 4×4. In each subblock the number of pixels are counted to generate an input matrix of 8×8 where each element is an integer in the range $0\ldots16$ [83].

Classic-3 Dataset: Classic-3 data corpus contains the paper abstracts of 3 different types of journals. They are namely MEDLINE, CISI and CRAN. MEDLINE contains the abstracts from medical journals, CISI contains the abstracts from information retrieval field and CRAN contains the abstracts from aeronautical systems area. In the experiments Term Frequencies (TF) of words are used as features and they are

obtained using Term-to-Matrix generator (TMG) Matlab Toolbox [93]. For each class equal number of instances are selected in order to balance the dataset.

UCI Isolated Letter Speech Recognition Dataset: This dataset contains the 617 speech features (contour, sonorant, pre-sonorant and post-sonorant features) with 480 instances from B and C letters [83].

Multiple Features (Mfeat) dataset: Multiple Features (Mfeat) dataset consist of 2000 instances of handwritten digits with 10 classes. There are 649 features: 76 Fourier coefficients of the character shapes, 216 profile correlations, 64 Karhunen-Love coefficients, 240 pixel averages in 2×3 windows, 47 Zernike moments and 6 morphological features [83].

APPENDIX D. Linear Discriminant Classifier

When the underlying probability density functions (pdf's) are known, Bayes classifier gives the minimum error [2] [103]. A posteriori probability function of class c_i given x is:

$$p(c_i|x) = \frac{p(x|c_i)p(c_i)}{p(x)} \tag{D.1}$$

where $p(x|c_i)$ is the class conditional pdf of c_i and $p(x)$ is the mixture density. The class with the highest posterior probability will be the choice for a given x. The posterior probabilities can be written with discriminant functions, g_i, as follows:

$$g_i(x) = p(c_i|x), \qquad i = 1,...,c \tag{D.2}$$

The decision for x , $D(x)$, is:

$$D(x) = \max_{1,..,c}\{p(c_i|x)\} = \max_{1,..,c}\{g_i(x)\} \tag{D.3}$$

The $p(x)$ for all classes are same then $g_i(x)$ can be written as:

$$g_i(x) = \log[p(c_i)p(x|c_i)], \qquad i = 1,...,c \tag{D.4}$$

Let all classes are normally distributed, $p(x|c_i) \sim N(\mu_i, \Sigma_i)$, with μ_i means and Σ_i covariance matrices and $i = 1,...,c$. Then $g_i(x)$ can be obtained as:

$$\begin{aligned} g_i(x) &= \log[p(c_i)] + \log\left\{ \frac{1}{(2\pi)^{n/2}\sqrt{|\Sigma_i|}} \exp\left[\frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i)\right]\right\} \\ &= \log[p(c_i)] - \frac{n}{2}\log(2\pi) - \frac{1}{2}\log(|\Sigma_i|) - \frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i) \end{aligned} \tag{D.5}$$

where $i = 1,...,c$ In our computations all covariance matrices are assumed to be same and $p(x|c_i) \sim N(\mu_i, \Sigma)$, and if we eliminate all the terms that are constant for all c_i then the discriminant functions can be written as follows:

$$\log[p(c_i)] - \frac{1}{2}(\mu_i)^T\Sigma^{-1}(\mu_i) + (\mu_i)^T\Sigma^{-1}(x) = w_{i0} + w_i^T x \tag{D.6}$$

where w_{i0} and w_i are the coefficients of the linear discriminant function [2]. Mean values and covariance matrix are estimated from training data [2].

APPENDIX E. K-Nearest Neighbour Classifier

K-Nearest Neighbour (KNN) method can be used to estimate density. In K-Nearest Neighbour density estimation, the aim is to find the volume, V, while fixing the probability of k/n. However the density estimation does not work very well. On the other hand KNN method can be used for non-parametric classification [39].

Let k_i be the samples belonging to class c_i in k samples and n_i be the total number of examples in class c_i. Then the estimate of the class conditional density can be written as follows:

$$\hat{p}(x|c_i) = \frac{k_i}{n_i V} \tag{E.1}$$

The estimate of the prior probability is:

$$\hat{p}(c_i) = \frac{n_i}{n} \tag{E.2}$$

Using the Bayes' theorem, the estimate of the posterior probability is:

$$\hat{p}(c_i|x) = \frac{\hat{p}(x|c_i)\hat{p}(c_i)}{\hat{p}(x)} = \frac{\frac{k_i}{n_i V}\frac{n_i}{n}}{\frac{k}{nV}} = \frac{k_i}{k} \tag{E.3}$$

The algorithm works as follows: For each test instance the k nearest examples are identified using Euclidean distance. The number of sampels, k_i, that belong to class c_i is obtained out of these k samples. Then the test instance is assigned to the class c_i with the maximum number of k_i [39] [104].

APPENDIX F. Decision Tree Classifier

Classification of patterns through questions, where the next question depends on the answer of the current one, is an intiutive way [9]. The sequence of the questions is described as decision tree where the first question constitutes the root node and the others constitute the branches. Classification of a data sample starts from the root node and based on the value of the sample the the subsequent or descending nodes are evaluated. Therefore the feature that best divides the classes should be selected as the root node. Different algorithms have been proposed to find the best feature that splits the data such as; information gain, gain ratio and gini index [9] [105]. In our experiments we used weka implemantation of decision tree J48 with default parameters [68]. J48 implements the C4.5 Quinlan's algorithm [105]. The algorithm works by evaluating the cases in the training set.

Let S be the set of cases and c be the number of classes. Then entropy of S can be obtained as follows:

$$Entropy(S) = \sum_{j=1}^{c} p_j \log_2 p_j \qquad \textbf{(F.1)}$$

where p_j is the probability of the cases belong to class j in S. The information gain for a feature F is $Gain(S,F)$:

$$Gain(S,F) = Entropy(S) - \sum_{v \in values(F)} \frac{|S_v|}{|S|} Entropy(S_v) \qquad \textbf{(F.2)}$$

where $values(F)$ represents the values that F may have and $|S_v|$ represents the number of samples in each subset. The algorithm selects the feature which increases the information gain as a node. The algorithm is applied recursively to obtain other nodes in the tree [105].

APPENDIX G. Support Vector Machines

Support vector machines were developed by Vapnik et al. [106]. Let X_i $(i = 1,2,...,n)$ be the d dimensional ith training sample, $X_i = (x_{i1}, x_{i2}, ..., x_{id})$, in the training dataset X $=(X_1, X_2, ..., X_n)$ with n samples. We consider two class case where l represents the labels ($l \in \{-1, 1\}$) and our aim is to learn a function $g(X_i) = l$. Each example X_i is assumed to be generated from an unkown but fixed probability distribution $P(X, l)$ [107]. The learning problem can be expressed as an optimization problem which aims to minimize the misclassification of the new instances drawn from the same pdf. Goodness of the classifier g can be measured using expected risk, $R(g)$:

$$R(g) = \int \ell(g(X), l) dP(X, l) \qquad \textbf{(G.1)}$$

Where ℓ is the loss function that penalizes the difference between predicted and true labels. Since the underlying distribution isn't known the risk, $R(g)$, can not be minimized directly. Instead the risk over the training set, emprical risk, is minimized:

$$R_{emp}(g) = \frac{1}{n} \sum \ell(g(X), l) \qquad \textbf{(G.2)}$$

With a probability of $1 - \mu$, the expeceted risk has the following boundary [107]:

$$R(g) \leq R_{emp}(g) + \sqrt{\frac{h(\ln(2n/h) + 1) - \ln(\mu/4)}{n}} \qquad \textbf{(G.3)}$$

Where h is the Vapnik-Chervonenkis (VC) dimension of g, and n is the number of training instances, $n > h$. VC dimension is the maximum number of data points that can be separated by any $g(X)$. A simple hypothesis space (small VC-dimension) may provide classifiers with high training error. On the other hand a hypothesis with a high VC dimension and small training error may fit the training data and inaccurately classify the new instances which is called "overfitting". Therefore using the hypothesis space with right complexity, optimum VC-dimension, is very important. It was shown that margin, the distance between the hperplane to the closest instance, can be used to upper bound the VC-dimension [107] and it is used for the fundamental derivations of the SVM.

SVM aims to find a separating hyperplane with the largest margin for linearly separable case. Let w be the weight vector and b be the threshold. The hyperplane separates the positive training examples into one side of the hyperplane and negative examples to the other side. This can be formulated for each training data (X_i, l_i) as follows:

$$l_i(w.X_i + b) > 0 \qquad \textbf{(G.4)}$$

There is only one hyperplane with maximum largest margin for separable case and the examples closest to the hyperplane are called support vectors [107]. The margin is $2/\|w\|$ and maximizing the margin is equivalent to the following optimization problem:

$$\min_{w,b} \frac{1}{2}\|w\|^2 \qquad \textbf{(G.5)}$$

subject to:

$$l_i(w.X_i + b) \geq 1, \qquad \forall i \qquad \textbf{(G.6)}$$

This constraint optimization problem is solved by introducing the Lagrangian:

$$L_p(w,b,\alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i \left[l_i(w.X_i + b) - 1\right] \qquad \textbf{(G.7)}$$

This function should be minimized with respect to w, b and miximized with respect to Lagrange multipliers, α. The saddle point is found at: $\partial L/w$ and $\partial L/b$. After differentiating the following dual optimization problem is obtained:

$$L_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j l_i l_j {X_i}^T X_j \qquad \textbf{(G.8)}$$

subject to:

$$\alpha_i \geq 0,\ i = 1,2,...,n \qquad \sum_{i=1}^{n} \alpha_i l_i = 0 \qquad \textbf{(G.9)}$$

The solution of this optimization problem is a linear decision function. The solution up to here only considers the separable case. However for noisy datasets this may not be the optimal choice. An alternative way to find a trade-off between emprical risk and capacity is to introduce slack variables, ξ, in Equation **6**:

$$l_i(w.X_i + b) \geq 1 - \xi_i,\ \xi \geq 0 \qquad i = 1,2,...,n \qquad \textbf{(G.10)}$$

The trade-off between emprical risk and capacity is controled by adding a constant C that penalizes the instances fall into the margin. Then the optimization problem becomes:

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i \qquad \textbf{(G.11)}$$

This can be turned into another dual optimization problem:

$$L_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j l_i l_j {X_i}^T X_j \qquad \textbf{(G.12)}$$

subject to:

$$0 \leq \alpha_i \leq C,\ i = 1,2,...,n \qquad \sum_{i=1}^{n} \alpha_i l_i = 0 \tag{G.13}$$

Solving quadratic optimization problems in order to find the α and support vector values can be cumbersome for large scales. Several algorithms have been proposed to find the support vectors, i.e. Sequential Minimal Optimization [107]. More details on SVM can be found in [107] [108] and [9].

APPENDIX H. T-Test

In the experiments accuracies are also evaluated with the hypothesis testing. A t-test is applied to determine whether the means of the experimental results are different enough from each other. Let x and y be the two vectors with size of n, the t score can be found as follows:

$$t = \frac{\bar{x} - \bar{y}}{\sqrt{\frac{var_x + var_y}{n}}} \tag{H.1}$$

where var_x and var_y are the variance of x and y respectively.

The significance, p, value is found using t-distribution table [8].

CURRICULUM VITAE

Candidate's full name: Yusuf Yaslan

Place and date of birth: Adıyaman, February 13^{th}, 1979

Permanent Address: Istanbul Teknik Üniversitesi, Bilgisayar Mühendisliği Bölümü, 34469, Maslak / İstanbul, Turkey

Universities and Colleges attended: Computer Science Engineering (B.Sc), Istanbul University, 2001.
Telecommunication Engineering (M.Sc.), Istanbul Technical University, 2004.

Publications:

Journal Papers

• **Yaslan Y.** and Cataltepe Z., "Co-training with Relevant Random Subspaces", *Neurocomputing*, 73,1652-1661, 2010.

• Peltonen J., **Yaslan Y.**, and Kaski S., "Relevant subtask learning by constrained mixture models", *Intelligent Data Analysis*, Accepted.

• **Yaslan Y.** and Günsel B., "An Integrated On-line Audio Watermark Decoding Scheme for Broadcast Monitoring", *Multimedia Tools and Applications*, vol. 40, 1-21, 2008.

• Cataltepe Z., **Yaslan Y.** and Sonmez A., "Music Genre Classification Using MIDI and Audio Features", *Journal of Applied Signal Processing (ISSN: 1687-6172)*, vol. 2007 (January), Article ID 36409.

Conference Papers

• **Yaslan Y.** and Cataltepe Z., "Random Relevant and Non-redundant Feature Subspaces for Co-training", *10th Int. Conference on Intelligent Data Engineering and Automated Learning - IDEAL 2009*, Burgos, Spain, September 23-26, 2009.

• **Yaslan Y.** and Cataltepe Z., "Audio Genre Classification with Semi-supervised Feature Ensemble Learning", *2nd International Workshop on Machine Learning and Music MML 2009, Conjunction with ECML-PKDD 2009*, Bled, Slovenia, September 7, 2009.

• **Yaslan Y.** and Cataltepe Z., "Audio Genre Classification with Co-MRMR/Müzik Türlerinin Co-MRMR ile Sınıflandırılması",*17th IEEE Conference on Signal Processing and Communications Applications (SIU 2009)*, Antalya, Turkey.

• **Yaslan Y.** and Cataltepe Z., "Co-Training with Adaptive Bayesian Classifier Combination", *ISCIS '08. 23rd International Symposium on Computer and Information Systems*, 27-29 October 2008, Istanbul, Turkey.

• Peltonen J., **Yaslan Y.**, and Kaski S., "Variational Bayes Learning from Relevant Tasks Only", *NIPS 2008 Learning from Multiple Sources Workshop*.

• **Yaslan Y.**, Cataltepe Z.,"A Comparison Framework of Similarity Metrics Used for Web Access Logs Analysis", *International Conference on Machine Learning and Data Mining (MLDM 2007)*, Leipzig Germany.

• **Yaslan Y.**, Cataltepe Z.,"Audio Music Genre Classification Using Different Classifiers and Feature Selection", *18th International Conference on Pattern Recognition (ICPR 2006)*, Hong Kong.

• Kirbiz S., **Yaslan Y.**, Gunsel B.,"Robust Audio Watermark Decoding by Nonlinear Classification", *13th European Signal Processing Conference*, 4-8 September 2005, Antalya/TURKEY.

• Kirbiz S., Gunsel B., **Yaslan Y.**, "Robust Audio Watermarking for Copyright Protection", *WSEAS Transactions on Circuits and Systems*, Issue 10, vol.3, pp. 2132-2138,2004

• **Yaslan Y.**, Gunsel B., "An Integrated Decoding Framework for Audio Watermark Extraction", *17th International Conference on Pattern Recognition (ICPR 2004)*, Cambridge/ENGLAND,Vol.2 pp: 879-882 2004.

• **Yaslan Y.**, Gunsel B.,"Isıl İmgelerden Deniz Hedeflerinin Tespiti", *IEEE 12. Sinyal İşleme ve İletişim Uygulamaları Kurultayı (SIU 2004)*, Kuşadası/TÜRKİYE

• Herkiloğlu K., **Yaslan Y.**, Sener S., Gunsel B., "Uyarlanır Psikoakustik Maskeleme Kullanılarak Gürbüz Ses Damgalama", *IEEE 12. Sinyal İşleme ve İletişim Uygulamaları Kurultayı (SIU 2004)*, Kuşadası/TÜRKİYE

• Gunsel B., Sener S., **Yaslan Y.**, "An Adaptive Encoder for Audio Watermarking", *WSEAS Transactions on Computers*, Issue 4, vol.2, pp. 1044-1048,2003

• **Yaslan Y.**, Gunsel B., "Temel Bileşen Analizi Kullanılarak Isıl İmgelerde Deniz Hedeflerinin Sezilmesi", *11. Sinyal İşleme ve İletişim Uygulamaları Kurultayı*, Istanbul/TÜRKİYE