ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

CLASSIFIER FUSION FOR MULTIMODAL CORRELATED CLASSIFIERS AND VIDEO ANNOTATION

M.Sc. THESIS

Ümit EKMEKÇİ

Department of Computer Engineering

Computer Engineering Programme

MAY 2014

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

CLASSIFIER FUSION FOR MULTIMODAL CORRELATED CLASSIFIERS AND VIDEO ANNOTATION

M.Sc. THESIS

Ümit EKMEKÇİ
(504101540)

Department of Computer Engineering

Computer Engineering Programme

Thesis Advisor: Assoc. Prof. Dr. Zehra ÇATALTEPE

MAY 2014

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

BAĞIMLI SINIFLANDIRICILAR VE VİDEO İŞARETLEME İÇİN SINIFLANDIRICI BİRLEŞTİRME

YÜKSEK LİSANS TEZİ

Ümit EKMEKÇİ
(504101540)

İlgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

Tez Danışmanı: Assoc. Prof. Dr. Zehra ÇATALTEPE

MAYIS 2014

Ümit EKMEKÇİ, a M.Sc. student of ITU Graduate School of Science Engineering and Technology 504101540 successfully defended the thesis entitled **"CLASSIFIER FUSION FOR MULTIMODAL CORRELATED CLASSIFIERS AND VIDEO ANNOTATION"**, which he/she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Assoc. Prof. Dr. Zehra ÇATALTEPE** Istanbul Technical University	.....
Jury Members :	**Assoc. Prof. Dr. Hazım Kemal EKENEL** Istanbul Technical University	.....
	Dr. Aydın ULAŞ Argela A.Ş.	.....
		.....

Date of Submission : **5 May 2014**
Date of Defense : **27 May 2014**

To my family,

FOREWORD

I would like to thank Dr. Zehra ÇATALTEPE for her guidance and support during my graduate studies. I would also like thank to my family for their endless support and love.

May 2014 Ümit EKMEKÇİ

TABLE OF CONTENTS

Page

FOREWORD ..... **ix**
TABLE OF CONTENTS ..... **xi**
ABBREVIATIONS ..... **xiii**
LIST OF TABLES ..... **xv**
LIST OF FIGURES ..... **xvii**
SUMMARY ..... **xix**
ÖZET ..... **xxi**
1. INTRODUCTION **1**
1.1 Eigenclassifiers 2
1.2 REPERE challenge 2
1.3 Contributions 3
2. BACKGROUND and NOTATION **5**
2.1 Notation 5
2.2 Background 5
2.2.1 Principal Component Analysis 5
2.2.2 Kernel Principal Component Analysis 6
3. RELATED WORK **9**
4. EXTENDED MULTIMODAL EIGENCLASSIFIERS and CRITERIA FOR FUSION MODEL SELECTION **13**
4.1 Introduction 13
4.1.1 Variance-Bias Trade off 14
4.2 Eigenclassifiers 15
4.3 Extended Eigenclassifiers with Multimodal Base Classifier Outputs 17
4.3.1 Unimodal Case 17
4.3.2 Recoding of inputs 18
4.3.3 Multimodal Case 19
4.4 Fusion Method Experiments 20
4.4.1 Simple Average 21
4.4.2 Kernelized Extended Multimodal Eigenclassifiers 21
4.4.3 SVMs and Eigen SVMs 21
4.4.4 Dropout 21
4.4.5 AYSU dataset 23
4.4.6 Fusion Experiment Results 24
4.5 Criteria for Fusion Method Selection 27
4.5.1 Average Eigenvalue Distributions and Diversity Metrics 27
4.6 Conclusion 30

5. FUSION FOR VIDEO ANNOTATION 33
5.1 REPERE Dataset 33
5.2 General Information on Speaker Identification Task 33
5.3 Propagation Based Fusion for Unsupervised Speaker Identification Task 34
5.4 Supervised Speaker Identification Task 35
5.4.1 Extracting candidate names from diarization and written names 35
5.4.2 Propagation over similarity graph 35
5.4.3 Overall Algorithm 36
5.4.4 Results and Discussion 36
6. CONCLUSIONS 39
REFERENCES 41
APPENDICES 45
APPENDIX A 47
CURRICULUM VITAE 49

ABBREVIATIONS

AdaBoost : Adaptive Boosting
Bagging : Bootstrap Aggregating
EC : Eigenclassifiers
EGER : Estimated Global Error Rate
KEC : Kernelized Eigenclassifiers
KXMEC : Kernelized Extended Multimodal Eigenclassifiers
MKL : Multiple Kernel Learning
PCA : Principal Component Analysis
SVM : Support Vector Machines
XMEC : Extended Multimodal Eigenclassifiers

LIST OF TABLES

Page

Table 4.1 : Test accuracies of fusion methods on AYSU dataset collection 25
Table 4.2 : Number of experiments each method performed the best..... 25
Table 4.3 : Average rank of each ensemble method..... 26
Table 4.4 : Average eigenvalue distribution 29
Table 4.5 : Average divergence metrics..... 29
Table 5.1 : EGER results on test and development datasets for Supervised Method 37
Table 5.2 : EGER results on test and development datasets for Unsupervised Method 37
Table A.1 : Detailed information on AYSU [1] datasets..... 47

LIST OF FIGURES

Page

Figure 4.1 : The ensemble methods shown in red are significantly different than the ensemble method shown in blue according to Tukey's critical value range shown by the vertical blue line. 26
Figure 4.2 : Variance of estimators for Eigenclassifiers (red) and Extended Multimodal Eigenclassifiers (blue). 26
Figure 4.3 : Basic rules found by Decision Tree. 30

CLASSIFIER FUSION FOR MULTIMODAL CORRELATED CLASSIFIERS AND VIDEO ANNOTATION

SUMMARY

Classifier fusion has become one of the key challenges in machine learning due to the increase in size and structural richness of available data. Thanks to the advances in computing power, we are also able to train many different classifiers; instead of using a single one of them we try to combine them hoping to get better performance. Classifier fusion benefits from classifiers as accurate and as independent as possible. How to generate independent local or base classifiers is a critical question. Adaboost Algorithm of Freund and Schapire (1994) and Bagging Algorithm of Breiman and Leo (1996) aim to create independent base classifiers by using different subsets of inputs generated through sampling for each classifier. Another method, which is used in this thesis, is the Eigenclassifiers approach, proposed by Alpaydın and Ulas in 2012. Eigenclassifiers method aims to create uncorrelated base classifier outputs by mapping to an uncorrelated space. However, for multiclass classification problems, since there are redundant features in the Eigenclassifier transformed classifier output space, they have correlations between them and this causes higher estimator variance and lower prediction accuracy. In this thesis, we extend Eigenclassifiers method to obtain truly uncorrelated base classifiers. We also generalize the distribution on base classifier outputs from unimodal to multimodal, which lets us handle the class imbalance problem.

There are many different classifier fusion methods, and the question of which one to use for a given dataset needs to be answered. In this thesis, we try to answer this question also. We generate a dataset by calculating the performances of nine different fusion methods on 38 different datasets provided by Ulas et. al in 2009. We investigate accuracy-diversity relationship of ensembles on this experimental dataset by using eigenvalue distributions and diversity metrics given by Kuncheva and Whitaker in 2001. We obtain basic rules which can be used to decide on a fusion method given a dataset.

In the second part of the thesis we use classifier fusion for video annotation. We develop a supervised method to combine audio and text information. The proposed method increases the accuracy by about 13 percent over the unimodal methods. This part of the thesis was done as part of a collaborative European Union project called Camomile that brings together researchers from four countries and six institutions together.

BAĞIMLI SINIFLANDIRICILAR VE VİDEO İŞARETLEME İÇİN SINIFLANDIRICI BİRLEŞTİRME

ÖZET

İnternet kullanıcılarının sayısının artması, sosyal iletişim platformu kullanıcılarının artmasına ve böylece her geçen gün internet üzerinde var olan bilgi boyutunun armasına sebep olmaktadır. Ayrıca sosyal platformlardaki yapısal zenginliğin artması, örneğin Facebook'un insanlar arasındaki ilişkileri arkadaşlık bağlantıları sayesinde grafiksel düzeyde, paylaşılan yazılar ve yorumlar sayesinde yazımsal düzeyde ve paylaşılan resimler ve oluşturulan galeriler sayesinde görsel düzeyde araştırmacılara sunması, bu farklı yapıdaki bilgilerin birleştirilebilmesi problemini oldukça önemli bir konu haline getirmektedir. Bu tür veri kümeleri sayesinde, bir sınıflandırma problemini çözmek için değişik veri örnekleri, öznitelik türleri ve sınıflandırma yöntemleri kullanılarak eğitilmiş çok sayıda sınıflandırıcı elde edilebilmektedir. Sınıflandırıcı birleştirme yöntemleri, eldeki sınıflandırıcıları birleştirerek daha iyi başarıma ulaşmayı hedeflemektedir.

Sınıflandırıcıların birleştirilmesi geç birleştirme (late fusion) ya da erken birleştirme (early fusion) yöntemleri ile yapılabilir. Daha sık kullanılan geç birleştirme yönteminde birden fazla yerel sınıflandırıcı çıkışı başka bir sınıflandırıcının eğitilmesi ile birleştirilir. Geç birleştirme yönteminin başarılı olması için gerekli olan önemli bir unsur yerel sınıflandırıcı çıkışlarının birbirlerinden mümkün olduğunca ilintisiz olmasıdır. Çünkü yerel sınıflandırıcıların ilintisiz olması birleştirme için kullanılan sınıflandırıcının varyansının azalmasına, dolayısı ile de başarımının artmasına sebep olmaktadır. Yerel sınıflandırıcılar arasındaki ilintisizlik farklı yollardan elde edilebilir. Örneğin aynı hata fonksiyonunu azaltmayı hedefleyen sınıflandırıcılar farklı girişler üzerinde eğitilebilirler. Boosting ve Bagging algoritmaları bu yöntemin en bilinen örneklerindendirler. Bunun haricinde aynı girişler üzerinde farklı amaç fonksiyonuna sahip sınıflandırıcılar ya da farklı mimariye, parametrelere sahip (örneğin farklı sayıda saklı sinir hücresine sahip yapay sinir ağları gibi) sınıflandırıcılar eğitilerek de sınıflandırıcılar arasında ilintisizlik oluşturulabilir.

Alpaydın ve Ulaş tarafından 2012 yılında önerilen, aynı zamanda bu tezin ilk kısmının temelini oluşturan, Eigenclassifiers (Özsınıflandırıcılar) yöntemi yerel sınıflandırıcı çıkışları arasındaki ilintisizliği doğrusal bir dönüşüm olan *Temel Bileşenler Analizi* (PCA: Principal Component Analysis) dönüşümünü kullanarak gerçekleştirmeyi amaçlamaktadır. Fakat bu dönüşüm kullanılırken çoklu etikete sahip problemlerde, etiketler arasındaki ilişkiler ele alınmadığı için dönüşüm sonucu oluşan özellik yöneyleri tam olarak doğrusal ilintisiz olmamaktadır. Bu durum özellik yöneylerinde fazladan ve gereksiz verinin oluşmasına ve varyansın artmasına, dolayısı ile performansın düşmesine sebep olmaktadır. Bu tez çalışmasının ilk kısmında Eigenclassifiers yöntemi çok sınıflı sınıflandırma problemleri için genişletilerek dönüşüm sonucu elde edilen özellik uzayı doğrusal olarak tam ilintisiz hale

getirilmiştir. Bu sayede, sınıflandırıcı çıkışlarını birleştiren sınıflandırıcı varyansı düşürülerek performans artırılmıştır.

Çok sınıflı sınıflandırma problemlerinde eğer bir sınıfta gözlemlenen örnek sayısı diğer sınıflardakilerden çok fazla ise, hata fonksiyonunu azaltmayı hedefleyen sınıflandırıcılar bütün örnekleri o sınıfa atayabilmektedir. Bu dengesiz örnek-etiket dağılımı problemi Eigenclassifiers yönteminin yerel sınıflandırıcı çıkışlarının çok modlu Gauss dağılımı izlediği varsayılarak tezde çözümlenmiştir.

Verilen bir veri kümesi için hangi sınıflandırıcı birleştirme yönteminin daha uygun olduğu önemli bir sorudur. Bu soruya cevap bulabilmek için, tezde, dokuz farklı sınıflandırıcı birleştirme yönteminin, 38 farklı veri kümesi üzerindeki performansları hesaplanarak, deneyimsel bir veri kümesi oluşturulmuştur. Sınıflandırıcı birleştirme yöntemleri olarak Ortalama, Eigenclassifiers, Extended Multimodal Eigenclassfiers, Dropout, Support Vector Machines (doğrusal ve doğrusal olmayan çekirdekli), Eigen Support Vector Machines, Kernelized Eigenclassifiers ve Kernelized Extended Multimodal Eigenclassifiers kullanılmıştır. Oluşturulan veri kümesi üzerinde Dropout yönteminin en iyi performansı verdiği görülmüştür. Genişletilmiş Eigenclassifiers yöntemi Eigenclassifiers yöntemine göre daha iyi performans göstermiş, çekirdekleştirilmiş yöntemler ise Dropout'tan sonra en iyi sonuçları vermiştir. Oluşturulan veri kümesi üzerinde sınıflandırıcı birleştirme yöntemlerinin doğruluk-ilintisizlikleri, 2001 yılında Kuncheva ve Whitaker tarafından önerilen sınıflandırıcı ilintililik ölçütleri (Q statistics, correlation coefficient ρ, disagreement measure, double-fault measure ve entropy) kullanılarak karşılaştırılmıştır. Ayrıca, tezde bilindiği kadarı ile ilk olarak, ortalama özdeğerler dağılımı kullanılarak da doğruluk-ilintisizlik yorumu yapılmıştır. Bir karar ağacı yardımı ile hangi sınıflandırıcı birleştirme yönteminin uygun olduğuna dair kurallar çıkarılmıştır. Elde edilen ilk sonuçlara göre Destek Vektör Makineleri tabanlı sınıflandırıcı birleştirme yöntemleri doğrusal ilintisi az olan veri kümeleri üzerinde ön plana çıkarken test edilen diğer sınıflandırıcı birleştirme yöntemleri doğrusal ilintisi daha fazla olan veri kümeleri üzerinde ön plana çıkmaktadır. Karar ağacı tarafından çıkarılan kurallara göre en önemli ayırt edici özelliklerin elde edilen özdeğerler ve disagreement measure olduğu görülmektedir.

Tezin ikinci kısmında, video işaretleme (video annotation) için sınıflandırıcı birleştirme yöntemleri kullanılmıştır. Bu kısımda bir Chistera projesi olan, *Collaborative Annotation of multi-modal, multi-lingual and multi-media documents*, CAMOMILE kapsamında çalışmalar yapılmıştır. CAMOMILE projesi üzerinde dört ülkeden altı araştırma grubu çalışmaktadır. Projenin amacı televizyon programlarında kimlerin konuştuğunu ya da kimlerin gözüktüğünü, farklı bilgi kaynaklarını birleştirerek bulmaktır. Projedeki başlıca bilgi kaynakları görüntü, ses ve altyazılardır. Projede kullanılan REPERE veri kümesi iki farklı Fransız kanalından, *BFM TV, LCP*, yedi farklı televizyon programından 30 saat kayıt edilmiş 188 videodan oluşmaktadır. Bu veri kümesi 24 saati eğitim, üç saati geliştirme ve üç saati test olmak üzere üç parçaya ayrılmıştır. Tezde, ses bilgisi ve altyazı bilgisi birleştirilerek hem gözetimsiz (unsupervised) hem de gözetimli (supervised) olarak o anda kimin konuştuğu bulunmaya çalışılmıştır. Ses bilgisi olarak, Camomile proje katılımcısı Claude Barras'ın (LIMSI) ekibi tarafından geliştirilen ve projedeki araştırmacılara sunulan konuşmacıların kümelenmiş fakat etiketlenmemiş (speaker diarization) halleri kullanılmıştır. Altyazı bilgisi olarak ise proje katılımcısı Georges Quénot (LIG-CNRS) tarafından elde edilen, televizyon

programlarının ekranın alt kısmında gösterdikleri, konuşmacıların isimlerini içeren yazıların işlenmesi ile elde edilen konuşmacıların isimleri kullanılmıştır. Böylelikle, video işaretlemede ses ve yazı kullanılarak sınıflandırıcı birleştirmede, elde edilen bölütlenmiş fakat etiketlenmemiş konuşmacı kümeleri ve konuşmacılara ait etiketlerin çıkarıldığı altyazı bilgisi bulunmaktadır. Yöntemler geliştirilirken, özellikle, önceki çalışmalarda başarı göstermiş olan yayılım ve grafik eşleştirme tabanlı algoritmalar üzerinde durulmuştur. Gözetimsiz olarak Bredin tarafından önerilen *term-frequency, inverse document-frequency (TF-IDF)* tabanlı yayılım algoritması kullanılmıştır. Gözetimli yöntemler tasarlanırken konuşmacı tanıma üzerine çıkış üreten 3 farklı sınıflandırıcının çıkışları kullanılmıştır. Bu çıkışlar özellikle yayılım tabanlı benzerlik grafiği oluşturulurken, düğümler arasındaki benzerliğin hesaplanması aşamasında kullanılmıştır. Özellikle yanlış tahmin edilen örneklerin sayısını azaltarak katkı sağlayan bir diğer yöntem ise kendi aralarında aynı konu hakkında konuşan kişilerin bir araya gruplanması ve bu grupların zaman aralıklarına denk gelen altyazılardan isimlerinin çıkartılarak, gruplar için aday isim listelerinin çıkarılmasıdır. Tezde 2014 yılında yayımlanan REPERE test kümesi üzerinde sonuçlar hesaplanmıştır. Elde edilen sonuçlara göre farklı bilgi kaynaklarının birleştirilmesi tek bilgi kaynağı kullanımına göre performansta %13 lük bir artış sağlamıştır. Bunun yanında tezde elde edilen sonuçlar projenin Fransız ortakları tarafından elde edilen sonuçlarla da karşılaştırılmıştır.

1. INTRODUCTION

Every year not only the size of the data, but also the heterogeneous structure of the data gets richer. For example social networks bring graphical representation of the interactions between both people and their behaviors. Also Twitter, Foursquare and other social networks give a lot of textual information to the researchers that was not available before. For a bioinformatic problem protein-protein interactions a researcher can both have a graphical representation of interactions, protein sequences and a gene ontology annotations [2]. Combining these different representations can give huge benefits to the researchers. For video annotation problems our source of information can be the face images, the audio of the people, the subtitles of the speech [3] and the colors of the clothes [4] that people wear. Using these different sources of information to identify a person will clearly increase the robustness and the accuracy. Another kind of problem that fusion helps is the case where there is just one representation of the data but there can be more than one model defined to explain the generative process. In the best case, each model handles one independent property of the process. For example, for a city the monthly temperature change can show different properties over the months. In summer the temperature can increase linearly and smoothly and in spring the temperature change can follow a periodic signal. To model this behavior of the data we can linearly combine the models that we generated. Fusion is generally performed in two levels: *early fusion* or *late fusion*. In the early fusion, features extracted from the different sources of the data are first combined and then sent to a classifier. In the late fusion, first each decision of the independent models are obtained and then using a final classifier, local decisions are combined. The advantage of the early level fusion is the capability to handle the correlation between multiple features from different modalities at an early stage. Also, it requires only one learning phase on the combined feature vector. Advantage of late fusion over the early fusion is that it allows to use the most suitable model for each modality and if local decisions are treated as probabilities they will be on the same scale which requires more work to have the same effect on the early fusion.

This thesis consists of two parts. In the first part we deal with the problems, which have *one representation and multiple base classifiers*. In practice base classifiers are correlated which affects the performance of fusion negatively. Eigenclassifiers [5] is one of the methods that try to decorrelate the base classifiers before combining them with a linear classifier. In the first part we showed how to kernelize the Eigenclassifiers, how to reduce the variance of the final stage estimator and hence improve the prediction accuracy and how to extend the distribution on the data to mixture of Gasussians to handle the imbalance data problem more accurately than Eigenclassifiers. In the second part we deal with a problem which has *multiple representations and one classifier*. We especially focused on the REPERE challenge and tried to identify people in TV broadcast shows by combining text and speech information.

In the following sections we briefly describe Eigenclassifiers [5] and the REPERE challenge. In section 1.3 the contributions of the thesis are given.

1.1 Eigenclassifiers

Eigenclassifiers were proposed by Ulas, Yıldız and Alpaydın in 2012. In practice most of the base classifiers are correlated with each other. One approach is to keep a small subset of base classifiers by reducing the correlated pairs, but if there are correlations between base classifiers, then it is clear that this will cause loss of information. Eigenclassifiers combine base classifiers taking into account that they are not independent. They treat the outputs of base classifiers as a feature vector and find a new uncorrelated feature space which is then combined with a stack classifier. In their work, Ulas, Yıldız and Alpaydın compared their method with AdaBoost [6] and Bagging [7]. They observed that Eigenclassifiers are either more accurate or achieve a comparable accuracy using a fewer number of classifiers.

1.2 REPERE challenge

The REPERE challenge aims to support the development of automatic systems for multimodal person identification. Dataset contains 30 hours of videos taken from two French TV channels with multimodal annotations, i.e speech transcriptions, extracted names from subtitles, video annotations. The dataset mostly contains news and debates. Dataset is divided into three parts, train (24h), development (3h) and test

(3h). There are two main tasks in the challenge, *who is speaking and when?*, *who is seen and when?*. Our contributions and results on 2014 test dataset are given in Chapter 5.

1.3 Contributions

Eigenclassifiers method [5] aims to reduce the correlation between base classifiers by a linear projection of base classifier outputs to a new uncorrelated feature space. As we will see in Chapter 4, Eigenclassifiers method does not use the correlations between class assignments. This causes redundant features to be produced when the test data is mapped using the transformation matrix computed on the training set. In this thesis, in order to avoid redundant features, we adopt the Eigenclassifiers method to use correlation between class assignments and to obtain truly uncorrelated base classifiers. We also relax the unimodal distribution assumption on base classifier outputs in order to handle the class imbalance problem. There are other well known fusion methods and the question of which fusion method should be used for a particular dataset is an important one. In order to answer this question, we generate an experimental database by calculating the results of nine different fusion methods on 38 different datasets used in AYSU dataset [1]. We experiment with the following fusion methods: simple Average, Eigenclassifiers [5], Extended Multimodal Eigenclassifiers, Dropout [8], Support Vector Machines (with linear and RBF kernels), Eigen Support Vector Machines, Kernelized Eigenclassifiers and Kernelized Extended Multimodal Eigenclassifiers. On the experimental dataset, we investigate the relationship between accuracy and diversity of an ensemble to decide on the suitable classifier fusion method for a particular case. We obtain basic rules that show which fusion method works best on a particular dataset. In the second part of the thesis propagation based unsupervised and supervised methods we used in the REPERE challenge are explained. Especially the two proposed methods we focus on, reducing the candidate labels for each diarization and propagation based similarity graph, help to improve performance by decreasing the number of false-positives. We present both our results and our French partners' results on the REPERE test dataset released in 2014.

The rest of the thesis is organized as follows. In Chapter 2, we introduce the notation we use and give some background on the base methods we use. Related work is given

in Chapter 3. Extended Multimodal Eigenclassifiers with strategy for fusion method selection is introduced in Chapter 4. In Chapter 5, multimodal fusion algorithms for video annotation are explained. Conclusions and future work are provided in the last Chapter.

2. BACKGROUND and NOTATION

In this chapter, we first introduce the notation used in the thesis. We also go through the Principal Component Analysis (PCA) and Kernel PCA which is used at the kernelization process of Eigenclassifiers [5]

2.1 Notation

In order to describe our task in more concrete mathematical terms, we introduce the following notation. Vectors are denoted by lower case and bold characters, ex: $\boldsymbol{x}$, Matrices are denoted by upper case and bold characters, ex: $\boldsymbol{X}$ and scalar values are denoted by lower case characters. When we are given a classification problem with K classes, N instances and R trained base classifiers we denote the the base classifier outputs for instance i, $i = 1...N$, by $R \times K$ dimensional matrix $\boldsymbol{X_i} \in \mathbb{R}^{R \times K}$. Each entry in $\boldsymbol{X_i}$, $x_i^{r,k} \in [0:1]$ is the probability value given by classifier r for the k_{th} label. $\Phi(\boldsymbol{x})$ is a non-linear mapping from some low dimensional space to an higher dimensional space and is induced by the decided kernel function $\boldsymbol{K}$. $\|\boldsymbol{x}\|^2$ denotes the vector norm of $\boldsymbol{x}$ and is the same as the dot product $< \boldsymbol{x}, \boldsymbol{x} >$. $\|\boldsymbol{X}\|_F$ denotes matrix norm and can be calculated by $trace(\boldsymbol{X^T X})$. The eigenvalues of a positive definite matrix $\boldsymbol{X}$ are denoted by $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$ and the corresponding eigenvectors are denoted by $v_1, v_2, \ldots, v_n$. $\mathbf{1_n}$ denotes the vector whose all values are 1 and $\mathbf{1}_{n \times n}$ denotes the matrix whose all elements are 1. $\boldsymbol{I}_{n \times n}$ denotes the identity matrix of size $n \times n$.

2.2 Background

2.2.1 Principal Component Analysis

PCA (principal component analysis) is at the heart of the eigenclassifiers, since we will need its formulation for kernelized eigenclassifiers also, we briefly explain PCA below. PCA is an unsupervised dimensionality reduction method. It is a linear mapping that maps the original space to a new space which covers as much of the variance in the data

as possible and giving an uncorrolated direction for each added dimension. We explain PCA from this view of maximum variance formulation. If we assume that we have a set of observations $\{\boldsymbol{x_n}\}$ where $n = 1,\ldots,N$, then our goal is to project the data onto a space where the variance of the projected data is maximum and the dimensionality is less or equal than the original data. If we define $\boldsymbol{W}$ as a projection matrix then the projected data is $\boldsymbol{Y} = \boldsymbol{W^T}(\boldsymbol{X} - \overline{\boldsymbol{X}})$. The variance of the projected data $E[\boldsymbol{Y}\boldsymbol{Y^T}]$ is given by:

$$\boldsymbol{W^T}\boldsymbol{E}[(\boldsymbol{X} - \overline{\boldsymbol{X}})(\boldsymbol{X} - \overline{\boldsymbol{X}})^T]\boldsymbol{W} = \boldsymbol{W^T}\boldsymbol{S}\boldsymbol{W} \tag{2.1}$$

where $\boldsymbol{S}$ is the data covariance matrix of $\boldsymbol{X}$ and $\overline{\boldsymbol{X}}$ is a matrix that consists of the mean vector of the data at each row.

To maximize projected variance $\boldsymbol{W^T}\boldsymbol{S}\boldsymbol{W}$ with respect to $\boldsymbol{W}$ and the constraint $\boldsymbol{W^T}\boldsymbol{W} = \boldsymbol{I}$ (we are only interested in a direction) we introduce a diagonal Lagrange multiplier matrix Λ. Then the objective function to maximize is:

$$\boldsymbol{W^T}\boldsymbol{S}\boldsymbol{W} + \Lambda(\boldsymbol{I} - \boldsymbol{W^T}\boldsymbol{W}) \tag{2.2}$$

The derivative of this function with respect to $\boldsymbol{W}$ is:

$$\boldsymbol{S}\boldsymbol{W} = \boldsymbol{W}\Lambda \tag{2.3}$$

This is a familiar equation where the columns of the $\boldsymbol{W}$ is the eigenvectors of $\boldsymbol{S}$ and digonal elements of Λ are the eigenvalues of $\boldsymbol{S}$. When we multiply both sides of 2.3 with $\boldsymbol{W^T}$ and we get the projected variance as $\boldsymbol{W^T}\boldsymbol{S}\boldsymbol{W} = \Lambda$. Since $\boldsymbol{W^T}\boldsymbol{W} = \boldsymbol{I}$, in order to maximize the variance we should select the eigenvectors which corresponds to largest eigenvalues.

2.2.2 Kernel Principal Component Analysis

For kernel PCA, instead of the original inputs $\boldsymbol{x_n}$ we study with $\phi(\boldsymbol{x_n})$ which are the basis function values. [1] Let Φ be the $n \times m$ matrix of basis function values for the n observed items, so $\Phi_{ik} = \phi_k(x_i)$. Even if $\boldsymbol{X}$ have zero mean probably Φ will not have zero mean. We should centralize the basis matrix as:

$$\overline{\Phi} = [\boldsymbol{I}_{n\times n} - \boldsymbol{1}_{n\times n}/\boldsymbol{n}]\Phi \tag{2.4}$$

[1] For kernel PCA formulation, we follow the notation used in Radford M. Neal's lecture notes in http://www.utstat.utoronto.ca/ radford/sta414.S12/week12.pdf.

where $\boldsymbol{I}_{n\times n}$ is the $n \times n$ identity matrix and $\mathbf{1}_{n\times n}$ is the matrix whose all elements are 1. We can now find eigenvectors of

$$\overline{\Phi\Phi^T} = [\boldsymbol{I}_{n\times n} - \mathbf{1}_{n\times n}/\boldsymbol{n}]\boldsymbol{\Phi}\boldsymbol{\Phi}^{\boldsymbol{T}}[\boldsymbol{I}_{n\times n} - \mathbf{1}_{n\times n}/\boldsymbol{n}] \tag{2.5}$$

Now if we substitute a kernel $K(x,\bar{x})$ instead of $\boldsymbol{\Phi}\boldsymbol{\Phi}^{\boldsymbol{T}}$ then we get a centralized kernel matrix

$$\overline{\boldsymbol{K}} = [\boldsymbol{I}_{n\times n} - \mathbf{1}_{n\times n}/\boldsymbol{n}]\boldsymbol{K}[\boldsymbol{I}_{n\times n} - \mathbf{1}_{n\times n}/\boldsymbol{n}] \tag{2.6}$$

let $\boldsymbol{v_1}, \boldsymbol{v_2}, \ldots, \boldsymbol{v_n}$ be the eigenvectors and $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$ be the eigenvalues, then the projection of a data point $\boldsymbol{x_*}$ on the m'th principal component is

$$[\boldsymbol{k} - \mathbf{1}_{\boldsymbol{n}}^{\boldsymbol{T}}\boldsymbol{K}/\boldsymbol{n}][\boldsymbol{I}_{n\times n} - \mathbf{1}_{n\times n}/\boldsymbol{n}]\boldsymbol{v_m}/\sqrt{\lambda_m} \tag{2.7}$$

where $\boldsymbol{k}$ is the vector of dimension n with $k_i = K(x_*, x_i)$ and $\mathbf{1}_{\boldsymbol{n}}^{\boldsymbol{T}}$ is a raw vector all ones.

3. RELATED WORK

Simple average and weighted average combination are the most well known and frequently used methods for classifier combination. Fumera and Roli [9] in 2005 investigated the theoretical and experimental analysis of these linear combiners. In the case of the weighted average they considered the simplest and the most widely used implementation of weighted average, where a set of nonnegative weights are assigned to each individual classifier. The conclusion they reached was, only for small classifier ensembles, if the individual classifiers exhibit a range of errors with non-negligible width (at least 0.05) and if the outputs of the individual classifiers are highly correlated, then weighted average can perform better than single average with the condition that a suitable validation data are available for optimal weight estimation [9]. The effect of correlation and variance of base classifiers in biometric authentication task is studied by PoH and Bengio [10] in 2005. One of the most important findings was, while positive correlation hurts fusion, greater diversity improves fusion. The other well known methods are minimum, maximum, median and majority voting. Kuncheva [11] performed a theoretical study on these fusion strategies. Minimum/maximum rule was found to be the best for uniformly distributed classifier outputs and for normally distributed outputs the methods gave very similar results. The work assumed the independence of the estimates which is restrictive and unrealistic for most cases. There are ensemble methods that try to overcome this restriction by trying to reduce the dependency among classifiers. ADAboost [6] and Bagging [7] are the two of the well known ones and the Eigenclassifiers [5] method is the one proposed by Ulas, Yıldız and Alpaydın. Performance comparison of Eigenclassifiers with ADAboost and Bagging is given in [5]. There are probabilistic classifier combination methods too. In the simplest case classifier outputs are assumed to be conditionally independent given the true class labels. Ghahramani and Kim [12] proposed three methods to model the correlation between classifier outputs. The first one was to define a hidden variable representing the difficulty of each data point and marginalizing over that variable resulting in a weakly dependent model. The second one was to explicitly

model the pairwise dependencies among classifiers using a Markov Network and the third one was the unification of the two models. They compared their methods with the independent case. SVM based fusion methods are mostly studied in the area of multimedia applications. For example Zhu, Yeh and Cheng [13] offered a fusion framework to classify the images, that have embedded text within their spatial coordinates, by combining visual and textual features with a pair-wise SVM.

The methods we mentioned above are all in the category of late fusion. The other level of fusion is the early fusion where the information is fused at the feature level. Multiple kernel learning (MKL) is one of the successful implementations of early fusion, especially because different information sources such as graphs or texts can be transformed into a common information representation, a kernel, and can be combined by that way. In 2006 Girolami and Zhong [14] adopted gaussian process priors and gave a fully bayesian solution to the problem of optimal combination of covariance functions (kernel functions). Because their model was fully probabilistic, from a bayesian view, inferring the weights of each kernel was nothing but the problem of inferring any unknown parameter. In 2012 Gonen [15] proposed a formulization that is fully conjugate bayesian model and derived a deterministic variational approximation which allowed them to combine hundreds or thousands of kernels very efficiently.

Especially for video annotation and identity detection in TV broadcasts, fusion of different modalities (speech, text and image) holds an important place in the literature. Poignant et. al. [3] proposed a method for unsupervised speaker identification in TV broadcast videos. Their first method was propagation of overlaid names (obtained via OCR) to the speech turns using a variant of the term frequency inverse document frequency (TF-IDF) information retrieval coefficient. Also Poignant et. al. [16] compared the pronounced names modality and written names modality and they concluded that despite a larger number of pronounced names ,speech to text errors and speech transcription errors reduce the potential of this modelity for naming speakers. Bredin et. al. [17] proposed a graph based fusion framework for person identification problem using diarization, written names information. For each video a multimodal probability graph is built and each vertices are connected by an edge weighted by the probability that they correspond to the same person. Person identification is achieved by looking for the maximum probability path between every turn and all available

identities. In 2012, Tapaswi, Bauml and Stiefelhagen [4] searched on identfying charaters in TV series. They aimed at labeling every character appearance, and not only where a face can be detected. They integrated face recognition, clothing appearance, speaker recognition and contextual constraints in a probabilistic manner. For the Big bang Theory dataset they achieved an improvement of 20% for person identification and 12% for face recognition.

4. EXTENDED MULTIMODAL EIGENCLASSIFIERS and CRITERIA FOR FUSION MODEL SELECTION

Diversity among base classifiers is one of the key issues in classifier combination. Although the Eigenclassifiers method proposed by (Ulaş, Yıldız and Alpaydın, 2012), aim to create uncorrelated base classifier outputs, having redundant features in the transformed classifier output space causes higher estimator variance and lower prediction accuracy. In this thesis, we extend Eigenclassifiers method to obtain truly uncorrelated base classifiers. We also generalize the distribution on base classifier outputs from unimodal to multimodal, which lets us handle the class imbalance problem. We also aim to answer the question of which classifier fusion method should be used for a given dataset. In order to answer this question, we generate a dataset by calculating the performances of nine different fusion methods on 38 different datasets. We investigate accuracy-divergence relationship of ensembles on this experimental dataset by using eigenvalue distributions and divergence metrics defined by (Kuncheva and Whitaker, 2001). We obtain basic rules which can be used to decide on a fusion method given a dataset.

4.1 Introduction

Classifier combination allows fusion of different classifiers trained on different modalities, for example visual and audio based classifiers can be combined for better annotation of a video. Even when there is no obvious multimodality, using different features, instance subsets, different types of classifiers or objective functions, we may be able to obtain a set of classifiers whose combination outperforms the best single classifier. Although, in theory, to reduce the variance of the ensemble combination method as much as possible, the combined classifiers should be as diverse as possible [18], in practice, diversity and accuracy of classifiers are competing criteria.

Eigenclassifiers method [5] is one of the proposed methods that aim to reduce the correlation between base classifiers by a linear projection of base classifier outputs to

a new uncorrelated feature space. As we will see in the next section, Eigenclassifiers method does not use the correlations between class assignments. This causes redundant features to be produced when the test data are mapped using the transformation matrix computed on the training set. In this thesis, in order to avoid redundant features, we adopt the Eigenclassifiers method to use correlation between class assignments and to obtain truly uncorrelated base classifiers. We also relax the unimodal distribution assumption on base classifier outputs in order to handle the class imbalance problem. There are other well known fusion methods and the question of which fusion method should be used for a particular dataset is an important one. In order to answer this question, we generate an experimental database by calculating the results of nine different fusion methods on 38 different datasets used in AYSU dataset [1]. We experiment with the following fusion methods: simple Average, Eigenclassifiers [5], Extended Multimodal Eigenclassifiers, Dropout [8], Support Vector Machines (with linear and RBF kernels), Eigen Support Vector Machines, Kernelized Eigenclassifiers and Kernelized Extended Multimodal Eigenclassifiers. The methods Kernelized Eigenclassifiers and Eigen Support Vector Machines are introduced in [19] and to the best of our knowledge, Extended Multimodal Eigenclassifiers and kernelized version are introduced for the first time in this study. On the experimental dataset, we investigate the relationship between accuracy and diversity of an ensemble to decide on the suitable classifier fusion method for a particular case. We obtain basic rules that show which fusion method works best on a particular dataset.

The rest of the thesis is organized as follows. We introduce the notation used in the thesis, and show the relationship between the variance of an estimator and the prediction error in Section 4.1.1. In Section 4.2 and 4.3, we review Eigenclassifiers method of [5] and introduce our method of Extended Multimodal Eigenclassifiers. In Section 4.4, we give the results of 10 different fusion methods on 38 datasets. In Section 4.5, we introduce eigenvalue distributions and also use the diversity metrics defined by [20] to investigate accuracy-diversity relationship of ensembles on the experimental database we generate in Section 4.3. We obtain basic rules that can be used to select a suitable fusion method. Conclusions are given in Section 4.6.

4.1.1 Variance-Bias Trade off

Both Eigenclassifiers method and our Extended Multimodal Eigenclassifiers, use a linear combination of uncorrelated base classifier outputs for classification. Assuming θ is the target value that we try to predict, the expected sum of squares loss can be written as:

$$E_d\left[(\boldsymbol{w}^T\boldsymbol{d}-\theta)^2\right]. \tag{4.1}$$

The expected loss can be decomposed into bias and variance components as:

$$\begin{aligned} & E\left[(\boldsymbol{w}^T\boldsymbol{d}-\boldsymbol{w}^TE[\boldsymbol{d}]+\boldsymbol{w}^TE[\boldsymbol{d}]-\theta)^2\right] && (4.2)\\ &= \underbrace{E\left[(\boldsymbol{w}^T\boldsymbol{d}-\boldsymbol{w}^TE[\boldsymbol{d}])^2\right]}_{\text{Var}} + E[(\underbrace{\boldsymbol{w}^TE[\boldsymbol{d}]-\theta}_{\text{Bias}^2})^2] \\ &= var(\boldsymbol{w}^T\boldsymbol{d})+\text{Bias}^2 \\ &= \boldsymbol{w}^TCov(\boldsymbol{d})\boldsymbol{w}+\text{Bias}^2 && (4.3) \end{aligned}$$

Minimization of (4.3) can be achieved by diagonalizing $Cov(\boldsymbol{d})$ and making $\boldsymbol{w}^T\boldsymbol{w}$ as small as possible, which corresponds to L_2 regularizer. Eigenclassifiers and our Extended Multimodal Eigenclassifiers use this information to create uncorrelated features $\boldsymbol{d}=\boldsymbol{U}^T\boldsymbol{X}\boldsymbol{v}$ whose covariance is a diagonal matrix. The difference between the two methods is the way they treat the vector $\boldsymbol{v}$. Eigenclassifiers use the vector $\boldsymbol{v}_{gt}$ which is previously known from the label information, on the other hand, Extended Multimodal Eigenclassifiers treat $\boldsymbol{v}$ as a vector to be optimized.

4.2 Eigenclassifiers

The key idea of Eigenclassifiers [5] is to create uncorrelated base classifiers that may help to reduce the prediction error by reducing the variance of the estimator. We first express this method using our notation.

Given the transformation matrix $\boldsymbol{U}$ and matrix $\boldsymbol{X}$ which is formed by the outputs of R classifiers for K classes for an instance, we first compute $\boldsymbol{U}^T\boldsymbol{X}\in\mathbb{R}^{R\times K}$. This matrix is flattened by concatenation of its column vectors to form a vector of dimension $R\cdot K$. For multimodal classification for K classes, instead of the weight vector $\boldsymbol{w}$ in Equation 4.3, we need to use a matrix $\boldsymbol{W}\in\mathbb{R}^{R\cdot K\times K}$ to get a linear combination of mapped classifier outputs. Let the operator $\text{Diag}_{\boldsymbol{U}}(\boldsymbol{U}^T\boldsymbol{M}\boldsymbol{U})$, if possible, find the transformation matrix $\boldsymbol{U}$ which transforms matrix $\boldsymbol{M}$ to a diagonal matrix.

We can express the problem of computation of the transformation matrix $\boldsymbol{U}$ as:

$$\underset{\boldsymbol{U}}{\text{Diag}}(\boldsymbol{W}^{\boldsymbol{T}} Cov(\boldsymbol{d})\boldsymbol{W}), \tag{4.4}$$

where $\boldsymbol{d} = \boldsymbol{U}^{\boldsymbol{T}}\boldsymbol{X}\boldsymbol{v}_{gt}$.

The purpose of $\boldsymbol{v}_{gt}$ is to select the column of $\boldsymbol{X}$ which corresponds to the ground truth label. We define the matrix $\boldsymbol{X}_{gt}$ as:

$$\boldsymbol{X}_{gt} = [\boldsymbol{X_1 v}_{gt} ... \boldsymbol{X_i v}_{gt} ... \boldsymbol{X_N v}_{gt}] \tag{4.5}$$

which is the concatenation of columns that correspond to true labels. Let $\boldsymbol{x}^{gt} = \boldsymbol{X}\boldsymbol{v}_{gt}$ be the column $\boldsymbol{gt}$ of $\boldsymbol{X}$. Using the definition of $Cov(\boldsymbol{d}) = E[\boldsymbol{d}\boldsymbol{d}^T]$ and its approximation by the training set, $E[\boldsymbol{X}\boldsymbol{v}_{gt}\boldsymbol{v}_{gt}^{\boldsymbol{T}}\boldsymbol{X}^{\boldsymbol{T}}] = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_{\boldsymbol{i}}^{gt}\boldsymbol{x}_{\boldsymbol{i}}^{gt^T} = \frac{1}{N}\boldsymbol{X}_{gt}\boldsymbol{X}_{gt}^{\boldsymbol{T}}$. Substituting this expected value and $\boldsymbol{d}$ in Equation (4.4), we get:

$$\underset{\boldsymbol{U}}{\text{Diag}}(\boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{U}^{\boldsymbol{T}}E[\boldsymbol{X}\boldsymbol{v}_{gt}\boldsymbol{v}_{gt}^{\boldsymbol{T}}\boldsymbol{X}^{\boldsymbol{T}}]\boldsymbol{U}\boldsymbol{W}) = \underset{\boldsymbol{U}}{\text{Diag}}\left(\frac{1}{N}\boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{U}^{\boldsymbol{T}}\boldsymbol{X}_{gt}\boldsymbol{X}_{gt}^{\boldsymbol{T}}\boldsymbol{U}\boldsymbol{W}\right) \tag{4.6}$$

Clearly, the solution for $\boldsymbol{U}$ is the eigenvectors of $\boldsymbol{X}_{gt}\boldsymbol{X}_{gt}^{\boldsymbol{T}}$.

We give the pseudocode for Eigenclassifiers in Algorithm 1.

Algorithm 1 Pseudo code for Eigenclassifiers [5]

1: $\boldsymbol{X}_{gt} \leftarrow [\,]$ empty matrix
2: **for each** $\boldsymbol{X_i}$ in TrainSet **do**
3: $\quad \boldsymbol{X}_{gt} \leftarrow [\boldsymbol{X}_{gt},\ \boldsymbol{X_i v}_{gt}]$ *//Equation 4.5*
4: **end for**
5: $\boldsymbol{U} \leftarrow eig\left(\boldsymbol{X}_{gt}\boldsymbol{X}_{gt}^{\boldsymbol{T}}\right)$ *//Equation 4.6*
6: $\boldsymbol{X}_{gt} \leftarrow [\,]$
7: **for each** $\boldsymbol{X_i}$ in TrainSet **do**
8: $\quad \boldsymbol{X}_{gt} \leftarrow \left[\boldsymbol{X}_{gt},\ flatten(\boldsymbol{U}^{\boldsymbol{T}}\boldsymbol{X_i})\right]$
9: **end for**
10: $\boldsymbol{W} \leftarrow \arg\min_{\boldsymbol{W}} ||\boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{X}_{gt} - \boldsymbol{Y}||^2 + ||\boldsymbol{W}||_F$
11: **for each** $\boldsymbol{X_i}$ in TestSet **do**
12: $\quad$ assign $\boldsymbol{y_i} \leftarrow softmax(\boldsymbol{W}^{\boldsymbol{T}} flatten(\boldsymbol{U}^{\boldsymbol{T}}\boldsymbol{X_i}))$
13: **end for**

In this algorithm, *flatten()* operator concatenates columns of a matrix to form a column vector. $\boldsymbol{Y}$ are the outputs for the training instances in $\boldsymbol{X}$.

We note that the transformation matrix $\boldsymbol{U}$ is applied to all columns of $\boldsymbol{X}$ on line eight and twelve. However $\boldsymbol{U}$ is found only taking into account the ground truth columns of training instances on line 5. This means $\boldsymbol{U}$ is a valid transformation only for one

column (the ground truth column) of test instance $\boldsymbol{X}$ and the product of $\boldsymbol{U}$ with the other columns of $\boldsymbol{X}$ will generate redundant features which increases the variance of the estimator. In the next section, we will introduce a method that can avoid these redundant features.

4.3 Extended Eigenclassifiers with Multimodal Base Classifier Outputs

In this section, we derive a solution for the transformation matrix $\boldsymbol{U}$ and the weighting vector $\boldsymbol{v}$ based on two different assumptions: i) unimodal case: we assume that $\boldsymbol{X}$ has a unimodal distribution, namely a multivariate Gaussian, ii) multimodal case: we assume that $\boldsymbol{X}$ has a multimodal distribution, namely mixture of multivariate Gaussians.

We show that the multimodal formulation automatically enables handling of the class imbalance problem.

4.3.1 Unimodal Case

In this section, we compute the value of $\boldsymbol{U}$ and $\boldsymbol{v}$ that diagonalizes the covariance in Equation (4.3), assuming that $\boldsymbol{X}$ is unimodal. We aim to minimize $\boldsymbol{w}^{\boldsymbol{T}} Cov(\boldsymbol{d})\boldsymbol{w} +$ bias^2, where $\boldsymbol{d} = \boldsymbol{U}^{\boldsymbol{T}}\boldsymbol{X}\boldsymbol{v}$. The role of vector $\boldsymbol{v}$ is to give weights on columns of $\boldsymbol{X}$. Since the matrix $\boldsymbol{X} \in \mathbb{R}^{R\times K}$ contain the base classifier outputs, for each classifier r and class k, $x^{rk} \in [0:1]$. In the optimistic case, where the base classifiers are mostly correct and correlated, the column which corresponds to the ground truth label will be dominated by values close to 1 and the other columns will have values close to 0. Intuitively, vector $\boldsymbol{v}$ will weight each column proportional to the sum of elements of columns, $v_k \approx \sum_{r=1}^{R} x^{rk}$. The role of $\boldsymbol{U}$ is same as in Eigenclassifiers which is, to generate uncorrelated base classifiers. Problem of variance minimization can now be defined as follows:

$$\underset{\boldsymbol{U},\boldsymbol{v}}{\text{Diag}}(\boldsymbol{W}^{\boldsymbol{T}} Cov(\boldsymbol{d})\boldsymbol{W}) \tag{4.7}$$

$$= \underset{\boldsymbol{U},\boldsymbol{v}}{\text{Diag}}(\boldsymbol{W}^{\boldsymbol{T}} E[\boldsymbol{U}^{\boldsymbol{T}}\boldsymbol{X}\boldsymbol{v}\boldsymbol{v}^{\boldsymbol{T}}\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{U}]\boldsymbol{W}) \tag{4.8}$$

We can factor random matrix $\boldsymbol{X}$ as a product of two vectors $\boldsymbol{X} = \boldsymbol{k}\boldsymbol{p}^{\boldsymbol{T}}$ using one rank approximation [21]. We assume that $\boldsymbol{k} \in \mathbb{R}^R$ is a random vector and $\boldsymbol{p} \in \mathbb{R}^K$ is deterministic. If we substitute $\boldsymbol{k}\boldsymbol{p}^{\boldsymbol{T}}$ in Equation (4.8) we get:

$$\underset{\boldsymbol{U},\boldsymbol{v}}{\text{Diag}}(\boldsymbol{W^T}E[\boldsymbol{U^T k p^T v v^T p k^T U}]\boldsymbol{W}]) \tag{4.9}$$

We can move $\boldsymbol{p^T v v^T p}$ to the end of the equation using the fact that $\boldsymbol{p^T v}$ is a scalar.

$$\underset{\boldsymbol{U},\boldsymbol{v}}{\text{Diag}}(\boldsymbol{W^T}E[\boldsymbol{U^T k k^T U v^T p p^T v}]\boldsymbol{W}) \tag{4.10}$$

Vector $\boldsymbol{k}$ is the only random entity in the equation, so we can move expectation operator inside the brackets as:

$$\underset{\boldsymbol{U},\boldsymbol{v}}{\text{Diag}}(\boldsymbol{W^T U^T}E[\boldsymbol{k k^T}]\boldsymbol{U}(\boldsymbol{v^T p p^T v})\boldsymbol{W}) \tag{4.11}$$

Lets define the eigen decomposition of $E[\boldsymbol{k k^T}]$ and $\boldsymbol{p p^T}$ as $\boldsymbol{\Gamma\Lambda\Gamma^T}$ and $\boldsymbol{\Sigma\Phi\Sigma^T}$ respectively and substitute them:

$$\underset{\boldsymbol{U},\boldsymbol{v}}{\text{Diag}}(\boldsymbol{W^T U^T \Gamma\Lambda\Gamma^T U v^T \Sigma\Phi\Sigma^T v W}) \tag{4.12}$$

Clearly, the solution for $\boldsymbol{U}$ is $\boldsymbol{U} = \boldsymbol{\Gamma}$ and $\boldsymbol{v}$ is the column of $\boldsymbol{\Sigma}$ that corresponds to the largest and only nonzero eigenvalue in $\boldsymbol{\Sigma}$.

We used $\boldsymbol{k p^T}$ as the one rank approximation of $\boldsymbol{X}$, but haven't yet defined the vectors $\boldsymbol{k} \in \mathbb{R}^R$ and $\boldsymbol{p} \in \mathbb{R}^K$ explicitly. We can find these vectors using the singular value decomposition of $\boldsymbol{X}$, $\boldsymbol{X} = \boldsymbol{S\Lambda D}$ and $\boldsymbol{X}$ can be written as:

$$\boldsymbol{X} = \sum_{i=1}^{l} \lambda_i \boldsymbol{s_i} \boldsymbol{d_i^T} = \sum_{i=1}^{l} \boldsymbol{k_i} \boldsymbol{p_i^T} \tag{4.13}$$

where l is the rank of the matrix $\boldsymbol{X}$ and we can write $\boldsymbol{k_i}$ and $\boldsymbol{p_i}$ as:

$$\boldsymbol{k_i} = \sqrt{\lambda_i}\boldsymbol{s_i}, \;\; \boldsymbol{p_i} = \sqrt{\lambda_i}\boldsymbol{d_i} \tag{4.14}$$

If (λ^*,$\boldsymbol{s^*}$,$\boldsymbol{d^*}$) is the triplet corresponding to the largest eigenvalue λ^*, according to Equation (4.14), vectors $\boldsymbol{k}$ and $\boldsymbol{p}$ will take the value:

$$\boldsymbol{k} = \boldsymbol{s^*}\sqrt{\lambda^*}, \;\; \boldsymbol{p} = \boldsymbol{d^*}\sqrt{\lambda^*} \tag{4.15}$$

4.3.2 Recoding of inputs

We can further utilize lower rank approximation by recoding the base classifier outputs $\boldsymbol{X} \in \mathbb{R}^{R \times K}$ as $\begin{pmatrix} \boldsymbol{X(1)} & 0 & 0 \\ 0 & \boldsymbol{X(2)} & 0 \\ 0 & 0 & \boldsymbol{X(3)} \end{pmatrix}$, for a 3 class ($K = 3$) classification problem, where $\boldsymbol{X}(i)$ represents the column i of $\boldsymbol{X}$. This new recoding will save

us from the calculation of the vector $\boldsymbol{v}$. We can write $\boldsymbol{X}$ as $\boldsymbol{X} = \sum_{i=1}^{K} \boldsymbol{X}(i)$, sum of the columns of itself which shows resemblance with the rank summation form of $\boldsymbol{X} = \sum_{i=1}^{K} \boldsymbol{k_i p_i^T}$. Every matrix $\boldsymbol{k_i p_i^T}$ corresponds to one of the columns of $\boldsymbol{X}$, for example $\boldsymbol{k_1 p_1^T}$ generates $\begin{pmatrix} \boldsymbol{X}(1) & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$. If we choose $\boldsymbol{k} = \boldsymbol{s^* \lambda^*}$ instead of $\boldsymbol{s^* \sqrt{\lambda^*}}$ (see Equation (4.15)), vector $\boldsymbol{p}$ must be a unit vector. According to Equation (4.12) $\boldsymbol{v}$ will be a unit vector too and therefore $\boldsymbol{p^T v}$ will be scalar 1. As a result we can avoid the calculation of vector $\boldsymbol{v}$ because the vectors $\boldsymbol{p}$ and $\boldsymbol{v}$ exist only as a dot product $\boldsymbol{p^T v}$ in our calculations. Lets use the full rank decomposition of $\boldsymbol{X}$ and also the fact that $\boldsymbol{p^T v}$ equals 1, in Equation (4.4):

$$\begin{aligned} &\underset{\boldsymbol{U}}{\text{Diag}}(\boldsymbol{W^T} Cov(\boldsymbol{d}) \boldsymbol{W}) \\ &= \underset{\boldsymbol{U}}{\text{Diag}}\, E\left[\left(\boldsymbol{W^T U^T}\left(\sum_{i=1}^{K} \boldsymbol{k_i p_i^T}\right) \boldsymbol{v v^T}\left(\sum_{j=1}^{K} \boldsymbol{k_j p_j^T}\right)^T \boldsymbol{U W}\right)\right] \\ &= \underset{\boldsymbol{U}}{\text{Diag}}\left(\boldsymbol{W^T U^T} E\left[\sum_{i,j=1}^{K} \boldsymbol{k_i k_j^T}\right] \boldsymbol{U W}\right) \end{aligned} \tag{4.16}$$

Solution for the transformation matrix $\boldsymbol{U}$ is the eigenvector of $E\left[\sum_{i,j=1}^{K} \boldsymbol{k_i k_j^T}\right]$. In our implementations we only used rank one approximation of $\boldsymbol{X}$ to reduce the noise in $\boldsymbol{X}$, so in our case $K = 1$ and $\boldsymbol{U}$ equals to the eigenvector of $E[\boldsymbol{k_1 k_1^T}]$.

4.3.3 Multimodal Case

In this section, we compute the values of $\boldsymbol{U}$ and $\boldsymbol{v}$ that diagonalize the covariance in Equation (4.3), assuming that X is multimodal. Expectation and covariance of a random variable x distributed according to mixture of Gaussians can be written as $E[x] = \sum_{i=1}^{K} P_i E[x|c=i]$ and $Cov(x) = \sum_{i=1}^{K} P_i Cov[x|c=i]$, where P_i denotes probability of class i.

If we substitute multimodal definition of covariance and one rank approximation of $\boldsymbol{X}$ in Equation (4.4) and let E_i denote expectation according to the ith class:

$$\underset{U}{\text{Diag}}(\boldsymbol{W^T} Cov(\boldsymbol{d}) \boldsymbol{W}) \tag{4.17}$$

$$= \underset{U}{\text{Diag}}(\boldsymbol{W^T} \sum_{i=1}^{K} P_i Cov[\boldsymbol{d}|c=i] \boldsymbol{W}) \tag{4.18}$$

$$= \underset{U}{\text{Diag}}(\boldsymbol{W^T U^T} \sum_{i=1}^{K} P_i E_i[\boldsymbol{k p^T v v^T p k^T}] \boldsymbol{U W}) \tag{4.19}$$

Since $\boldsymbol{p^T v}$ is scalar 1:

$$= \underset{U}{\text{Diag}} \left(\boldsymbol{W^T U^T} \left(\sum_{i=1}^{K} P_i E_i[\boldsymbol{k k^T}] \right) \boldsymbol{U W} \right) \tag{4.20}$$

Clearly, the solution for $\boldsymbol{U}$ should be the eigenvectors of $\sum_{i=1}^{K} P_i E_i[\boldsymbol{k k^T}]$ and P_i can be estimated using $P_i = N_i/N$, where N_i is the number of instances belong to class i and N is the total number of instances. Pseudocode for the Extended and Multimodal Eigenclassifiers is given in Algorithm 2.

Algorithm 2 Pseudocode for Extended Multimodal Eigenclassifiers

$\boldsymbol{K} \leftarrow 0$
$P_i \leftarrow N_i/N \ \ , i \in [1, \ldots, K]$
for each $\boldsymbol{X_i}$ in TrainSet **do**
  $\boldsymbol{k} = \boldsymbol{s_i^*} \lambda_i^* \ \ , \ \boldsymbol{K} \leftarrow \boldsymbol{K} + P_i \boldsymbol{k k^T}$
end for
$\boldsymbol{K} \leftarrow \boldsymbol{K}/N$
$\boldsymbol{U} \leftarrow eig(\boldsymbol{K})$
$\boldsymbol{T} \leftarrow [\,]$
for each $\boldsymbol{X_i}$ in TrainSet **do**
  $\boldsymbol{k} = \boldsymbol{s_i^*} \lambda_i^* \ \ , \ \boldsymbol{T} \leftarrow [\boldsymbol{T} \ , \boldsymbol{U^T k}]$
end for
$\boldsymbol{W} \leftarrow \arg\min_{\boldsymbol{W}} ||(\boldsymbol{W^T T - Y})||^2 + ||\boldsymbol{W}||_F$
for each $\boldsymbol{X_i}$ in TestSet **do**
  $\boldsymbol{k} = \boldsymbol{s_i^*} \lambda_i^*$
  assign $\boldsymbol{y_i} \leftarrow softmax(\boldsymbol{W^T U^T k})$
end for

4.4 Fusion Method Experiments

In this section, we consider nine late fusion methods which are simple Average, Eigenclassifiers [5], Extended Multimodal Eigenclassifiers introduced in this thesis, Kernelized Eigenclassifiers [19] and Kernelized Extended Multimodal Eigenclassifiers, Support Vector Machines (SVMs) with linear and RBF kernels, Eigen Support

Vector Machines [19] and Droupout [8], a recently popular fusion method usually known as a regularizer. For each fusion method we calculate test accuracies on test data and show our results in Table 4.1. In the next section, we consider the results of these fusion experiments as a new experimental dataset and we infer the relationship between accuracy and diversity of each fusion method to guide us on the selection of the suitable fusion method.

We first give a brief review of the fusion methods we experiment with.

4.4.1 Simple Average

This method simply takes the average of the classifier outputs for each class to be the fusion output. If classifier outputs are uncorrelated, the average method may have reduced variance and hence less expected test error.

4.4.2 Kernelized Extended Multimodal Eigenclassifiers

Because the Kernelized Eigenclassifiers [19] gives better accuracy than Eigenclassifiers [5], we kernelized our Extended Multimodal Eigenclassifiers using the same approach we followed in [19]. Finding linear patterns in a nonlinear feature space with suitable kernels, clearly helps to increase the accuracy on AYSU dataset. The main approach is to adapt the Kernel PCA [22] into the algorithm of Extended Multimodal Eigenclassifiers.

4.4.3 SVMs and Eigen SVMs

Support Vector Machines are popular classifiers which have also been used for late fusion in many applications. We use SVMs in two ways. First we directly give the base classifier outputs as inputs to the SVM after flattening the matrix $\boldsymbol{X}$ to a column vector $col(\boldsymbol{X})$. Secondly the transformed matrix $\boldsymbol{X}_{gt}$, line eight in Algorithm 1, is given to the SVM as an input. Because these inputs are obtained after eigenanalysis, we call this method Eigen SVMs [19].

4.4.4 Dropout

Dropout method, which is usually known as a regularizer, is also a very effective method of combining the predictions of many neural networks with different

architectures [8]. In this method, a smaller random subset of instances, which is called a mini-batch, is used for each iteration of learning. For each mini-batch, outputs of each hidden neuron are set to zero with probability 0.5. This corresponds to training neural networks with different architectures at each mini-batch, while all the weights are shared by all the networks. So if we assume that we have a neural network with one hidden layer and H hidden neurons, we have 2^H different architectures and in each mini-batch, we sample one of them. Sharing the weights is the key point that achieves the regularization in dropout neural networks. Random omission of some of the neurons reduces the dependencies among them in the learning phase. This forces the neurons to adapt their weights without communicating with the omitted neurons, so each architecture learns simple and robust representations or features [23]. When a new test instance is given, all the neurons are used and the outgoing weights of each hidden neuron are multiplied by 0.5. It is stated in [8] that, this operation gives the exact geometric mean when there is one hidden layer and the output layer is softmax and gives a very good approximation to geometric mean when there are more hidden layers.

We follow the learning process described in [23], but with a different learning rate, momentum and mini-batch size settings. We use stochastic gradient descent with 10-150 mini-batches and the cross-entropy objective function. Since, in our case the 38 datasets have different instance sizes, we decide on the mini-batch size according to instance size and performance on the validation set. Our base architecture has one hidden layer with the number of hidden neurons in $\{60, 120, 150, 160\}$ for each dataset. We initialize the weights to small random values drawn from zero mean normal distribution with standard deviation 0.01. We use a linearly increasing momentum with iteration, which is initially 0.7 and 0.99 at the last iteration. Our weight decay parameter is fixed at the value of 0.000001. From our experiments we observe that weight decay parameter is important for minimization of the training error. Proportional to the number of iterations, a linearly decreasing learning rate is used which starts at the value of 0.05 and ends at the value of 0.001. The incoming weight vector corresponding to each hidden neuron is constrained to have a maximum squared length of $L = 25$. If the result of an update exceeds L, the vector is scaled down to a squared length of L. Based on performance on the validation set, we apply to choose

one of $0, 0.1, 0.2$ dropout probabilities on input features and one of $0.2, 0.3, 0.4, 0.5$ dropout probabilities on hidden neurons. The update formulas for weights, learning rate and momentum are as follows:

$$\Delta \boldsymbol{w}^t = -\eta^t \left(\frac{\partial E}{\partial \boldsymbol{w}^t} - 0.000001 \boldsymbol{w}^t \right) + \alpha^t \Delta \boldsymbol{w}^{t-1} \tag{4.21}$$

$$\eta^t = 0.05 - \frac{0.01 - 0.001}{T} t \tag{4.22}$$

$$\alpha^t = 0.7 + \frac{0.29}{T} t \tag{4.23}$$

Here, t denotes the iterations (epochs). Parameter η is used for learning rate and α for momentum. Gradient of the objective function at $\boldsymbol{w}^t$ is $\frac{\partial E}{\partial \boldsymbol{w}}|_{\boldsymbol{w}^t}$ and T is the maximum number of iterations.

4.4.5 AYSU dataset

In our experiments, we use the AYSU [1] dataset, which is a ready to use dataset for model combination. AYSU has been prepared at Boğaziçi University and is based on the datasets from other data repositories. The dataset contains the posterior probabilities of already trained classifiers that can be used in assessment of the classifier combination algorithms. There are 38 datasets and a total of 19 classifiers which have been produced by training nine different algorithms using different parameters. Detailed information on each dataset is given in Appendix A. In this table, train# and test# denote the number of training and test instances respectively. feature# is the input feature dimension size and target# is the number of classes.

The AYSU dataset consists of train-all ($2/3$ of all instances) and test ($1/3$ of all instances) partitions. Each train-all dataset is resampled using 5×2 cross-validation (cv) to generate ten training and validation folds, $train_i, val_i, i = 1, \ldots, 10$. In [5], authors divided validation set into two parts as $valA_i$ and $valB_i$, and they used $valA_i$ to train the linear combiner at the last stage and $valB_i$ for model selection. In our work, we combine $train_i$ and $valA_i$ to form the training set and use $valB_i$ as a validation set. This way, we end up using 1/2 of all the available data for $train_i$, 1/6th for val_i and 1/3rd for $test_i$. We use val_i for early stopping of linear classifier training at the last stage, to tune the penalty factor in SVM, to find the variance parameter of the RBF

kernel, to decide on dropout probability values, to find suitable number of neurons in the hidden layer and to decide on mini-batch sizes.

4.4.6 Fusion Experiment Results

We show test accuracy performances of all fusion methods for 38 datasets in Table 4.1. The results in Table 4.1 will be used as an empirical dataset for fusion method selection in the next section. In Table 4.1, EC and KEC denotes Eigenclassifiers [5] and kernelized version [19] respectively, XMEC and KXMEC denotes Extended Multimodal Eigenclassifiers and kernelized version respectively, SVM(L) denotes Support Vector Machines with linear kernel, E+SVM(R) denotes Eigen Support Vector Machines [19] with RBF kernel. We give a more through comparison of the fusion methods below, but, a first look at Table 4.1, shows that the Dropout method performs the best for most datasets.KXMEC and KEC seems to perform better than XMEC, and XMEC is better than EC. However, in agreement with the results stated in [5], Average seems to perform as well as those four methods.

In order to compare the fusion methods across all the datasets, we perform a number of tests. First, we show pairwise comparison of the ensemble methods in Table 4.2. Each cell entry in Table 4.2 shows the number of data sets on which the algorithm i is the overall winner. Keeping in mind that these results are claimed only for this collection of datasets, we see that the Dropout method has the overall best performance. Combination methods that include SVM perform the worst. On the second row of the table, we compare only the XMEC, KXMEC, KEC and EC methods. The KXMEC and KEC methods perform better than the XMEC and EC. According to the third row of this table, XMEC performs better than EC on more datasets.

We also applied Wilcoxon signed-rank test [24], to see if there is a significant difference between two methods. Wilcoxon signed-rank test rejects the null hypothesis ("the median of the ranking of the differences of performances is 0") at the 6% significance level. In order to compare all ensemble methods on all 38 datasets, we applied nonparametric Friedman test [24], to see if any method is significantly different from other methods. The average rank of each ensemble method is shown in Table 4.3. The found ρ value is smaller than 0.05, which means Friedman test rejects the null hypothesis that all ensembles are the same, so we continue with a post-hoc

Table 4.1: Test accuracies of fusion methods on AYSU dataset collection

	Average	XMEC	KXMEC	EC	KEC	SVM(L)	SVM(R)	E+SVM(L)	E+SVM(R)	Dropout
australian	83.98	83.98	84.84	84.41	85.28	84.41	84.41	83.54	84.41	**87.01**
balance	91.38	98.08	98.08	97.60	98.08	96.65	95.69	98.08	96.17	**99.04**
breast	94.01	94.01	**94.87**	94.01	94.44	91.45	82.47	94.01	94.01	94.02
bupa	68.96	66.89	67.24	63.79	65.51	61.20	62.06	61.20	62.06	**71.55**
car	95.84	96.53	98.26	97.40	99.13	98.96	98.61	98.96	98.78	**99.48**
cmc	**52.23**	**52.23**	52.03	47.56	47.96	43.08	43.08	43.29	42.88	52.03
credit	84.84	84.84	85.71	85.28	86.58	80.95	83.98	80.95	83.98	**87.87**
cylinder	76.11	78.88	78.88	77.22	78.33	76.11	64.44	75.55	63.88	**80.55**
dermatology	95.20	95.20	95.2	**96.80**	96.00	96.00	96.00	96.00	96.00	**96.80**
ecoli	**88.69**	87.13	87.82	86.95	86.95	87.82	86.95	84.34	86.95	85.21
flags	**67.16**	64.17	64.17	58.20	61.19	56.71	58.20	56.71	58.20	50.74
flare	**88.07**	**88.07**	**88.07**	**88.07**	**88.07**	86.23	**88.07**	87.15	87.15	**88.07**
glass	59.45	59.45	59.45	59.45	60.81	60.81	59.45	62.16	59.45	**67.56**
haberman	74.50	73.72	74.50	73.52	74.50	69.60	73.52	70.58	71.56	**75.49**
heart	**86.66**	86.44	**86.66**	85.55	85.55	85.55	74.44	84.44	82.22	82.22
hepatitis	**82.69**	81.15	**82.69**	80.76	80.76	80.76	78.84	80.76	78.84	**82.69**
horse	**88.70**	**88.70**	**88.70**	87.09	87.09	85.48	82.25	86.29	82.25	87.90
ionosphere	87.17	89.91	89.74	89.74	**90.59**	89.74	67.52	**90.59**	80.34	88.88
iris	94.11	94.11	94.11	94.11	**96.07**	94.11	86.27	94.11	90.19	**96.07**
monks	82.63	94.02	97.91	90.27	97.91	97.22	95.83	96.52	95.83	**100.00**
mushroom	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	99.74	**100.00**	99.96	99.92
nursery	99.53	99.59	99.76	99.65	99.76	**99.95**	**99.95**	**99.95**	**99.95**	99.93
optdigits	98.43	98.35	98.35	97.80	98.20	98.43	98.43	**98.51**	**98.51**	98.35
pageblock	96.05	96.20	96.44	97.09	96.77	96.22	96.00	95.84	96.22	**97.42**
pendigits	99.20	99.24	99.32	99.24	99.32	99.32	99.32	99.32	99.32	**99.44**
pima	75.09	75.01	75.09	75.09	75.09	69.64	65.75	69.64	67.31	**76.65**
ringnorm	95.25	98.21	98.50	98.21	**98.58**	98.50	91.12	98.46	90.47	97.69
segment	95.06	95.97	96.49	96.62	97.27	97.01	96.36	95.32	96.49	**97.66**
spambase	93.61	93.78	93.87	94.00	94.00	91.33	92.83	85.92	92.63	**94.46**
tae	55.76	59.23	61.53	55.76	61.53	57.69	48.07	65.38	40.38	**67.30**
thyroid	98.18	98.18	**99.28**	98.18	98.28	98.28	98.28	98.28	98.28	98.50
tictactoe	99.06	99.37	**99.68**	99.37	**99.68**	99.37	99.37	99.37	99.37	99.37
titanic	80.65	80.65	80.65	80.65	80.65	80.51	80.65	80.65	80.65	**80.92**
twonorm	**97.56**	**97.56**	**97.56**	97.40	97.48	96.79	94.85	96.55	95.74	**97.56**
vote	95.86	95.86	95.86	95.17	95.86	93.79	91.03	94.48	92.41	**97.24**
wine	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	98.33	**100.00**	98.33	**100.00**
yeast	60.04	56.94	59.23	58.23	59.03	51.00	52.00	51.00	52.81	**61.04**
zoo	**100.00**	94.59	94.59	91.89	94.59	97.29	89.18	94.59	83.78	**100.00**

Table 4.2: Number of experiments each method performed the best

	Average	XMEC	KXMEC	EC	KEC	SVM(L)	SVM(R)	E+SVM(L)	E+SVM(R)	Dropout
All	11	6	10	4	7	3	2	5	2	25
Eigen		12	27	8	23					
XMEC vs EC		27	21							

Tukey's test to determine which pairs of ensembles are significantly different, and which are not. Figure 4.1 shows the average ranks of each ensemble and the range between vertical green dots is the critical value according to Tukey's test. Figure 4.1 shows that, Extended Multimodal Eigenclassifiers and kernelized version, Kernelized Eigenclassifiers, Dropout and Dropout + ES (early stop) methods (shown in red) are significantly different from the least accurate ensemble method SVM(R) (shown in blue). The Dropout method is significantly different from the Average method, while the other methods are not.

We developed the XMEC ensemble method so that the estimator variance would be reduced. In Figure 4.2, we compare variances of estimators according to Equation

Table 4.3: Average rank of each ensemble method

Average	XMEC	KXMEC	EC	KEC	SVM(L)	SVM(R)	E+SVM(L)	E+SVM(R)	Dropout + ES	Dropout
6.18	6.03	4.21	6.40	4.27	6.71	8.40	6.88	8.10	5.43	3.34

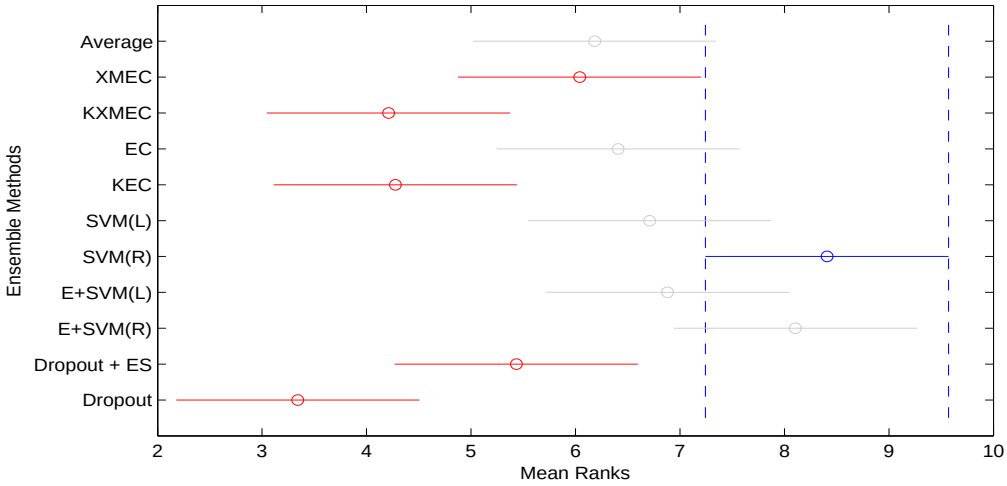

Figure 4.1: The ensemble methods shown in red are significantly different than the ensemble method shown in blue according to Tukey's critical value range shown by the vertical blue line.

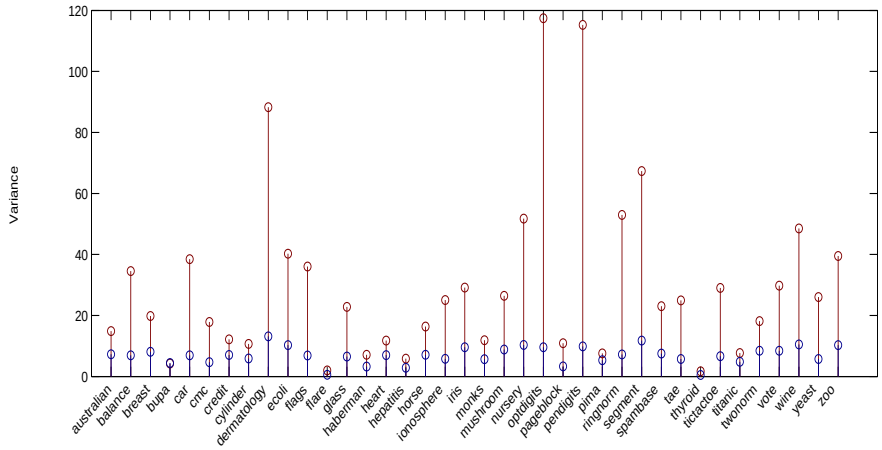

Figure 4.2: Variance of estimators for Eigenclassifiers (red) and Extended Multimodal Eigenclassifiers (blue).

(4.4), for Eigenclassifiers (EC) and Extended Multimodal Eigenclassifiers (XMEC). We show the variances on 38 datasets from AYSU [1]. The variance is computed according to $\mathbf{1}^T Cov(\boldsymbol{d})\mathbf{1} = \mathbf{1}^T Cov(\boldsymbol{U}^T \boldsymbol{X} \boldsymbol{v})\mathbf{1}$, where $\mathbf{1}$ is a vector whose all elements are 1s. Note that this expression corresponds to summing all entries in the covariance matrix Cov(d). According to this figure, the variances for the XMEC method are lower than the variances for the EC method, which could be the explanation for the better performance of the XMEC method.

4.5 Criteria for Fusion Method Selection

In the previous section, we presented accuracy results obtained using different fusion methods. In this section, based on a number of measurements on the available base classifier outputs, we aim to determine the suitable fusion method(s) for a particular dataset.

Previous studies [11,20] considered diversity and accuracy of classifiers to characterize the classifier ensemble used for classifier fusion. We consider five different diversity measures, namely, Q statistics, correlation coefficient ρ, disagreement measure, double-fault measure and the entropy [20]. In this thesis, in addition to these measures, we introduce a metric, which is based on the distribution of the eigenvalues of the output matrix of base classifiers.

Table 4.1 shows that Eigenclassifiers method works best for Dermatology, Flare, Mushroom and Wine datasets, so we can use the average of the diversity measures of these datasets as a representation of Eigenclassifiers method, and the same is true for average eigenvalue distribution representation.

4.5.1 Average Eigenvalue Distributions and Diversity Metrics

If the base classifier outputs are highly correlated, we can approximate the output matrix with a fewer eigenvalue, eigenvector pair and as the number of pairs reduce, the first normalized eigenvalue will be closer to 1, and if the base classifiers are not correlated eigenvalues will be distributed more uniformly. We are going to use this observation first to identify the features of ensemble methods and secondly we are going to use eigenvalues of each dataset with the diversity metrics defined below as an input to a decision tree to see if we can learn rules that associates diversity and the accuracy of ensemble methods.

We use 5 diversity metrics, namely, Q statistics, correlation coefficient ρ, disagreement measure, double-fault measure and the entropy, as defined in (Kuncheva et al. 2001) [20]. Please see [20] for more details on these metrics.

When comparing the classifiers i and k, let $N^{00}, N^{11}, N^{01}, N^{10}$ be the number of instances for which both classifiers were wrong, both classifiers were right, classifier i was wrong and k was right, and classifier i was right and k was wrong, respectively. Let L be the number of base classifiers and define $y_{j,i} = 1$, if the base classifier i correctly recognizes the instance x_j. $I(x_j) = \sum_{i=1}^{L} y_{j,i}$ is the number of classifiers that correctly recognize x_j.

Based on these definitions, diversity metrics are defined as follow:
Q statistics:

$$Q_{i,k} = \frac{N^{11}N^{00} - N^{01}N^{10}}{N^{11}N^{00} + N^{01}N^{10}} \tag{4.24}$$

$$Q_{av} = \frac{2}{L(L-1)} \sum_{i=1}^{L-1} \sum_{k=i+1}^{L} Q_{i,k} \tag{4.25}$$

Correlation coefficient ρ:

$$\rho_{i,k} = \frac{N^{11}N^{00} - N^{01}N^{10}}{\sqrt{(N^{11}+N^{10})(N^{01}N^{00})(N^{11}+N^{01}(N^{10}+N^{00}))}} \tag{4.26}$$

$$\rho_{av} = \frac{2}{L(L-1)} \sum_{i=1}^{L-1} \sum_{k=i+1}^{L} \rho_{i,k} \tag{4.27}$$

Disagreement measure:

$$Dis_{i,k} = \frac{N^{01} + N10}{N^{11} + N^{10} + N^{01} + N^{00}} \tag{4.28}$$

$$Dis_{av} = \frac{2}{L(L-1)} \sum_{i=1}^{L-1} \sum_{k=i+1}^{L} Dis_{i,k} \tag{4.29}$$

Double-fault measure:

$$DF_{i,k} = \frac{N^{00}}{N^{11} + N^{10} + N^{01} + N^{00}} \tag{4.30}$$

$$DF_{av} = \frac{2}{L(L-1)} \sum_{i=1}^{L-1} \sum_{k=i+1}^{L} DF_{i,k} \tag{4.31}$$

Entropy measure:

$$E = \frac{1}{N} \sum_{j=1}^{N} \frac{1}{(L - \lceil L/2 \rceil)} min(I(x_j), L - I(x_j)) \tag{4.32}$$

In order to identify the features of ensemble methods, first we note the datasets on which each method gave the best results using Table 4.1 and then we average the eigenvalue distributions of each group of datasets. We also followed the same

Table 4.4: Average eigenvalue distribution

Average	XMEC	KXMEC	EC	KEC	SVM(L)	SVM(R)	E+SVM(L)	E+SVM(R)	Dropout
62.44	56.01	62.55	58.60	53.61	44.13	53.29	46.20	38.56	60.01
12.88	16.04	13.11	12.68	16.00	19.04	17.13	19.00	23.16	13.83
7.35	9.56	7.71	10.09	8.34	12.79	10.18	10.61	13.22	8.08
4.74	5.36	4.89	5.41	5.79	7.89	5.95	6.50	7.23	4.92
3.30	3.45	3.50	3.93	4.34	5.20	4.06	4.90	5.39	3.38
2.23	2.22	2.29	2.22	3.08	2.87	2.36	2.95	3.01	2.51
1.88	1.72	1.69	1.79	2.35	2.16	1.86	2.28	2.20	1.80
1.41	1.30	1.18	1.23	1.70	1.65	1.39	1.96	1.94	1.36
0.99	1.02	0.85	1.10	1.28	1.38	1.16	1.41	1.35	1.08

procedure to identify the features of ensemble methods using diversity metrics instead of eigenvalue distributions. Average eigenvalue distributions and diversity metrics of each fusion method is given in Table 4.4 and Table 4.5. According to these tables and the empirical dataset we studied on, three methods SVM(L), E+SVM(L) and E+SVM(R) differs from other methods on non-correlated datasets (more uniform distribution on the first two eigenvalues also, lower Q statistics and Correlation coefficient). We also used a decision tree to learn simple rules that associates diversity and accuracy of ensemble methods. Diversity is defined by eigenvalue distributions and the diversity metrics defined above. Accuracy is defined by normalizing each row of Table 4.1 into range [0-1] and specifying a threshold, which we selected 0.8, to turn Table 4.1 into a label matrix. We measured the number of mis predictions by leave-one-out cross-validation as our evaluation method. The performance averaged over 38 datasets was two misprediction among eleven methods. The rules extracted by the Decision Tree are given in Fig 4.3.

Table 4.5: Average divergence metrics

	Average	XMEC	KXMEC	EC	KEC	SVM(L)	SVM(R)	E+SVM(L)	E+SVM(R)	Dropout
Q statistic	0.588	0.514	0.630	0.357	0.505	0.126	0.577	0.336	0.339	0.595
Correlation coeff	0.370	0.355	0.404	0.292	0.292	0.082	0.446	0.144	0.082	0.346
Disagreement	0.165	0.110	0.107	0.110	0.151	0.172	0.155	0.173	0.257	0.210
Double-fault	0.065	0.031	0.037	0.029	0.036	0.014	0.058	0.018	0.020	0.100
Entropy	0.221	0.133	0.136	0.130	0.195	0.210	0.192	0.212	0.312	0.288

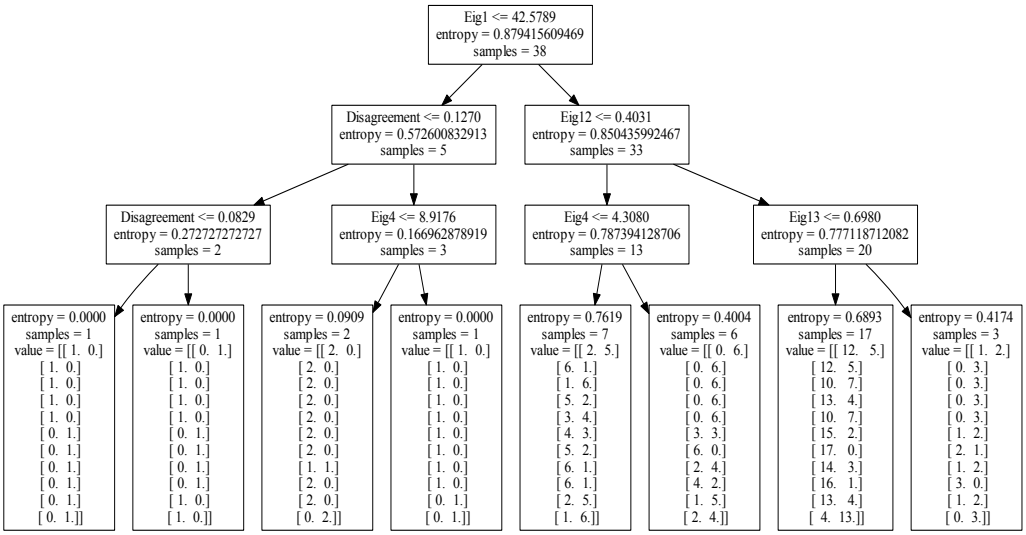

Figure 4.3: Basic rules found by Decision Tree.

4.6 Conclusion

In this thesis, we extended Eigenclassifiers [5] to avoid creating redundant features by using the correlation between class assignments and also generalized the assumption of unimodal distribution on base classifier outputs to mixture of gaussians to handle the imbalance class distribution problem better than Eigenclassifiers. We showed that Extended Multimodal Eigenclassifiers have lower variance and also they are more accurate than Eigenclassifiers [5]. We also generated an empirical dataset to be able to answer the question of, which fusion method is more suitable for a given dataset? The empirical dataset is constructed by calculating the accuracies of 38 datasets on eleven fusion methods. Dropout [8] and kernelized version of both Extended Eigenclassifiers and Eigenclassifiers were significantly successful than other methods. The reason behind the success of Dropout is the regularization effect where we can increase the complexity of the network without the danger of overfitting the data. But the disadvantage of using complex neural networks is the absence of an efficient method to tune the learning rate, momentum and weight decay parameters which are important for Dropout method [25]. We also investigated accuracy-diversity relationship of fusion methods on 38 dataset which led us to select a suitable fusion method for a given dataset. The accuracy-diversity relationship is investigated both by using the metrics defined by [20] and a new metric which is based on average eigenvalue distribution. An

important observation according to the defined metrics, was the different behavior of both SVM and Eigen SVM methods on non-correlated datasets where they were more accurate than other methods. We also achieved to predict which ensemble methods we should use when given the eigenvalue distributions and diversity metrics of any dataset by extracting the rules given by the decision tree in Figure 4.3.

5. FUSION FOR VIDEO ANNOTATION

This year we took part in the REPERE [17] challenge which brings different communities (face recognition, speaker identification, optical character recognition and named entity detection) together for the purpose of multimodal person recognition in TV broadcasts. There are two tasks in the challenge which are *who is speaking when?* and *who is seen when*? We worked only on the first task. The rest of the sections are as follows: In Section 5.1, we explained the components of the REPERE dataset briefly. In Section 5.2, we gave general information on which modalities we used and explained briefly the methods we used for both unsupervised and supervised speaker identification task. In the following sections we gave detailed information on the methods we used. In section 5.4.4 we presented our results and discussions on the task.

5.1 REPERE Dataset

REPERE video corpus [26] (training, development and test sets) contains 188 videos with 30 hours and seven different shows from French TV channels BFM TV and LCP. There are four modalities available for the data. First modality is the names which are extracted from subtitles and called *written names modality*. The second modality is called *spoken names modality* and constructed by a speech to text program. The third modality is the speech of a person and used for monomodal speaker classification task. The last modality is the image of a person and used for monomodal face classification task. There are three monomodal classifiers trained for speaker identification. Also the speaker diarization (speech clustering) and face clustering files are available.

5.2 General Information on Speaker Identification Task

In order to perform fusion of different modalities for both unsupervised and supervised speaker identification task, we utilized diarization outputs, written names modality and audio classifier outputs. The person identification task was considered as an

assignment problem both for supervised and unsupervised cases. For unsupervised case we implemented the algorithm based on TF-IDF measure which is described at [3]. For the supervised case first we sampled the audio clusters to group the people who are in the same conversation using the speaker diarization file. This process resulted in a reduction in the number of candidate names for a certain person. Afterwards, the supervised speaker classifier outputs were scanned and if the name suggested by them was among the candidate names, the person was assigned to be that person. Then we used the unsupervised method (tf-idf) for person-name assignment, again if the name was among the candidate names. At the next step, we produced similarity graph between people, based on the supervised audio classifier outputs. Using this graph we propagated names, similar to the approaches before, assigning the name if it is among the candidate names. Details of the algorithms and performances of these methods are given in the following sections.

5.3 Propagation Based Fusion for Unsupervised Speaker Identification Task

According to the analysis [17] on the dataset the following two observations are made available:

1. In the time interval of a speaker cluster if there is only one name written on the screen then it is very likely(%95 *precision*) that the speaker cluster is the person uttered by this name.

2. There can be oversegmented speaker clusters produced by the speaker diarization system

Matching speaker clusters with written names has two steps. First speaker clusters co-occurring with only one name is directly matched then the remaining speaker clusters are matched with the names which maximize the objective criteria given below.

$$f(s) = argmax_{n \in N} TF(s,n) \cdot IDF(n) \tag{5.1}$$

$$TF(s,n) = \frac{\text{duration of } n \text{ in cluster } s}{\text{total duration of all names in cluster } s} \tag{5.2}$$

$$IDF(n) = \frac{\text{\#speaker clusters}}{\text{\#speaker clusters co-occurring with } n} \tag{5.3}$$

5.4 Supervised Speaker Identification Task

5.4.1 Extracting candidate names from diarization and written names

The aim of this process is to group the speakers, who contribute to the same discussion and extracting the names that are in the interval of this speaker group by using written names file. As a result when we are given a speaker cluster from diarization file we can specify the group which the speaker belongs to and we can extract the candidate names for that speaker cluster. The first thing we manage by doing this process is to reduce the candidate names for a speaker cluster and secondly we can assign the identity to speaker cluster only from the candidate names or we can give high confidence to the candidate names among other names.

Pseudo code of the algorithm is given below

Algorithm 3 Candidate Names

```
1:  speaker interval ← empty map
2:  for each speaker in Tv show do
3:      speaker interval[speaker] ← [t1,t2]
          //t1: first time the speaker speaks, t2: last time the speaker speaks
4:  end for
5:  groups[0] ← speaker interval[0]
6:  groupid ← 0
7:  for each speaker in speaker interval do
8:      if speaker intersects with any groups then
9:          add speaker to this groups
10:     else groupid ← groupid + 1,  groups[groupid] ← speaker
11: end for
12: candidate names ← empty map
13: for each group in groups do
14:     candidate names[group] ← {names that intersect with group}
15: end for
```

5.4.2 Propagation over similarity graph

We construct a similarity graph in which the nodes represent speakers and the connections represent similarity. The connection weights (similarities) are found by counting the number of names that are agreed on by supervised classification algorithms. Ex. Lets assume for speaker1, classifier A gave "name A", classifier B gave "name C" and classifier C gave "name T", and for speaker 2, classifiers gave "name C, name D, name A" respectively, then $sim(speaker1, speaker2) = 2$. To assign

a name to a speaker cluster using similarity graph we followed the procedure given below.

Algorithm 4 Similarity Graph

1: select a speaker sp that is not assigned yet
2: **for each** speaker in the $group$ of sp **do**
3: rank the speakers according to $sim(sp, speaker)$
4: $sp_{name} \leftarrow speaker_{name}$ **if** name is in $candidate\ names$ **else** goto next speaker
5: **end for**

5.4.3 Overall Algorithm

The Pseudo code of the algorithm is given below.

Algorithm 5 Overall Algorithm

1: **//Assignment using supervised classifiers**
2: **for each** speaker which is not assigned yet **do**
3: **if** all supervised classifiers give the same output **and** output is in $candidate\ names$
4: $speaker_{name} \leftarrow name$
5: **end for**
6: **//Assignment using TF-IDF**
7: **for each** speaker which is not assigned yet **do**
8: assign using unsupervised method TF-IDF based propagation **if** proposed name is in $candidate\ names$
9: **end for**
10: **//Assignment using similarity graph**
11: **for each** speaker which is not assigned yet **do**
12: assign using similarity graph
13: **end for**
14: **for each** speaker which is not assigned yet **do**
15: assign the name **if** the name is the only name in the $candidate\ names$
16: **end for**

5.4.4 Results and Discussion

The evaluation criteria is called Estimated global error rate (EGER) and defined below.

$$EGER = \frac{\#fa + \#miss + \#conf}{\#total} \tag{5.4}$$

$\#total$ is the number of person utterances to be detected, $\#fa$ is the number of false alarms, $\#miss$ is the number of missed utterances and $\#conf$ is the number of utterances wrongly identified. We showed both our results and our French partner's results [27] in Table 5.1 and 5.2. According to test dataset released in 2014 multimodal supervised

method is more accurate about 13% over the monomodal method. The actual reason of this improvement is the habit of TV debate shows giving the names of the speakers under the screen as they start to speak. Because annotating each person is a time consuming job and there are lots of labels, there is a high probability that monomodal supervised classifier haven't been trained on every person that is present in TV shows. In that case extracting the labels from subtitles and matching them with diarizitaion information highly increases the number of true-positives. When we look at 5.4 other important step is to decrease the number of false alarms and confusions. To do that we need to decrease the number of false-positives and true-negatives. With the algorithm we defined in Algorithm 3, we achieved to obtain very small, averagely three to four, number of candidate names for each speaker cluster. While assigning the names offered by both monomodel supervised classifiers and similarity graph propagation methods, we checked if the offered name is in the extracted candidate name list for that speaker cluster. By this method we decreased the number of misassigned names and managed to improve EGER criteria. The over-clustering errors, segmenting the time interval into pieces that belongs to unique person, caused by the diarization system, is solved by Algorithm 4. Because each cluster is represented by a node on the graph and the similarity between nodes that are over-segmented are higher than any other nodes, propagating the names between similar nodes fixes the problem of over-segmentation.

Table 5.1: EGER results on test and development datasets for Supervised Method

	Our Method	Ref [27]	MonoSpeaker
Development set	% 20.2	% 19.7	%37.5
Test set	% 23.3	%20.7	%36.6

Table 5.2: EGER results on test and development datasets for Unsupervised Method

	Our Method	Ref [27]
Development set	% 36.2	% 35.6
Test set	% 46.3	% 44.0

6. CONCLUSIONS

In this thesis we studied two different aspects of classifier fusion. In the first part of the thesis, we extended Eigenclassifiers [5] for multiple classes so that creation of redundant features is avoided. We also generalized the assumption of unimodal distribution on base classifier outputs to mixture of Gaussians to handle the unbalanced class distribution problem. We also investigated accuracy-diversity relationship of fusion methods on 38 datasets which helped us to select a suitable fusion method for a given dataset. We showed that Extended Multimodal Eigenclassifiers have lower variance and also they are more accurate than Eigenclassifiers [5]. In order to answer the question of which fusion method is more suitable for a given dataset, we also generated an empirical dataset. This empirical dataset is constructed by calculating the accuracies of 38 datasets on nine fusion methods. Dropout [8] and kernelized version of both Extended Eigenclassifiers and Eigenclassifiers were significantly more successful than the other methods. The reason behind the success of Dropout is the regularization effect which allows an increase in the complexity of the network without the danger of overfitting the data. But the disadvantage of using complex neural networks is the absence of an efficient method to tune the learning rate, momentum and weight decay parameters which are important for Dropout method [25]. In order to decide on an ensemble method, the accuracy-diversity relationship was investigated both by means of using the metrics defined by [20] and a new average eigenvalue distribution based metric we proposed. An important observation according to the defined metrics was that both SVM and Eigen SVM methods were more accurate than other methods on uncorrelated datasets. We also tried to predict which fusion method should be used for given eigenvalue distributions and diversity metrics for any dataset by extracting the rules with the help of a decision tree. According to the decision tree the first eigenvalue among other eigenvalues and the disagreement metric among other metrics are the most discriminative ones.

In the second part of the thesis, we developed several algorithms to fuse different modalities for the REPERE video annotation challenge. The aim of the challenge is to annotate persons using video, speech and text information. According to the test dataset released in 2014, our multimodal supervised method is about 13 per cent more accurate than the monomodal method. The underlying actual reason of this improvement is because the dataset contains TV debate shows where the names of the speakers are shown under the screen as they start speaking. Since annotating each person is a time consuming job and there are lots of labels, there is a high probability that monomodal supervised classifiers haven't been trained on every person present in TV shows. In that case extracting the labels from subtitles and matching them with diarization information highly increases the number of true-positives. In order to improve the EGER criterion which has been used for measuring the performance of video annotation systems, the number of false alarms and confusions also have to be decreased. We obtained very small number, on average three to four, candidate names for each speaker cluster using the algorithm we defined in Algorithm 3. While assigning the names proposed by both monomodal supervised classifiers and similarity graph propagation methods, we checked if the proposed name was in the extracted candidate name list for that speaker cluster. By this method we decreased the number of incorrectly assigned names and managed to improve the EGER criterion. The over-clustering errors, segmenting a time interval that belongs to a unique person into a number of pieces, caused by the diarization system, is solved by Algorithm 4. Because each cluster is represented by a node on the graph and the similarity between nodes that are over-segmented are higher than any other nodes, propagating the names between similar nodes fixed the problem of over-segmentation.

REFERENCES

[1] **Ulaş, A.**, **Semerci, M.**, **Yıldız, O.T. and Alpaydın, E.** (2009). Incremental construction of classifier and discriminant ensembles, *Information Sciences*, ***179***(9), 1298–1318.

[2] **Ben-Hur, A. and Noble, W.S.** (2005). Kernel methods for predicting protein–protein interactions, *Bioinformatics*, ***21***(suppl 1), i38–i46.

[3] **Poignant, J.**, **Bredin, H.**, **Le, V.B.**, **Besacier, L.**, **Barras, C.**, **Quénot, G.** *et al.* (2012). Unsupervised Speaker Identification using Overlaid Texts in TV Broadcast, *Proceedings of the 13th Annual Conference of the International Speech Communication Association (Interspeech)*.

[4] **Tapaswi, M.**, **Bauml, M. and Stiefelhagen, R.** (2012). "Knock! Knock! Who is it?" probabilistic person identification in TV-series, *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, IEEE, pp.2658–2665.

[5] **Ulaş, A.**, **Yıldız, O.T. and Alpaydın, E.** (2012). Eigenclassifiers for combining correlated classifiers, *Information Sciences*, ***187***, 109–120.

[6] **Freund, Y.**, **Schapire, R.E.** *et al.* (1996). Experiments with a new boosting algorithm, *ICML*, volume 96, pp.148–156.

[7] **Breiman, L.** (1996). Bagging predictors, *Machine learning*, ***24***(2), 123–140.

[8] **Hinton, G.E.**, **Srivastava, N.**, **Krizhevsky, A.**, **Sutskever, I. and Salakhutdinov, R.R.** (2012). Improving neural networks by preventing co-adaptation of feature detectors, *arXiv preprint arXiv:1207.0580*.

[9] **Fumera, G. and Roli, F.** (2005). A theoretical and experimental analysis of linear combiners for multiple classifier systems, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, ***27***(6), 942–956.

[10] **Poh, N. and Bengio, S.** (2005). How do correlation and variance of base-experts affect fusion in biometric authentication tasks?, *Signal Processing, IEEE Transactions on*, ***53***(11), 4384–4396.

[11] **Kuncheva, L.I.** (2002). A theoretical study on six classifier fusion strategies, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, ***24***(2), 281–286.

[12] **Ghahramani, Z. and Kim, H.C.** (2003). Bayesian classifier combination.

[13] **Zhu, Q.**, **Yeh, M.C. and Cheng, K.T.** (2006). Multimodal fusion using learned text concepts for image categorization, *Proceedings of the 14th annual ACM international conference on Multimedia*, ACM, pp.211–220.

[14] **Girolami, M. and Zhong, M.** (2007). Data Integration for Classification Problems Employing Gaussian Process Priors, *Advances in Neural Information Processing Systems: Proceedings of the 2006 Conference*, volume 19, The MIT Press, p.465.

[15] **Gönen, M.** (2012). Bayesian Efficient Multiple Kernel Learning, *Proceedings of the 29th International Conference on Machine Learning*.

[16] **Poignant, J.**, **Besacier, L.**, **Le, V.B.**, **Rosset, S. and Quénot, G.** (2013). Unsupervised naming of speakers in broadcast TV: using written names, pronounced names or both?

[17] **Bredin, H.**, **Poignant, J.**, **Fortier, G.**, **Tapaswi, M.**, **Le, V.B.**, **Roy, A.**, **Barras, C.**, **Rosset, S.**, **Sarkar, A.**, **Yang, Q.**, **Gao, H.**, **Mignon, A.**, **Verbeek, J.**, **Besacier, L.**, **Quénot, G.**, **Ekenel, H.K. and Stiefelhagen, R.** (2013). QCompere @ REPERE 2013, *SLAM 2013, First Workshop on Speech, Language and Audio for Multimedia*, Marseille, France, pp.49–54.

[18] **Lam, L.**, (2000). Classifier combinations: implementations and theoretical issues, Multiple classifier systems, Springer, pp.77–86.

[19] **Cataltepe, Z. and Ekmekci, U.** (2013). Classifier Combination with Kernelized EigenClassifiers, *16th International Conference on Information Fusion*.

[20] **Kuncheva, L.I. and Whitaker, C.J.** (2001). Ten measures of diversity in classifier ensembles: limits for two classifiers, *Intelligent Sensor Processing (Ref. no. 2001/050), A DERA/IEE Workshop on*, IET, pp.10–1.

[21] **Eckart, C. and Young, G.** (1936). The approximation of one matrix by another of lower rank, *Psychometrika*, ***1***(3), 211–218.

[22] **Schölkopf, B.**, **Smola, A. and Müller, K.R.**, (1997). Kernel principal component analysis, Artificial Neural Networks ICANN'97, Springer, pp.583–588.

[23] **Krizhevsky, A.**, **Sutskever, I. and Hinton, G.** (2012). Imagenet classification with deep convolutional neural networks, *Advances in Neural Information Processing Systems 25*, pp.1106–1114.

[24] **Demšar, J.** (2006). Statistical comparisons of classifiers over multiple data sets, *The Journal of Machine Learning Research*, ***7***, 1–30.

[25] **Sutskever, I.**, **Martens, J.**, **Dahl, G. and Hinton, G.** On the importance of initialization and momentum in deep learning.

[26] **Giraudel, A.**, **Carré, M.**, **Mapelli, V.**, **Kahn, J.**, **Galibert, O. and Quintard, L.** (2012). The REPERE Corpus: a multimodal corpus for person recognition., *LREC*, pp.1102–1107.

[27] **Bredin, H.**, **Roy, A.**, **Le, V.B. and Barras, C.** (2014). Person Instance Graphs for Mono-, Cross- and Multi-Modal Person Recognition in Multimedia Data. Application to Speaker Identification in TV Broadcast, *International Journal of Multimedia Information Retrieval*.

APPENDICES

APPENDIX A : Detailed information on AYSU [1] datasets

APPENDIX A

Table A.1: Detailed information on AYSU [1] datasets

	mushroom	nursery	optdigits	pageblock	pendigits	pima	ringnorm
Train#	5415	8638	2545	3646	4994	511	4932
Test#	2709	4320	1278	1827	2500	257	2468
Feature#	22	8	65	10	17	8	21
Target#	2	4	10	5	10	2	2

	segment	spambase	tae	australian	balance	breast	bupa
Train#	1540	3066	99	459	416	465	229
Test#	770	1535	52	231	209	234	116
Feature#	20	58	6	14	5	9	6
Target#	7	2	3	2	3	2	2

	car	cmc	credit	cylinder	dermatologecoli		flags
Train#	1151	981	459	360	241	221	127
Test#	577	492	231	180	125	115	67
Feature#	7	9	15	35	34	7	26
Target#	4	3	2	2	6	8	8

	flare	glass	haberman	heart	hepatitis	horse	ionosphere
Train#	214	140	204	180	103	244	234
Test#	109	74	102	90	52	124	117
Feature#	10	9	3	13	19	26	34
Target#	3	7	2	2	2	2	2

	iris	monks	thyroid	tictactoe	titanic	twonorm	vote
Train#	99	288	1865	638	1467	4932	290
Test#	51	144	935	320	734	2468	145
Feature#	4	7	27	9	3	20	16
Target#	3	2	4	2	2	2	2

	wine	yeast	zoo
Train#	118	986	64
Test#	60	498	37
Feature#	13	8	16
Target#	3	10	7

CURRICULUM VITAE

Name Surname: Ümit EKMEKÇİ

Place and Date of Birth: Malatya 29.06.1984

Adress:

E-Mail: uekmekci@itu.edu.tr, umut.ekmekci@gmail.com

B.Sc.: Uludağ University

PUBLICATIONS/PRESENTATIONS ON THE THESIS

- **Ekmekci, U. and Cataltepe, Z. (2013).: Classifier Combination with Kernelized EigenClassifiers *16th International Conference on Information Fusion*, July 9-12, 2013 İstanbul, Turkey.**