MULTI-VIEW FEATURE EXTRACTION BASED ON CANONICAL CORRELATION ANALYSIS

by

Cemal Okan Şakar

B.S., Mathematical Engineering, Yildiz Technical University, 2006

M.S., Computer Engineering, Bahcesehir University, 2008

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

Graduate Program in Computer Engineering

Boğaziçi University

2014

MULTI-VIEW FEATURE EXTRACTION BASED ON CANONICAL CORRELATION ANALYSIS

APPROVED BY:

Prof. Fikret Gürgen (Thesis Supervisor)	.....
Assoc. Prof. Olcay Kurşun (Thesis Co-supervisor)	.....
Prof. H. Levent Akın	.....
Prof. Tunga Güngör	.....
Prof. Mehmed Özkan	.....
Assoc. Prof. Çiğdem Eroğlu Erdem	.....

DATE OF APPROVAL: 06.01.2014

ACKNOWLEDGEMENTS

I would first like to thank my supervisor Prof. Fikret Gürgen for his guidance and encouragement throughout this thesis study. His advices were very helpful and are reflected throughout this thesis. His understanding and support were motivational through the good and the bad times.

I would like to express my deepest appreciation to my co-supervisor Assoc. Prof. Olcay Kurşun. His door was always open for questions and discussions. I greatly value his knowledge and personal reflections shared with me through our discussions. This thesis would not be possible without him.

I want to thank Prof. Tunga Güngör and Prof. Mehmed Özkan for their useful comments throughout the preparation of this thesis. I would like to thank Prof. H. Levent Akın and Assoc. Prof. Çiğdem Eroğlu Erdem for their participation in my thesis jury and helpful comments. I would also like to thank Assoc. Prof. Oleg V. Favorov from University of North Carolina for his valuable suggestions.

I am very grateful to have been a Ph.D. student at the Department of Computer Engineering. In particular, I want to thank Orhan Ermiş, Heysem Kaya, Umut Konur, Can Kavaklıoğlu, and İsmail Arı for their discussions and friendship over the years.

I would like to thank my parents Mesude and Gökalp, my brother Gökhan, every member of my family, and my friends who have always supported and encouraged me all these years. Finally, I would like to give a special thank to my wife Betül. Without her love and support it would have been impossible for me to finish this work.

This thesis has been supported by Boğaziçi University Scientific Research Project 11A01D5, and the Ph.D. scholarship (2211) from the Scientific and Technological Research Council of Turkey (TÜBİTAK).

ABSTRACT

MULTI-VIEW FEATURE EXTRACTION BASED ON CANONICAL CORRELATION ANALYSIS

Canonical Correlation Analysis (CCA) aims at identifying linear dependencies between two sets of variables. CCA has recently become popular in the field of machine learning with the increase in the number of multi-view datasets, which consist of different representations coming from different sources or modalities. This thesis presents our efforts to improve the robustness and discriminative ability of CCA. CCA uses the views as complex labels to guide the search of maximally correlated projection vectors (covariates). Therefore, CCA can overfit the training data. Although, ensemble approaches have been effectively used to avoid such overfittings of classification and clustering techniques, an ensemble approach has not yet been formulated for CCA. In this thesis, we propose an ensemble method for obtaining a final set of covariates by combining multiple sets of covariates extracted from subsamples. Experimental results on various datasets demonstrate the usefulness of ensemble CCA approach. The correlated features extracted by CCA may not be class-discriminative since it does not utilize the class labels in its implementation. This thesis introduces a method to explore correlated and also discriminative features. Our proposed method utilizes two (alternating) multi-layer perceptrons, each with a linear hidden layer, learning to predict both the class-labels and the outputs of each other. The experimental results show that the features found by the proposed method accomplish significantly higher classification accuracies. Another contribution of this thesis is the use of CCA to improve a filter feature selection algorithm. We also present our works on ensemble classification and clustering for multi-view datasets.

ÖZET

KANONIK KORELASYON TABANLI ÇOK-BAKIŞLI ÖZNİTELİK ÇIKARIMI

Kanonik Korelasyon Analizi (KKA) iki değişken kümesi arasındaki doğrusal bağıntıları belirlemeyi amaçlayan bir yöntemdir. KKA son zamanlarda makine öğrenme alanında aynı verinin farklı temsillerinden oluşan çok-bakışlı veri kümelerinin artmasıyla birlikte çokça kullanılmaya başlamıştır. Bu tez, KKA yönteminin gürbüzlüğü ve sınıflandırma başarısının arttırılmasına yönelik çalışmaları içermektedir. KKA, maksimum korelasyona sahip izdüşüm vektörlerinin (eşdeğişkenler) bulunması için bakışları karmaşık sınıf etiketleri gibi kullanmaktadır. Bu nedenle, KKA eğitim kümesi üzerinde aşırı öğrenmeye sebep olabilir. Topluluk öğrenme yöntemleri sınıflandırma ve kümeleme yöntemlerinin bu tür aşırı öğrenme sorunlarını engellemek için kullanılmış, ancak KKA için bir topluluk yaklaşımı henüz önerilmemiştir. Bu tezde, birden fazla alt-örneklemden elde edilen eşdeğişken kümelerinin birleştirilmesiyle nihai bir eşdeğişken kümesinin elde edilmesi için bir topluluk yöntemi önerdik. Çeşitli veri kümeleri üzerinde elde edilen deneysel sonuçlar topluluk KKA yönteminin başarısını göstermektedir. KKA yönteminin gerçeklemesinde sınıf etiketlerinden yararlanılmadığı için, bu yöntemle elde edilen öznitelikler sınıf-ayırıcı özelliğe sahip olamayabilmektedir. Bu tez iki bakış arasındaki ortak bilgiyi içeren ve aynı zamanda farklı sınıflara ait örnekleri ayırt edebilen öznitelikler arayan bir yöntem önermektedir. Önerdiğimiz yöntem her biri doğrusal gizli katmanlı ve hem sınıf örneklerini hem de birbirlerinin çıktılarını tahmin etmeyi öğrenmeyi amaçlayan iki çok katmanlı algılayıcıdan oluşmaktadır. Deneysel sonuçlar, önerilen yöntemle çıkartılan özniteliklerin daha yüksek sınıflandırma başarısı verdiğini göstermiştir. Bu tez çalışmasının diğer bir katkısı, KKA yönteminin bir öznitelik seçme yönteminin geliştirilmesinde kullanılmasıdır. Bunun yanında çok-bakışlı veri kümeleri için topluluk sınıflandırma ve kümeleme üzerine çalışmalarımızı da içermektedir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

0_j	Column vector of j zeros
1_j	Column vector of j ones
l^i	Column vector whose ith element is 1 and the others are zero
B	Total number of independent subsamples
C	Covariance matrix
C_{xx}	Within-set covariance matrix
C_{xy}	Between-set covariance matrix
D	Block matrix of two datasets
$D(\cdot,\cdot)$	Dependency function
D_*^i	A subsample of block data matrix D
$E(X,Y)$	Expectation taken over the joint distribution of X and Y
$f_{j,u}(X_j)$	uth correlated function of jth feature
F_j	Set of correlated functions with target variable
I	Mutual information
$IR(\cdot,\cdot)$	Irrelevant redundancy function
K_b	Between-class correlation matrix
K_w	Within-class correlation matrix
L	Class label matrix
$\mathbf{M_x}$	Covariate correspondence matrix
N	Total number of instances
N_j	Number of samples in class j
p	Number of classes
$\mathbb{R}$	Real numbers
$R(\cdot,\cdot)$	Redundancy function
$RR(\cdot,\cdot)$	Relevant redundancy function
S_b	Between-class scatter matrix
S_t	Total scatter matrix
S_w	Within-class scatter matrix

t	Target variable
T	Transpose
w_x	Canonical vector of view X
W_x^i	Set of canonical vectors extracted using X_*^i
$\mathbf{W}_x$	Matrix containing all canonical vectors of subsamples of X
$\overline{x}$	Global mean of X
x_j^i	jth sample of class i
X_i^*	A subsample of dataset X
δ	Eigenvalue
$\eta_{\mathbf{0}}$	Inhibition coefficient of output layer of D-ARNet
η	Learning factor of MLP
λ	Discrimination factor of D-Ar method
ρ_u	Correlation coefficient between $f_{j,u}(X_j)$ and target variable

LIST OF ACRONYMS/ABBREVIATIONS

AR	Alternating Regression
CCA	Canonical Correlation Analysis
CCFS	Canonical Correlation Feature Selection
D-AR	Discriminative Alternating Regression
D-ARNet	Discriminative Alternating Regression Network
DCCA	Discriminative Canonical Correlation Analysis
ECCA	Ensemble Canonical Correlation Analysis
ECCA-B	Ensemble Canonical Correlation Analysis-Bagging
ECCA-J	Ensemble Canonical Correlation Analysis-Jackknife
ECCA-P	Ensemble Canonical Correlation Analysis-Partitioning
FMCD	Fast-Minimum Covariance Determinant
k-NN	k-Nearest Neighbor
KCCA	Kernel Canonical Correlation Analysis
KCCAmRMR	Kernel Canonical Correlation Analysis based minimum Redundancy-Maximum Relevance
LDA	Linear Discriminant Analysis
Local-DCCA	Local Discriminative Canonical Correlation Analysis
MCD	Minimum Covariance Determinant
MI	Mutual Information
MLP	Multi-layer Perceptron
mRMR	minimum Redundancy-Maximum Relevance
PCA	Principal Component Analysis
PD	Parkinson's Disease
PIML	Parallel Interacting Multi-view Learning
RCE	Random Correlation Ensemble
SCCA	Sparse Canonical Correlation Analysis
SINBAD	Set of INteracting BAckpropagating Dendrites
SVM	Support Vector Machine

1. INTRODUCTION

Recently, in parallel with the advances in hardware technology, data collection devices and sensors are becoming more portable and less costly, which enable the researchers in the field of machine learning to collect multiple types of data (multi-view data) about the same underlying phenomenon. The term "view" is used to refer each related set of features in the field of machine learning. In consideration of these technological advances, it can be said that gathering data is becoming easy. However, finding test subjects, especially in biomedical applications where the test subjects are mostly consisting of patients is still not easy. For example, Parkinson's disease (PD) is generally observed mostly in elderly people whose physical visits to the clinic are inconvenient and costly, so collecting sufficient number of samples from these patients is often not possible [1]. On the other hand, PD is a neurodegenerative disorder of central nervous system which causes disorders in both speech and handwriting motor abilities of patients. Hence, the number of samples required to construct a generalizable non-invasive PD diagnosis decision support system can be reduced by gathering both speech and handwriting samples of the patients together. Alternatively, one can constitute a multi-view data by applying different feature representation techniques to the raw single view data, e.g. extracting morphological features and zernike moments from the original handwriting samples [2], or using amino-acid and dipeptide composition of protein sequences [3].

Due to this recent popularity of multi-view datasets, the need for specific techniques to analyze the relationships between the views of multi-view data and reduce the dimensions of the multi-view data by extracting robust features is increasing. While traditional dimensionality reduction methods such as Principal Component Analysis (PCA) or Linear Discriminant Analysis (LDA) can be used to extract features from a single view data, when the data is represented with multiple sets of features, treating the data as a single view by concatenating the views from different probability distributions and applying single-view dimension reduction methods further decreases the number of samples to number of features ratio. Besides, these methods cannot

be used to detect correlations between the views of a multi-view data. Accordingly, although it has been available to researchers in theory for roughly 80 years [4], Canonical Correlation Analysis (CCA) [5] has recently become popular for discovering linear relationships between two views (multidimensional variables) in the field of machine learning.

Canonical Correlation Analysis (CCA) is a well-established statistical method which has been proposed by Hotelling [5] in 1936. Ignoring its various extensions to more than two views [6], CCA can be said to seek a pair of linear transformations, one for each of the two sets of variables, such that when they are projected onto these canonical vectors, the corresponding coordinate scores (covariates) are maximally correlated [7]. Thus, CCA is a feature extraction method that remains in between the unsupervised and supervised methods: similar to unsupervised methods it does not use the class labels; on the other hand, just like a supervised method it uses each view as a complex label of the underlying semantics to guide the other view [7]. In fact, CCA is equivalent to LDA when the data features are used in one view and the class labels are used as the other view [8]. Figure 1.1 shows how CCA can be used to extract features between views of a multi-view PD dataset. Two views of the PD dataset, speech signals and handwriting samples of the subjects, are guiding each other, and the extracted features $f(X)$ and $g(Y)$ are maximally correlated functions of the views which are expected to have useful information for PD diagnosis since both of speech and handwritten impairments are seen in patients with PD.

CCA has been used in a wide range of disciplines with different purposes. Preliminary studies that used CCA are mostly based on quantifying and analyzing the relations between different but related multidimensional variables [9–13]. For example, in one of these early studies, Varis [9] applied CCA to explore the associations between small species and physical and chemical growth factors in a natural environment. Many recent studies that address engineering problems have also utilized from CCA with the same purpose [14–20]. Afterwards, CCA has started to be used for classification and regression problems as a feature extractor in the field of machine learning based on the idea that common information included in both of the views obtained from the same

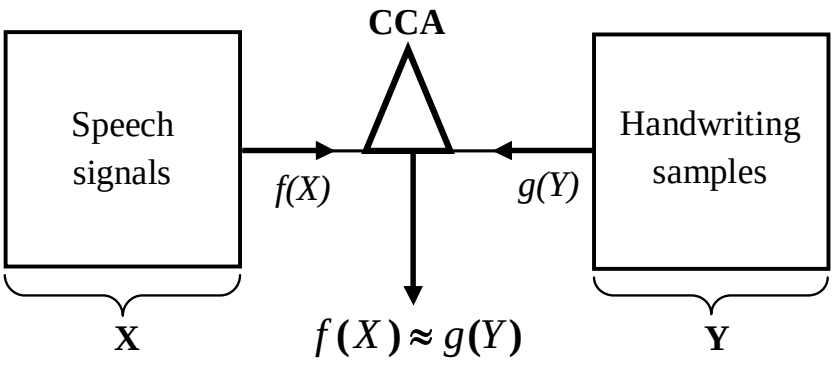


Figure 1.1. Feature extraction between views of a Parkinson's disease dataset using CCA.

data source comprises the important discriminant information [7, 21–23]. In parallel with the increasing number of applications that utilize CCA for different purposes, some studies have focused on improving the robustness and discriminative ability of the CCA covariates. In this thesis, we are concerned with both the robustness and discriminative issues of CCA. Besides, we utilize CCA to improve a recently popular filter feature selection method, and also present our preliminary studies on ensemble classification and clustering.

1.1. Motivation

The traditional formulation of CCA utilizes the within-set and between-set sample covariance matrices to explore the linear transformations that maximize the correlation between two sets of variables (views). However, these sample covariance matrices can be very sensitive to outliers and noisy samples [24–26], which can cause CCA to tune to dummy dependencies in the training set that do not hold in the test set. Accordingly, the literature studies that have focused on improving the robustness of CCA features replaced the sensitive sample covariance matrices [27–29] of the views with their robust estimations or totally avoided the use of sample covariance matrices [30–32]. A simpler typical regularization approach to improve the generalization ability of the features extracted by CCA is reducing the dimension of each view separately using PCA before the application of CCA [19, 21, 33, 34]. However, although ensemble approaches in

which multiple classifiers or clusterings are combined in attempt to produce a stronger model have been successfully applied for dealing with such generalization and overfitting problems in classification, regression, and clustering problems, an ensemble CCA has not yet been proposed to address the robustness issue of CCA features. In this thesis, our aim is to propose an ensemble CCA approach which combines many weak correlations obtained from resampled subsets of the views to produce a final set of stronger correlations with good generalization on the unseen test set examples. As the proposed ensemble approach is used to improve the generalization of traditional CCA features, it can also be applied to the robust implementations of CCA to further improve their generalization.

The existing studies that aim to improve the discriminative ability of CCA utilize the class information while fusing the views. The straightforward approach to incorporate the class labels into the feature fusion framework is to use an objective function similar to that of LDA. Such studies employed the class information by maximizing the correlation between feature vectors in the same class and minimizing the correlation between feature vectors belonging to different classes [35, 36]. However, it has already been shown that the extracted features with this approach are identical to those by LDA with respect to an orthogonal transformation [37]. Besides, such discriminative CCA methods, similar to traditional CCA and LDA, still use the sample covariance matrices, which are sensitive to outliers and noisy samples. In this thesis, we aim to propose a discriminative CCA method which can really take advantage of the class labels in CCA computation and also avoids the use of sensitive sample covariance matrices.

1.2. Contributions

In this section, we summarize the major contributions of this thesis study, and refer to the particular sections of the thesis where they can be found:

- *Ensemble Canonical Correlation Analysis (ECCA):* We adapt the ensemble idea to the CCA problem and propose an ensemble CCA (ECCA) method for obtaining

a final set of covariates by combining multiple sets of covariates extracted from subsamples (Section 3.3). We address the covariate correspondence problem and propose the use of covariate correspondence matrix to combine the covariates of various subsamples. Experimental results on various datasets have shown that ECCA has better generalization for both the test set correlations of the covariates and the test set accuracy of classification performed on these covariates (Section 5.1).

- *Discriminative Alternating Regression (D-AR):* We propose a method called D-AR that integrates the class labels into its feature extraction framework without the use of sensitive sample covariance matrices (Section 4.2). It is a linear dimensionality reduction method based on the alternating regression approach implemented by a multi-layer neural network with a "linear" hidden layer. We show that the D-AR features accomplish significantly higher classification accuracies on test sets of experimental datasets than its counterparts (Section 5.2).
- *Kernel Canonical Correlation Analysis based minimum Redundancy-Maximum Relevance (KCCAmRMR):* We show that CCA can be used to improve a popular filter feature selection algorithm called minimum Redundancy-Maximum Relevance (mRMR). We first show on a simple toy/synthetic problem that using mRMR can lead to inaccurate orderings of the variables because it does not deal with the type of the dependency, but only with its quantity (Section 2.4.2). Instead, we propose a method called KCCAmRMR which utilize kernel CCA to explore and use all the correlated functions (covariates) between variables to compute their unique (conditional) information about the target (Section 2.4.3). We demonstrate the usefulness of our method on both toy and benchmark datasets (Section 5.3).
- *Cluster Stacking and Parallel Interacting Multi-view Learning Methods:* As a part of our preliminary studies on multi-view datasets, in the context of this thesis, we propose a cluster ensembles method, called cluster stacking, and a two stage supervised multi-view learning technique, called Parallel Interacting Multi-view Learning (PIML). Cluster stacking approach is based on augmenting the clustering indices of multiple clusterings and using this augmented consensus

partition as the final partition (Section 3.2.3.3). The augmented cluster index matrix we used in the cluster stacking method formed a basis to propose the use of covariate correspondence matrix in the ensemble CCA method. We use our cluster stacking approach to reduce each view of a multi-view protein structure prediction dataset down to a single variable and compare its robustness with co-association based matrix cluster ensembles method (Section 5.4). Our proposed PIML method is based on the idea that the classical ensemble approach does not take conditional interdependences among the views (Section 3.2.2) into account. We demonstrate and compare the classification performance of PIML with that of the classical ensemble approach on protein and arrhythmia datasets (Section 5.5).

1.3. Organization of the Thesis and Publications

The contributions of this thesis have been published in [38–46]. The chapters of this thesis and the related publications are as follows:

In Chapter 2, we give the theoretical formulation of CCA and its relation to LDA. We provide an overview of the literature studies from various fields that use CCA for analyzing the relationships between different views of the a multi-view data, extracting features from a multi-view data, and selecting the most informative features of a given dataset. We also present our proposed feature selection method called KCCAmRMR, in which we utilized Kernel CCA to improve a recently popular minimum Redundancy-Maximum Relevance (mRMR) feature selection method. Parts of this chapter are published in [38, 40, 45].

In Chapter 3, we first summarize the existing robust CCA methods. Then, we give an overview of ensemble idea and its use for classification and clustering problems. We also present our proposed cluster stacking approach and supervised multi-view learning technique which are published in [39, 42, 43, 45]. Then, we review the existing ensemble approaches to CCA and mention their drawbacks. Finally, we define the covariate correspondence problem and present our ensemble canonical correlation

analysis (ECCA) method to deal with this problem. The ECCA method has been published in [41].

In Chapter 4, first we review the existing research, which aim to increase the discriminative ability of CCA features. Then, we propose an alternating regression based discriminative CCA method, called Discriminative Alternating Regression (D-AR), which aims to explore the discriminative covariates by incorporating the class labels into the view fusion framework. The D-AR method also avoids the use of the sensitive sample covariance matrices to extract the covariates. We give the architecture of the D-AR method which is based on the alternating regression approach implemented by a multi-layer neural network with a "linear" hidden layer. Some parts of this chapter are published in [44].

Chapter 5 presents the experimental results obtained with the proposed ECCA, D-AR, KCCAmRMR, cluster stacking, and PIML methods. We demonstrate the robustness of the covariates extracted by ECCA method on the emotion recognition, handwritten digit recognition, content-based retrieval, and object recognition experimental datasets. We also compare ECCA methods with CCA on a small toy dataset. Moreover, we compared the discriminative power of the covariates extracted by CCA and the proposed ECCA methods on the emotion recognition, handwritten digit recognition, and content-based retrieval experimental datasets. In our D-AR experiments, the discriminative power of the covariates extracted with the proposed D-AR method are evaluated and compared against of the traditional CCA, PCA+CCA, AR, and LDA methods on the emotion recognition and object recognition datasets under various training set sizes. We also provide experimental results to evaluate the discriminative ability of the features selected by the proposed KCCAmRMR method on six UCI datasets [47]. In our cluster stacking experiments, we present the comparative results on a protein dataset with multiple views that are used to predict protein structure. Finally, we provide the experimental results of the PIML method on two real protein datasets (secondary structure and subnuclear location prediction from sequence features) and one real dataset on arrhythmia type prediction. All the experimental results presented in Chapter 5 are published in [38, 39, 41–46].

We conclude and summarize the main contributions of this thesis in Chapter 6. We also discuss possible future work related to the proposed methods.

2. CANONICAL CORRELATION ANALYSIS

Canonical Correlation Analysis (CCA) aims at measuring linear relationships between two sets of variables (views). CCA is a well-established statistical method but with the recent developments in kernel and multi-view methods, it has become popular technique for discovering linear/nonlinear relationships between two views (multidimensional variables) [5, 7].

In this chapter, first, the theoretical formulation of CCA and the relation between CCA and LDA are given. Thereafter, the literature studies from various fields that use CCA for different aims are summarized. Finally, we present our proposed feature selection method called KCCAmRMR, in which we utilized Kernel CCA to improve a recently popular minimum Redundancy-Maximum Relevance (mRMR) [48] feature selection method.

2.1. Theoretical Formulation of CCA

Canonical Correlation Analysis (CCA) aims at discovering linear dependencies between two different but related views of the same underlying semantics [5]. It is used to detect the maximum correlation between linear combinations of the two views. The views guide each other behaving like complex labels as a way of feature selection towards the underlying semantics [7].

Suppose that we have a dataset D composed of two such views with N pairs of feature vectors:

$$D = d_i = \{(x_i, y_i), i = 1, 2, \cdots, N\},$$

where $x_i \in \mathbb{R}^m$, $y_i \in \mathbb{R}^n$, and N is the total number of instances in the whole sample.

We can consider the dataset D as a block matrix of centered datasets

$$X = [x_1, x_2, \cdots, x_N] \in \mathbb{R}^{m \times N},$$
$$Y = [y_1, y_2, \cdots, y_N] \in \mathbb{R}^{n \times N}.$$

In CCA, the objective is to project X and Y datasets onto basis vectors w_x and w_y, respectively, such that the correlation between the projections of the variables onto these basis vectors is mutually maximized. In other words, the aim is to maximize the correlation between the linear combinations $w_x^T X$ and $w_y^T Y$:

$$\begin{aligned} \rho &= \max_{w_x, w_y} corr(w_x^T X, w_y^T Y) \\ &= \max_{w_x, w_y} \frac{cov(w_x^T X, w_y^T Y)}{\sigma_{w_x^T X} \sigma_{w_y^T Y}} \end{aligned} \tag{2.1}$$

where a superscript T denotes transpose, *cov* denotes the covariance, and $\sigma_{w_x^T X}$ is the standard deviation of $w_x^T X$. The above correlation expression can be rewritten as

$$\begin{aligned} \rho &= \max_{w_x, w_y} \frac{\mathbb{E}[(w_x^T X)(w_y^T Y)^T]}{\sqrt{\mathbb{E}[(w_x^T X)(w_x^T X)^T]\mathbb{E}[(w_y^T Y)(w_y^T Y)^T]}} \\ &= \max_{w_x, w_y} \frac{\mathbb{E}[w_x^T X Y^T w_y]}{\sqrt{\mathbb{E}[w_x^T X X^T w_x]\mathbb{E}[w_y^T Y Y^T w_y]}} \\ &= \max_{w_x, w_y} \frac{w_x^T \mathbb{E}[X Y^T] w_y}{\sqrt{w_x^T \mathbb{E}[X X^T] w_x w_y^T \mathbb{E}[Y Y^T] w_y}} \end{aligned} \tag{2.2}$$

in which $\mathbb{E}$ denotes the expectation.

The covariance matrix of (X, Y) is

$$C(X, Y) = \mathbb{E} \begin{pmatrix} C_{XX} & C_{XY} \\ C_{YX} & C_{YY} \end{pmatrix} = C \tag{2.3}$$

where total covariance matrix C is a block matrix, $C_{XX} \in \mathbb{R}^{m \times m}$ and $C_{YY} \in \mathbb{R}^{n \times n}$ are within-set covariance matrices, $C_{XY} \in \mathbb{R}^{m \times n}$ and $C_{YX} \in \mathbb{R}^{n \times m}$ are between-set covari-

ance matrices, and $C_{XY} = C_{YX}^T$. Substituting these covariance matrices in Equation 2.2, we obtain

$$\rho = \max_{w_x, w_y} \frac{w_x^T C_{XY} w_y}{\sqrt{w_x^T C_{XX} w_x w_y^T C_{YY} w_y}}. \tag{2.4}$$

It is clearly seen that re-scaling of the canonical vectors w_x or w_y does not affect the solution of the maximization problem given in Equation 2.4, that is to say replacing w_x by αw_x gives the following equivalence:

$$\frac{\alpha w_x^T C_{XY} w_y}{\sqrt{\alpha^2 w_x^T C_{XX} w_x w_y^T C_{YY} w_y}} = \frac{w_x^T C_{XY} w_y}{\sqrt{w_x^T C_{XX} w_x w_y^T C_{YY} w_y}}. \tag{2.5}$$

Hence, the CCA optimization problem given in Equation 2.2 can be rewritten as:

$$\begin{aligned} \rho &= \max_{w_x, w_y} \frac{w_x^T C_{XY} w_y}{\sqrt{w_x^T C_{XX} w_x w_y^T C_{YY} w_y}} \\ \text{subject to } & w_x^T C_{XX} w_x = 1 \\ & w_y^T C_{YY} w_y = 1. \end{aligned} \tag{2.6}$$

Using the Lagrangian relaxation method to solve the above optimization problem, we obtain the following Lagrangian:

$$L(\lambda, w_x, w_y) = w_x^T C_{XY} w_y - \frac{\lambda_X}{2}(w_x^T C_{XX} w_x - 1) - \frac{\lambda_Y}{2}(w_y^T C_{YY} w_y - 1). \tag{2.7}$$

Finally, the CCA optimization problem given in Equation 2.6 is reduced to

$$\begin{aligned} XY^T (YY^T)^{-1} YX^T w_x &= \lambda^2 XX^T w_x \\ C_{XY} C_{YY}^{-1} C_{YX} w_x &= \lambda^2 C_{XX} w_x, \end{aligned} \tag{2.8}$$

and similarly for the canonical vectors of Y:

$$\begin{aligned} YX^T(XX^T)^{-1}XY^Tw_y &= \lambda^2 YY^Tw_y \\ C_{YX}C_{XX}^{-1}C_{XY}w_y &= \lambda^2 C_{YY}w_y \end{aligned} \tag{2.9}$$

which are eigenvalue problems of the form $Ax = \lambda Bx$. The canonical vectors, w_x and w_y, are obtained using the eigenvectors corresponding to the largest eigenvalues of $C_{XX}^{-1}C_{XY}C_{YY}^{-1}C_{YX}$ and $C_{YY}^{-1}C_{YX}C_{XX}^{-1}C_{XY}$, respectively. The projections of X and Y onto these canonical vectors, i.e. w_x^TX and w_y^TY, are called canonical variables (covariates).

After obtaining the first projective directions, (w_{x1}, w_{y1}), which satisfies Equation 2.6, next pair of projective directions corresponding to the second largest eigenvalues of $C_{XX}^{-1}C_{XY}C_{YY}^{-1}C_{YX}$ and $C_{YY}^{-1}C_{YX}C_{XX}^{-1}C_{XY}$ can be obtained by solving the following optimization problem:

$$\begin{aligned} \rho = \max_{w_x, w_y} & \frac{w_x^TC_{XY}w_y}{\sqrt{w_x^TC_{XX}w_xw_y^TC_{YY}w_y}} \\ \text{subject to } & w_x^TC_{XX}w_x = 1 \\ & w_y^TC_{YY}w_y = 1 \\ & w_{x1}^TC_{XX}w_x = 0 \\ & w_{y1}^TC_{YY}w_y = 0. \end{aligned} \tag{2.10}$$

The number of non-zero solutions to the eigenvalue problem of CCA is limited to the smaller dimensionality of views X and Y. Hence, the maximum number of covariates that can be extracted between X and Y is equal to the smaller of $m-1$ and $n-1$.

2.2. Relation Between CCA and LDA

Shortly after the proposal of CCA by Hotelling [5], in 1938, Bartlett [49] has revealed the connection between CCA and LDA by showing that LDA components

can be obtained with the application of CCA between the data matrix and the class label matrix L. Suppose that the matrix L represents the class label information of the data matrix and constituted using the $1-of-C$, or the more compact $1-of-(C-1)$ coding [8], defined as:

$$L = \begin{bmatrix} 1_{N_1} & 0_{N_1} & 0_{N_1} & \cdots & 0_{N_1} \\ 0_{N_2} & 1_{N_2} & 0_{N_2} & \cdots & 0_{N_2} \\ 0_{N_3} & 0_{N_3} & 1_{N_3} & \cdots & 0_{N_3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0_{N_p} & 0_{N_p} & 0_{N_p} & \cdots & 1_{N_p} \end{bmatrix}^T \in \mathbb{R}^{p \times N} \tag{2.11}$$

where p is the number of classes, N_j is the number of samples in class j, 1_j is a column vector of j ones, 0_j is a column vector of j zeros, and $N = \sum_{i=1}^{p} N_i$. The CCA problem here is to find a pair of linear transformations, one for input data X and one for class label matrix L, such that when they are projected onto these canonical vectors, the corresponding coordinate scores (covariates) are maximally correlated [7]:

$$\rho = \max_{w_x, w_l} corr(w_x^T X, w_l^T L) \tag{2.12}$$

Similar to Equation 2.6, the CCA problem between X and L can be written as:

$$\begin{aligned} \rho &= \max_{w_x, w_l} \frac{w_x^T C_{XL} w_l}{\sqrt{w_x^T C_{XX} w_x w_l^T C_{LL} w_l}} \\ \text{subject to } & w_x^T C_{XX} w_x = 1 \\ & w_l^T C_{LL} w_l = 1 \end{aligned}$$

which is transformed to the following eigenvalue problem as shown in Section 2.1:

$$XL^T(LL^T)^{-1}LX^T w_x = \lambda^2 XX^T w_x. \tag{2.13}$$

In the above problem, XL^T can be written as

$$\begin{aligned} XL^T &= \sum_{i=1}^{p}\sum_{j=1}^{N_i}(x_j^i - \overline{x})(l^i)^T \\ &= \sum_{i=1}^{p} N_i(\overline{x^i} - \overline{x})(l^i)^T \\ &= \widehat{X}I \\ &= \widehat{X} \end{aligned} \tag{2.14}$$

where $\overline{x}$ is the global mean, x_j^i denotes the jth sample of class i, (l^i) is a p dimensional column vector whose ith element is 1 and the others are zero, $\overline{x^i} = \frac{1}{N_i}\sum_{j=1}^{N_i} x_j^i$ is the class mean of X, and

$$\widehat{X} = [N_1(\overline{x^i} - \overline{x}), \cdots, N_p(\overline{x^p} - \overline{x})] \in \mathbb{R}^{m\times p}. \tag{2.15}$$

Now, we can rewrite the left-side of the eigenvalue problem given in Equation 2.13 as

$$XL^T(LL^T)^{-1}LX^Tw_x = \widehat{X}(LL^T)^{-1}\widehat{X}^T, \tag{2.16}$$

where

$$\begin{aligned} LL^T &= \sum_{i=1}^{p}\sum_{j=1}^{N_i} l^i(l^i)^T \\ &= \sum_{i=1}^{p} N_i l^i(l^i)^T \\ &= \begin{bmatrix} N_1 & & & \\ & N_2 & & \\ & & \ddots & \\ & & & N_p \end{bmatrix} \in \mathbb{R}^{p\times p}. \end{aligned} \tag{2.17}$$

Thus, replacing this matrix in Equation 2.16, we obtain

$$\begin{aligned} XL^T(LL^T)^{-1}LX^T w_x &= \widehat{X}(LL^T)^{-1}\widehat{X}^T \\ &= \widehat{X}\begin{bmatrix} N_1 & & & \\ & N_2 & & \\ & & \ddots & \\ & & & N_p \end{bmatrix}\widehat{X}^T \\ &= \sum_{i=1}^{p} N_i(\overline{x^i} - \overline{x})(\overline{x^i} - \overline{x})^T \\ &= S_b, \end{aligned} \tag{2.18}$$

where S_b is between-class scatter matrix:

$$S_b = \sum_{i=1}^{p} N_i(\overline{x^i} - \overline{x})(\overline{x^i} - \overline{x})^T. \tag{2.19}$$

Recall that LDA aims to project the data onto vectors such that the between-class scatter is maximized while the within-class scatter is minimized, so the instances that belong to different classes will be as well separated as possible, and the instances from the same class to be as close to their mean as possible [50]. Thus, in LDA, we want to obtain W which maximizes

$$\frac{|W^T S_b W|}{|W^T S_w W|} \tag{2.20}$$

where S_w is within-class scatter matrix, and is defined as

$$S_w = \sum_{i=1}^{p}\sum_{j=1}^{N_i}(x_j^i - \overline{x^i})(x_j^i - \overline{x^i})^T. \tag{2.21}$$

The LDA problem is reduced to the following eigenvalue problem:

$$S_b w = \lambda S_w w \tag{2.22}$$

and the LDA components are the eigenvectors corresponding to the largest $p-1$ eigenvalues of this problem. Now, we can rewrite the solution of the eigenvalue problem given in Equation 2.13 as

$$S_b w_x = \lambda S_t w_x \tag{2.23}$$

where $S_t = XX^T$ is the total scatter matrix, and $S_t = S_b + S_w$. Replacing this in Equation 2.23, we obtain

$$\begin{aligned} S_b w_x &= \lambda(S_b + S_w) w_x, \\ S_b w_x &= \frac{\lambda}{1-\lambda} S_w w_x \end{aligned} \tag{2.24}$$

which is equal to the problem of LDA given in Equation 2.22.

There are some research efforts that aim to obtain more useful projections than LDA using the equivalence between CCA and LDA [8]. These studies are based on using different encoding modes for class labels. In one of such study, Sun and Chen [51] proposed a method which uses soft labels instead of using a common label for all the samples of the same class in order to assign different values to the samples of a class that lie on different regions of the input space. Thus, they assigned more importance on the samples near the class boundaries when compared with the high-density regions with class centers. Their experimental results showed that the extracted features using soft label based CCA have higher classification performance than those of extracted with traditional LDA. Loog *et al.* [52] proposed a similar idea for an image segmentation task. They constituted the second view of CCA by incorporating the class labels of neighborhood pixels into their feature extraction framework with the aim of taking advantage of the spatial context. In a more recent study, Kursun *et al.* [8] applied

CCA to maximize the correlation between all pairs of data samples that belong to the same class. The proposed method which they called within class coupling CCA (WCCCA) performs a form of implicitly-supervised LDA and useful when the class labels are embedded in the spatial and/or temporal patterns of the data rather than being explicitly available. They demonstrated the usefulness of WCCCA on a face database.

2.3. Applications of Canonical Correlation Analysis

Canonical Correlation Analysis (CCA) is a statistical method which has been proposed by Hotelling [5] in 1936. However, although CCA has been available to researchers in theory for roughly 80 years [4], it has recently found use in the field of machine learning due to the rapid rise of multi-view datasets.

In multi-view setting, the dataset is composed of two or more related sets of features (views). The term "multi-view" is used to refer multiple sets of features about the same underlying phenomenon. The views can be obtained from totally different natural data sources that belong to the same phenomenon. For example, in some speech recognition problems, the researchers use both the audio and visual information to obtain more generalizable models [53]. Besides, a multi-view data can be obtained from a single view data by applying different feature representation techniques to the original input. For example, the handwriting samples of subjects may be represented with morphological features and Zernike moments [2]. The views belong to the same object or class label in supervised settings. Most of the existing studies apply CCA to extract/fuse features in multi-view classification/regression problems or analyze the relationships between related views of a phenomenon. Besides, there are also literature studies which utilize CCA for feature selection task.

2.3.1. Analysis of Relations

CCA has at first found use in a wide range of disciplines with the aim of exploring and analyzing the relations between different but related multidimensional variables. In

1991, Varis [9] used CCA to make an analysis of an environmental engineering problem. He studied the relations between the most prevalent phytoplankton species and nine physical and chemical growth factors in a polyhumic Finnish lake. In the same year, Bartos *et al.* [10] applied CCA to analyze another ecological problem. They applied CCA to study the potentialities of increasing wheat production by dividing the considered 30 variables into two sets. By the aid of eigenvalues Bartlett-test they determined that the two sets can be described by 7 canonical variables based on 30 variables. They also explored many canonical relations among the ecological factors (e.g. humus%, CaCO3, stickiness, NO2 + NO3, fertilizer N) and nutrient content variables (e.g. dry matter, P, K, Na, Ca, Mg%, crude protein, crude fibre). In 1992, Wade *et al.* [11] analyzed the relationship between neuroticism and extraversion on the 4 major stages of pain processing, that of pain sensation intensity, pain unpleasantness, suffering, and pain behavior, in 205 chronic pain patients (88 male and 117 female). They found that personality traits effect the ways in which people cognitively process the meanings that chronic pain holds for their life, and hence the extent to which they suffer. Finkenberg [12] conducted CCA to explore the relations between personal incentives for exercise and body esteem using the Personal Incentives for Exercise Questionnaire and the Body Esteem Scale of two hundred twelve women and 93 men enrolled in physical education courses. In another study, Cserhati and Forgacs [13] aimed to find the relationships between the retention characteristics and molecular structure using CCA and compared CCA with various multivariate mathematical-statistical methods such as PCA and cluster analysis. They showed that molecular substructures have significant influence on the reversed-phase retention behaviour of non-ionic surfactants. They also found significant linear correlations between the covariates and the first coordinate of the nonlinear map of principal component variables.

CCA is also used in more recent studies with the aim of exploring relationships between different multidimensional variables. In a study of crew interaction with the automatic flight control system of an aircraft, Degani *et al.* [14] observed 60 flights and recorded every change in the aircraft control modes. They also recorded every observable change in the operational environment. They used CCA to quantify the relationships between the state of the operating environment and pilots' actions and

responses. Instead of presenting CCA by means of numerical tables, they proposed to use a sun-ray-like diagram to present the multiple patterns that exist in the data by employing Alexander's theory of centers [15]. In a facial expression recognition study, Zheng *et al.* [16] manually located 34 landmark points from each facial image and then converted these geometric points into a labeled graph vector to represent the facial features. As the second view, for each training facial image, they generated a six-dimensional semantic expression vector by combining the semantic ratings which describes the basic expressions. They utilized kernel CCA to learn the correlation between the labeled graph vector and the semantic expression vector, and according to this correlation, they estimated the associated semantic expression vector of a given test image and then performed the expression classification according to this estimated semantic expression vector.

In 2008, Cao *et al.* [17] aimed to examine how microbial community composition and denitrification enzyme activities (DEA) at a California salt marsh vary with environmental conditions, nutrient availability, and levels of pollutants. They used CCA to directly assess the relationships between microbial community profiles and environmental variables such as elevation, total organic carbon and dissolved organic carbon. In yet another study, using CCA, Ozkan *et al.* [18] aimed to investigate to which extent yield components are related to canopy components in Gerek-79 cv. of bread wheat. In their study, the first view is composed of yield components which are obtained using the biological yield, grain yield, 1000-grain weight, fertile spikelet number and harvest index. They used the canopy components as the second view which consists of spike number, spike length and plant height. The results of their analyses showed that there is significant high correlation of 0.923 between yield and canopy components. In the same year, Yang *et al.* [19] utilized CCA to learn the mapping between 2D face image and 3D face data, and observed that the proposed 2D-3D face matching method decreased the computation complexity drastically compared to the conventional 3D-3D face matching while keeping relative high recognition rate. Considering the requirement of complex multivariate relationships identification in large scale genomic studies with multiple phenotypic or genotypic measures, Parkhomenko *et al.* [20] proposed Sparse Canonical Correlation Analysis (SCCA) method which examines the relation-

ships between two types of variables and provides sparse solutions that include only small subsets of variables of each type by maximizing the correlation between the subsets of variables of different types while performing variable selection. They illustrated the practical use of SCCA on a human gene expression data. In a more recent study, Huang and He [54] applied CCA to establish the coherent subspaces between the principal component analysis (PCA) based features of high-resolution and low-resolution face images.

2.3.2. Feature Extraction

The simplest approach to learning from the data with multiple views (i.e. multiple feature sets) is to concatenate all the features and use it as a single view data. However, single view approach suffers from two main shortcomings: First, it increases the chances of facing the curse of dimensionality, which is a problem encountered in regression and classification problems when using a large number of features [55]. It is known that reducing the number of features by eliminating the irrelevant and redundant ones lead to more accurate and generalizable classification and regression models and so decreases the complexity of the system. Second, it fails to model the individual views of the data sampled from different multivariate statistical distributions, thus achieves lower generalization of the classifiers [56]. It has been shown that treating the multi-view data as if it consists of a single view decreases the performance when compared with combining the multiple representations of the data with a more sophisticated approach [57, 58]. Such an approach to supervised learning on multi-view data is ensemble learning which is based on employing separate classifiers on each view and combining the predictions of the views using techniques such as voting and stacked generalization (stacking) [59]. Besides, co-training [57] and co-EM [60] multi-view approaches are used to assign a label to the unlabeled examples in semi-supervised learning problems.

The ensemble approach which is based on learning from each view individually and then combining the predictions does not take the correlation information between the views into account [61]. Therefore, CCA has recently found use in the field of

machine learning as a feature extraction method since it can explore the correlated information between different views of multi-view data. In 2003, considering that CCA is a very powerful and versatile statistical tool that is especially well suited for relating two sets of measurements, Melzera *et al.* [23] showed that kernel CCA (KCCA) is an efficient non-linear feature extractor. They applied KCCA to build appearance object models for pose estimation. In the same year, Hardoon *et al.* [7] proposed an overview of CCA with application to learning methods. They attempted to learn the semantic representation of images and their associated text to enable content based retrieval with no reference to labeling. They used the extracted features which represents the derived semantic space in an image retrieval task.

Extracting features with CCA between different representations of the data and feeding these features to classifiers has also gained much attention in the field of pattern recognition. The primordial study that used CCA features for classification and regression goes back to Sun *et al.* [21]. They stated that since different views obtained from the same data source reflect the different characteristic of patterns, fusing these views to extract features not only preserves the important discriminant information, but also eliminates the redundant information to certain degree. On the handwritten Arabic numerals and Yale face databases, they showed that recognition rate of CCA features with minimum-distance classifier is significantly higher than that of using each view individually or the existing feature fusion algorithms. In another related study, Liu *et al.* [62] reduced the dimension of the original face image sample using PCA to overcome the singularity problem of the covariance matrix, and then applied CCA to extract the linear optimal discriminant features without losing Fisher discriminatory information. They fed the CCA features to k-nearest neighbor classifier and observed that PCA+CCA performed higher classification accuracy than Fisherface.

In face recognition problems, the number of samples is always smaller than the dimension of the dataset. In order to solve this small size problem encountered in CCA-based face recognition problems, Sun *et al.* [22] proposed a new supervised learning method called two-dimensional CCA. While traditional CCA extracts the features after transforming the matrix face data to vector representation, 2DCCA directly extracts

the features from image matrix. They observed that 2DCCA method achieves better recognition performance than several other CCA-based methods on face recognition datasets.

There are some research efforts in the literature that aimed at improving the discriminative ability of CCA features by incorporating the class labels into feature fusion and extraction framework. In 2007, Kim *et al.* [35] proposed a discriminative CCA method based on Linear Discriminant Analysis that maximizes the canonical correlations of within-class sets and minimizes the canonical correlations of between-class sets. Shortly after this work, Sun *et al.* [36] proposed a very similar method which also incorporated the class information into the framework of CCA for combined feature extraction. By feeding the extracted features to k-Nearest Neighbor (k-NN) and Naive Bayes classifiers, they compared the discriminative ability of DCCA with CCA and Partial Least Squares (PLS) on text categorization, face recognition and handwritten digit recognition datasets, and showed that DCCA features achieved higher classification accuracy than those of the alternative methods. Based on DCCA method, Peng *et al.* [63] proposed a CCA model with local discrimination (called Local-DCCA), which considers the local correlations of the within-class sets and the between-class sets. However, in 2011, Shin and Park [37] analyzed the correlation based dimension reduction methods and showed that the projective directions by DCCA are equal to the ones obtained from LDA with respect to an orthogonal transformation. They also proposed a method using within-class nearest neighborhood scatter which is especially effective for data with non-normal class distributions.

For the semi-supervised multi-view case where the class labels of some patterns are missing but the underlying phenomenon is represented with at least two views, Kursun and Alpaydin [64] proposed a CCA based semi-supervised feature extraction method called Semi-supervised CCA (SCCA). For the samples with missing class label, the SCCA method keeps the pattern of the other view. Otherwise, it represents the class-label by the class-center of the samples in that other view. CCA is used to extract features between thus generated views.

2.3.3. Feature Selection

As CCA has ability to learn a semantic space between two multi-dimensional variables, it can be used to represent one view in terms of the other one. As explained in Section 2.1, the objective of CCA is to project views X and Y onto basis vectors w_x and w_y, respectively, such that the correlations between the projections of the variables onto these basis vectors are mutually maximized. So, the features of view X can be sorted according to their importance in the explanation of view Y by the magnitudes of their corresponding weights in vector w_x. Based on this idea, Hardoon *et al.* [65] used CCA and its nonlinear extension Kernel CCA (KCCA) between fMRI and the activity signal to identify the pixels with high weights as pixels that are associated with high correlation will have a high weight value.

In 2008, Paskaleva *et al.* [66] presented a problem-specific CCA based feature selection algorithm, called Canonical Correlation Feature Selection (CCFS), for the general class of sensors whose bands are both noisy and spectrally overlapping. CCFS combines a generalized canonical correlation analysis framework and a minimum mean-square-error criterion for the selection of feature subspaces. They applied CCFS to classify rock species using laboratory spectral data and a quantum-dot infrared photodetector sensor, and to classify and estimate abundance of hyperspectral imagery obtained from the Airborne Hyperspectral Imager sensor. Their classification results on both applications showed that in the presence of noise, the proposed CCFS algorithm can effectively reduce the sensorspace dimensionality while maintaining good separability and classification results. In a more recent bioinformatics study, since Sparse CCA method [20] does not directly provide a ranking of features, Zheng *et al.* [67] proposed a simplified version of sparse CCA which produces a ranking of the features. Their study addressed the problem of identifying correlations between genes or features of two related biological system in which not all the features of predictor view is related with the other multi-dimensional view. The aim of their proposed CCA based feature selection model is ranking and selecting a subset of the features of the input view which are dependent on the other view and eliminating the remaining features which constitute a noise set. They demonstrated the effectiveness of the method

on a recent infant intestinal gene expression and metagenomics dataset.

2.4. Proposed KCCAmRMR Feature Selection Method

Feature selection methods allow obtaining more robust and accurate machine learning algorithms by reducing the high dimensional problem to a minimum set with maximum joint dependency [48, 68]. In this section, we propose an individual feature selection method called KCCAmRMR [38], in which we utilized kernel CCA (KCCA) to improve a recently popular minimum Redundancy-Maximum Relevance (mRMR) feature selection method [48]. Firstly, we briefly review the mRMR method. Then we give the problem with the mRMR approach on a demonstrative toy example to show that mRMR algorithm may cause inaccurate orderings. Finally, we present the details of our proposed KCCAmRMR method. In simple terms, KCCAmRMR explores the correlated functions between the variables and uses them to compute the redundancy term instead of directly using themselves (i.e. their exact values).

2.4.1. The minimum Redundancy-Maximum Relevance (mRMR) Method

The minimum Redundancy-Maximum Relevance (mRMR) method [48] is based on recognizing that the combinations of individually good variables do not necessarily lead to good classification/prediction performance. In other words, to maximize the joint dependency of top ranking variables on the target variable, the redundancy among them must be reduced, which suggests incrementally selecting the maximally relevant variables while avoiding the redundant ones. Based on this idea, mRMR uses mutual information as a filter in order to obtain maximum classification/prediction performance with a minimal subset of variables by reducing the redundancies among the selected variables to a minimum.

In mRMR algorithm, firstly, the mutual information (MI) between the candidate variable and the target variable is calculated (the relevance term). MI is a measure of the mutual dependence of two random variables. The definition of mutual information is based on Shannon's entropy [69]. The entropy of a random variable X, denoted

$H(X)$, is a function of the probability distribution function $P(X)$, and is sometimes written as $H(P(X))$ since the entropy of X does not depend on the actual values of X, it only depends on $P(X)$. Shannon's entropy is a measure of the uncertainty of a random variable X and thus, it quantifies how difficult it is to predict that variable. The definition of Shannon's entropy can be written as an expectation:

$$H(X) = -\mathbb{E}[\log P(X)] = -\sum_{x}[p(x)\log(p(x))], \tag{2.25}$$

where $p(x) = P(X = x)$ is the probability distribution function (more it is the precisely probability mass function for the discrete case but the results are generalizable) of X. Hence the Shannon's entropy is the average amount of information contained in random variable X. In other words, it is the uncertainty removed after the actual outcome of X is revealed. MI is a measure of mutual dependence of the two variables based on the entropy:

$$I(X;Y) = H(X) + H(Y)–H(X,Y), \tag{2.26}$$

where $I(X;Y)$ denotes the MI between random variables X and Y. The measure I is also the Kullback-Leibler divergence of the product $P(X)P(Y)$ of the two marginal probability distributions from the joint probability distribution, $P(X,Y)$. The MI between random variables X and Y can be defined in terms of their probabilistic density functions as:

$$I(X;Y) = \sum_{x}\sum_{y}\left[p(x,y)\log\left(\frac{p(x,y)}{p(x)\cdot p(y)}\right)\right], \tag{2.27}$$

where $p(x,y) = P(X = x, Y = y)$. Intuitively, mutual information is the shared information between X and Y. In other words, it measures how much uncertainty is removed from Y by knowing the state of X. Hence, if X and Y are independent, then knowing the state of X does not give any additional information about the state of Y and vice versa, so their mutual information is zero. On the other hand, if the two random variables are identical, knowing one of the variables completely removes the

uncertainty of the other one, so in this case the mutual information is equal to the entropy of one of these variables.

In mRMR algorithm, after the computation of the MI between the candidate variable and target, the average MI between the candidate variable and the variables that are already selected is computed (the redundancy term). The entropy-based mRMR score (the higher it is for a feature, the more that feature is needed) is obtained by subtracting the redundancy from relevance. This scheme is shown in Figure 2.1.

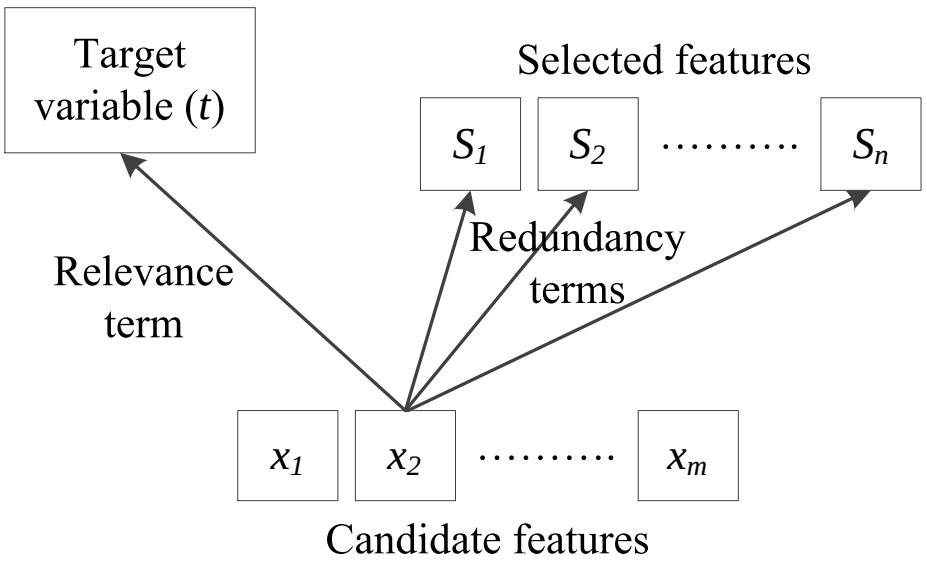


Figure 2.1. Computation of mRMR score of a candidate variable.

The maximal statistical dependency of the target variable (or class) t on the data distribution in the subspace $\mathbb{R}^m$ (and vice versa) is called maximal dependency (Max-Dependency) where m is the number of features chosen. In other words, Max-Dependency scheme is finding a feature set S with m features which jointly have the largest dependency on the target variable t [70]. This has the following form:

$$\max D(S,t), D = I(\{x_i, i = 1, 2, \ldots, m\}; t), \tag{2.28}$$

where x_i is the ith feature, and $I(x_i; t)$ denotes the mutual information between x_i and t. This form corresponds to adding one feature at each iteration which contributes to

the largest increase of $I(S;t)$. However, as Peng *et al.* [48] also stated, it is hard to get an accurate estimation for multivariate density $p(x_1,\ldots,x_m)$ and $p(x_1,\ldots,x_m,t)$ because the multivariate density estimation often involves computing the inverse of the high-dimensional covariance matrix, which is usually an ill-posed problem. Besides, the number of samples is often insufficient to accurately estimate the multivariate densities, and Max-dependency has slow computational speed. Therefore, Peng *et al.* [48] proposed an alternative way of selecting features based on maximal relevance criterion (Max-relevance). Max-relevance aims to maximize the mean value of all mutual information scores between individual feature x_i and target class t as an approximation of Equation 2.28:

$$\max D(S,t), D = \frac{1}{|S|}\sum_{x_i \in S} I(x_i;t). \tag{2.29}$$

This Max-relevance criterion approximates the maximum joint dependency of features on the target class by maximizing the summation of mutual information between individual features and target class. The features chosen with Max-relevance scheme are expected to have rich redundancy because we choose them according to their individual dependency with the target. Therefore, Peng *et al.* [48] proposed the following minimal redundancy (Min-redundancy) condition to select mutually exclusive features:

$$\min R(S), R = \frac{1}{|S|^2}\sum_{x_i, x_j \in S} I(x_i;x_j). \tag{2.30}$$

Finally, an operator for combining Equation 2.29 and Equation 2.30 is defined by:

$$\max \Phi(D,R), \Phi = D - R. \tag{2.31}$$

Thus, according to mRMR approach, mth feature chosen for inclusion in the set of selected variables, S_m, must satisfy the below condition:

$$S_m = \underset{x_j \in X - S_{m-1}}{\arg\max} \left[I(x_j;t) - \frac{1}{m-1}\sum_{x_i \in S_{m-1}} I(x_j;x_i) \right], \tag{2.32}$$

where X is the whole set of features. To better understand how this difference makes good sense, Equation 2.32 can be expressed in proportional to the entropy of x_j as shown below in Equation 2.33:

$$S_m = \underset{x_j \in X - S_{m-1}}{\arg\max} \left[H(x_j) \left(\frac{I(x_j;t)}{H(x_j)} - \frac{1}{(m-1)H(x_j)} \sum_{x_i \in S_{m-1}} I(x_j;x_i) \right) \right], \tag{2.33}$$

where $H(x_j)$ is the entropy of x_j. In the above equation, first term, $\frac{I(x_j;t)}{H(x_j)}$ denotes how much entropy of the candidate variable x_j in percent is common with the target t. The second term $\frac{1}{(m-1)H(x_j)} \sum_{x_i \in S_{m-1}} I(x_j;x_i)$ measures how much entropy of the candidate variable x_j in percent (average) is common with the selected variables (e.g. a variable x_j might have 60% of its entropy common with the target variable t and 40% of its entropy common with the other variables, in average, which seemingly points out that 20% if its entropy could be unique information about t that would be gained when x_j is included). Multiplying the difference of these terms with the variable's entropy gives the unique information that the variable has about the target class. Among the candidate variables, the variable with the maximum mRMR score is chosen next into the selected set of variables.

Although it has been showed that mRMR algorithm works well for some experimental studies, it is known that it causes inaccurate orderings in some cases since it only measures the quantity of redundancy between the candidate variables and the selected variables but does not deal with the type of this redundancy [40, 71, 72]. In this context, Peng *et al.* [48] applied the backward elimination wrapper technique after feature selection step by mRMR to get rid of these ineffective variables. Specifically, mRMR chooses some irrelevant variables too early and some useful variables too late. This is due to the fact that candidate variable that seems highly redundant with the already selected variables might carry unique information about the target variable. To address this problem, Sotoca and Pla [72] uses conditional mutual information to estimate relevant redundancies and cluster variables based on these redundancies; thus, each variable-cluster becomes a feature subset selected by an mRMR-like criterion. In another study, Gurban and Thiran [73] incrementally selected the features also us-

ing their conditional mutual information but taking difference of mutual information (with target); but conditional mutual information is not a good match as they note in their paper by requiring an additional parameter β as in [68] to bring them to the same scale. In our approach, all the entropies are computed by Shannon's mutual information (i.e. no conditionals) and the relevance and redundancy terms, thus, are comparable without need for further adjustments.

2.4.2. The Problem with mRMR Method

The problem with mRMR approach is that the subtracted redundancy term punishes features that are related to each other but carry different (i.e. unique or conditionally independent) information about the target variable. To illustrate, in Figure 2.2, two random variables, X_1 and X_2, a target variable t, and relations among them are visualized. Each region shown with r_i represents one unit of information. Suppose that X_1 has been already selected into the feature set S. In the latter iterations, mRMR score of each variable is computed to estimate the unique information they have about target t. For example, mRMR score computation of variable X_2 is shown below:

$$mRMR(X_2) = I(X_2; t) - I(X_2; X_1). \tag{2.34}$$

The mutual information between X_2 and t, $I(X_2, t)$, is the sum of r_3 and r_4 which are the intersection regions of X_2 and t. For the computation of the unique information that candidate variable X_2 carries about target t, the mutual information between X_2 and the set of selected variables must be computed and subtracted from $I(X_2, t)$, called the redundancy term. In our example, the redundancy between X_1 and X_2 is the intersection regions of these variables, the sum of r_3 and r_7. Therefore, according to Equation 2.34, mRMR score of X_2 can be computed as:

$$mRMR(X_2) = (r_3 + r_4) – (r_3 + r_7) = r_4 - r_7. \tag{2.35}$$

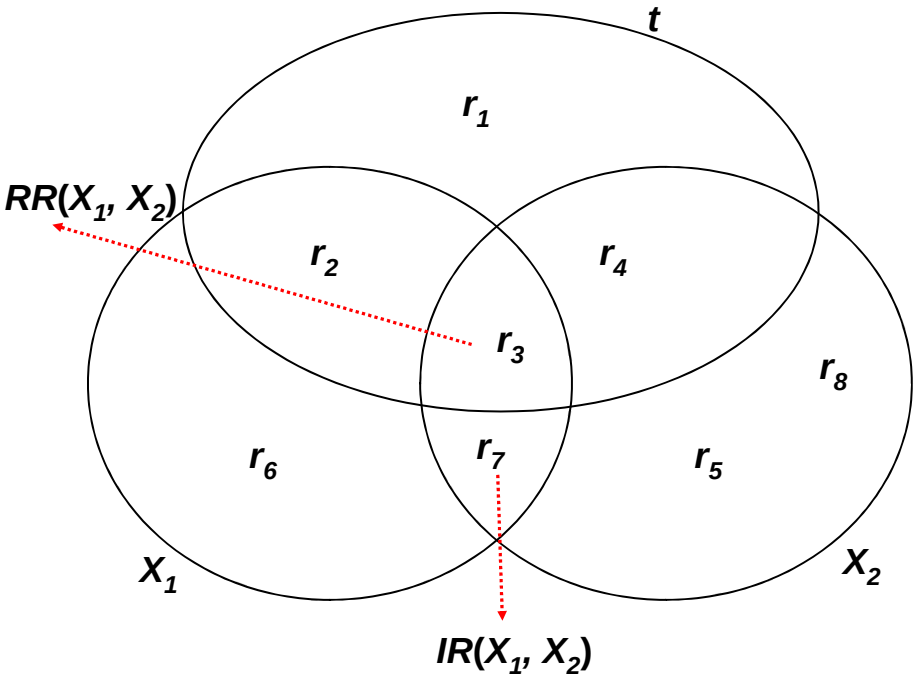


Figure 2.2. Venn diagram visualization of the relations between the variables for the definition of the problem.

However, as it is seen in Figure 2.2, the unique information that X_2 carries, i.e. distinct from the already selected variable X_1, about t is region r_4. Therefore, mRMR score of X_2 must be r_4. This is because of computing the redundancy term as the mutual information between X_1 and X_2. However, redundancy term must not be the quantity of common information that they carry about each other. It must be the common information that they carry about t. This information can be called as relevant redundancy (RR).

As seen in Figure 2.2, only region r_3 is the common information that X_1 and X_2 carry about t. Therefore, redundancy term must be only r_3, not $r_3 + r_7$. We see that r_7 is also common between X_1 and X_2, but it is not common with target t. In this context, we refer to this type of redundancy as irrelevant redundancy (IR). The redundancy term of mRMR algorithm may be expressed as the sum of RR and IR as:

$$Redundancy(X_1, X_2) = RR(X_1, X_2) + IR(X_1, X_2). \tag{2.36}$$

However, as it is mentioned above, redundancy term must only include $RR(X_1, X_2)$, which gives the quantity of common information that X_1 and X_2 carry about target t (i.e. relevant to the learning task).

As another demonstrative example showing the problem with mRMR, suppose that we have 4 variables, X_1, X_2, X_3, X_4, and our task is to predict the target variable t where

$$\begin{aligned} t &= r_2 + r_3 + r_4 + r_7 + r_8, \\ X_1 &= r_4 + r_5 + r_8 + r_9, \\ X_2 &= X_1 + r_6 = r_4 + r_5 + r_8 + r_9 + r_6, \\ X_3 &= r_7 + r_8 + r_9 + r_{10}, \\ X_4 &= r_1 + r_2. \end{aligned} \tag{2.37}$$

In Figure 2.3, the relations among the five random variables, X_1, X_2, X_3, X_4, and the target variable t, are visualized. Table 2.1 shows the mutual information scores among the variables. Each variable has different quantity of entropy. All r_i are integers distributed uniformly but their ranges differ to vary the mutual information scores. The example is created in such a way that X_1 is the most relevant variable to t and selected first by mRMR. Then, X_3 must be chosen as the next variable but since X_3 has high irrelevant redundancy (IR) with X_1, X_4 will be selected by mRMR and X_3 will be added one iteration later than it must have been.

Table 2.1. Mutual information scores among the variables.

	$\mathbf{X}_1$	$\mathbf{X}_2$	$\mathbf{X}_3$	$\mathbf{X}_4$
$\mathbf{X}_1$	3.1172	2.1728	0.2043	0.0219
$\mathbf{X}_2$	2.1728	3.1649	0.2434	0.0155
$\mathbf{X}_3$	0.2043	0.2434	2.8840	0.0125
$\mathbf{X}_4$	0.0219	0.0155	0.0125	1.4720
t	0.4826	0.4536	0.2218	0.0399

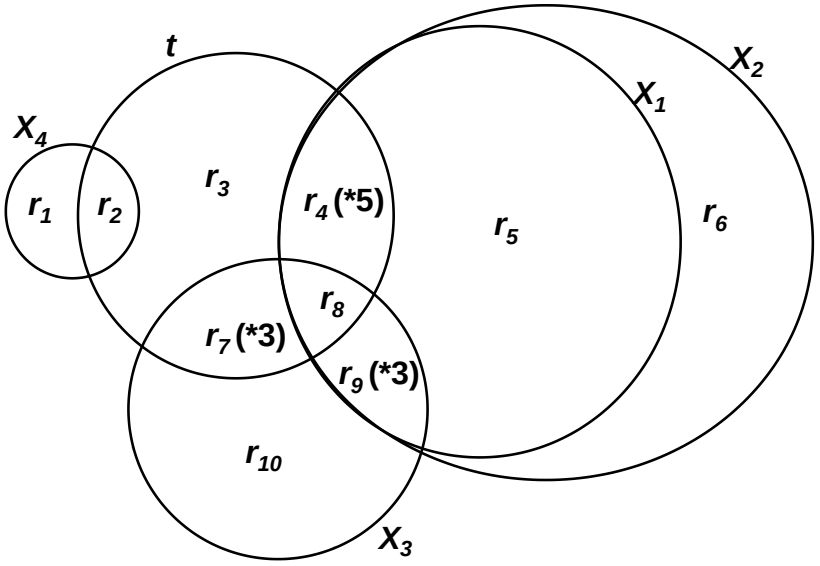


Figure 2.3. Venn diagram visualization of relations between variables of toy dataset.

In particular, X_1 consists of regions r_4, r_5, r_8, and r_9. Here, r_5 and r_8 are binary (0 or 1) and the variable r_4 ranges from 0 to 5. This bigger range of r_4 increases the mutual information between X_1 and target t and makes X_1 the most relevant variable to t. The variable r_9 ranges from 0 to 3 and increases the irrelevant redundancy between X_1 and X_3. X_3 consists of regions r_7, r_8, r_9, and r_{10} of which r_7 and r_8 contributes to the variability of t. r_7 takes on values between 0 and 3 so that X_3 is the second most important variable for predicting t. X_2 contains X_1 and additionally covers r_6 which takes values 0 or 1. Due to this additional uncertainty, which is not common with the target t, mRMR algorithm will prefer X_1 to X_2 initially. mRMR approach successfully leaves picking X_2 to the last step. X_4 consists of binary variables r_1 and r_2. r_1 is the random variable that contributes to the mutual information between X_4 and t.

mRMR algorithm chooses the variable that has the highest mutual information score with the target variable in the first iteration since there are no selected variables yet. X_1 and X_2 are designated to provide the most information to target t with r_4 and r_8 as described above. However, X_2 includes the region r_6 in addition to X_1 which decreases its mutual information with t because r_6 contributes to the variability of X_2 and is not common with t. Therefore, X_1 is chosen in the first order by mRMR. The problem with mRMR appears in the second iteration of this demonstrative example as

follows. X_2 has the highest mutual information with t among X_2, X_3, X_4 but it is very redundant with the already selected variable X_1, so the relevance term (the mutual information between variable and target) is mostly cancelled out by the redundancy term (average mutual information between variable and the set of selected variables). Thus, mRMR score of X_2 is nearly zero. X_3 has four units of common information with target t (r_7 is between 0 and 3 and r_8 is between 0 and 1 summing up to four) and four units of common information with selected variable X_1 (r_8 is between 0 and 1 and r_9 is between 0 and 2). Therefore, mRMR score of X_3 becomes practically zero. However, the unique information r_7 that X_3 supplies for target t should not have been cancelled out by X_1's r_9. In other words, the common information, r_7, with the target t that X_3 includes, which X_1 does not possess, is overlooked. X_4 has one unit of common information, r_2, with t and no common information with the already selected variable X_1. Thus, X_4 is chosen second for the set of selected variables as a mistake. Table 2.2 shows a representative table of mRMR scores of the variables and the selected variables in each iteration (even though the mRMR score of X_4 is just a bit larger than X_3, our aim here was to come up with a most simplistic example that this can happen, which can be thought as a kind of imperfection/noise in the algorithm).

Table 2.2. mRMR scores of variables in each iteration.

	$\mathbf{X}_1$	$\mathbf{X}_2$	$\mathbf{X}_3$	$\mathbf{X}_4$	**Selected Variable**
Iteration 1	0.4826	0.4536	0.2218	0.0399	$\mathbf{X}_1$
Iteration 2	-	-1.7192	0.0175	0.0180	$\mathbf{X}_4$
Iteration 3	-	-0.6406	0.1134	-	$\mathbf{X}_3$
Iteration 4	-	-0.3570	-	-	$\mathbf{X}_2$

2.4.3. KCCAmRMR Method

The KCCAmRMR method has been built on our earlier work [40] in which we used an unsupervised machine learning tool, SINBAD (Set of INteracting BAckpropagating Dendrites) [74, 75], while computing the relevant redundancy between the features and the target. In our KCCAmRMR approach, we use kernel canonical correla-

tion analysis (KCCA) [76, 77] in order to find the nonlinear relations, i.e. correlated functions, between the features and the target t. We use the explored functions with KCCA instead of the features themselves as in mRMR, and thus, filter out the irrelevant redundancies and take only the relevant redundancies into account while computing the redundancy term.

2.4.3.1. Kernel Canonical Correlation Analysis. Kernel Canonical Correlation Analysis (KCCA) is a nonlinear correlation measure for determining statistical dependencies between two random variables. KCCA is a kernelized version of CCA that leads to a generalized eigenvector problem in a reproducing kernel Hilbert space by making the use of a kernel for catching nonlinear relations that correspond to influential hidden factors responsible for the correlations [7,74,78]. Figure 2.4 shows that KCCA lifts the data into a higher dimensional feature space and reduces to a CCA problem that can be efficiently carried out in the input space anyway, known as the "kernel trick" [79,80].

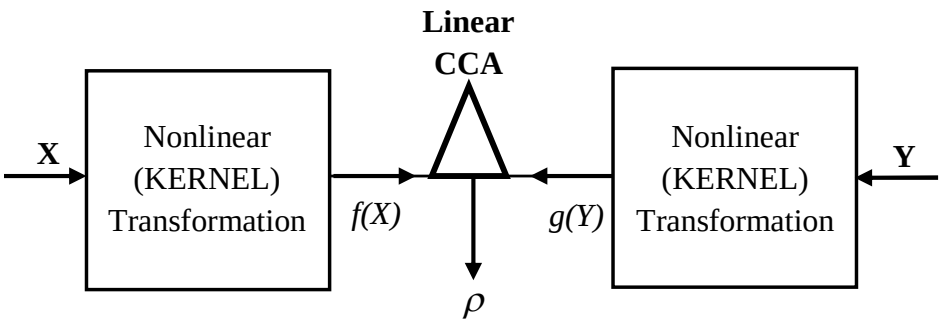


Figure 2.4. KCCA is reduced to a CCA problem with the use of kernel trick.

2.4.3.2. KCCAmRMR Formulation. The feature selection criterion of our KCCAmRMR method has two terms as in mRMR: relevance and redundancy. The difference is that in KCCAmRMR the correlated functions explored with KCCA are used instead of the features directly while computing the mutual information scores. Each function $f_{i,u}(x_i)$ corresponds to a relation of x_i with t (or a function of t). Thus, only the relevant information of x_i with the target is taken into account whereas the irrelevant

information of x_i about t is filtered out. The relevance term of KCCAmRMR's feature selection criterion is given below:

$$D_{x_j,t} = \sum_{u \in F_j} \left[I(f_{j,u}(X_j); t) \cdot \rho_u^2 \right], \tag{2.38}$$

where $D_{x_j,t}$ is the relevance between feature x_j and target t, F_j is the set of correlated functions with target t explored by KCCA, $f_{j,u}(x_j)$ is the uth correlated function between x_j and t, and finally ρ_u is the correlation coefficient between $f_{j,u}(x_j)$ and target t. In Equation 2.38, since there are multiple correlated functions, each of which represents a different relation, the relevance term is summed over all these correlated functions (the number of these correlated functions can be different for different features with varying correlation coefficients). We multiply the mutual information between each of the correlated function and target with the square of correlation coefficient, ρ_u^2, which is used to weigh the meaningfulness of the relation by taking into account the variance explained by the covariate.

As for the computation of the redundancy term, while computing a candidate feature's redundancy with already selected features, we use the correlated functions of the features (with the target) instead of using the features themselves. We sum redundancies over all pairs of correlated functions:

$$R_{x_j,x_i} = \frac{1}{m-1} \sum_{x_i \in S_{m-1}} \sum_{u \in F_j, v \in F_i} \left[I(f_{j,u}(x_j); f_{i,v}(x_i)) \cdot \rho_u^2 \cdot \rho_v^2 \right] \tag{2.39}$$

where R_{x_j,x_i} represents the target-relevant redundancy between features x_j and x_i. As in mRMR, in order to find the average redundancy of the candidate feature with the selected set of features, we divide the sum of mutual information among correlated functions to the number of already selected features. We also multiply the mutual information with the correlation coefficients, $\rho_u^2 \cdot \rho_v^2$, as in the case of computing the relevance term, as the weights of the relations with the target. Thus, according to our approach, the mth variable that will be selected next among the candidate variables

must satisfy the condition:

$$S_m = \underset{x_j \in X - S_{m-1}}{\arg\max} \left[\sum_{u \in F_j} \left[I(f_{j,u}(x_j); t) \cdot \rho_u^2 \right] - \frac{1}{m-1} \sum_{x_i \in S_{m-1}} \sum_{u \in F_j, v \in F_i} \left[I(f_{j,u}(x_j); f_{i,v}(x_i)) \cdot \rho_u^2 \cdot \rho_v^2 \right] \right]. \quad (2.40)$$

For the computation of KCCAmRMR, all the correlated functions among the variables and between each variable and target are found using KCCA. Then, these correlated functions are used instead of using the features themselves as inputs to mRMR.

3. ENSEMBLE CANONICAL CORRELATION ANALYSIS

Ignoring its various extensions to more than two views, CCA uses these two views of the same underlying semantics as complex labels to guide the search of maximally correlated projection vectors (covariates). Therefore, CCA can overfit the training data, meaning that different correlated projections can be found when the two-view training dataset is resampled. Although, to avoid such overfitting, ensemble approaches that utilize resampling techniques have been effectively used for improving generalization of many machine learning methods, an ensemble approach has not yet been formulated for CCA. The existing studies embracing both CCA and ensembles are similar to ensemble classification idea. They do not combine multiple sets of covariates into a final set of covariates but only apply CCA to extract covariates on each subsample separately, train a classifier on this reduced set of dimensions defined by the covariates, and then combine the predictions of such separate classifiers. In this chapter, we propose an ensemble CCA (ECCA) method for obtaining a final robust set of covariates by combining multiple sets of covariates extracted from subsamples. The aim of ECCA method is reaching a final set of covariates by combining sets of covariates obtained from various resamplings.

The remaining of this chapter is organized as follows: In Section 3.1, the existing robust canonical correlation analysis methods are summarized. Section 3.2 reviews the ensemble idea and its applications to classification, clustering, and CCA problems. Then, in Section 3.3 our proposed ensemble solution for CCA is given.

3.1. Robust CCA Methods

From the computational point of view, in the traditional generalized eigenvalue formulation of CCA, the canonical vectors are computed based on the within-set and between-set sample covariance matrices which can be very sensitive to outliers and noisy samples [24]. Based on these matrices with poor generalization, it has been shown that CCA can tune to dummy dependencies in the training set that do not

hold in the test set without proper regularization [25, 26]. The studies that deal with this sensitive sample covariance matrix problem can be categorized into three groups: (i) reducing the dimension of each view independently using Principal Component Analysis (PCA) before feature fusion as a preprocessing step, (ii) iterative alternating regression approaches which avoid the use of sample covariance matrices, (iii) and robust CCA approaches which utilize the robust estimation of the sample covariance matrices used in the computation of canonical vectors. The related studies with robust CCA methods are be summarized in this section.

3.1.1. PCA plus CCA

The application of PCA before classification/regression and feature extraction tasks with the aim of avoiding over-fitting problem and dealing with curse of dimensionality is a very common preprocessing step in the field of machine learning. Also PCA has been used before CCA to reduce the dimensions of the views independently especially in face recognition applications due to small sample size problem and high dimensionality of image vectors. As stated in Section 2.1, in the traditional formulation of CCA, the canonical vectors are computed based on the sample covariance matrices, and in face recognition problems, it is very difficult to obtain enough number of samples so as to avoid the singularity of the sample covariance matrix. In 2005, Sun *et al.* [21] used PCA plus CCA method to deal with the singularity of the covariance matrix problem and also obtained generalizable canonical variables. They extended the applicable range of CCA by applying PCA to each view for dimensional reduction and extracting covariates with CCA in the PCA transformed space.

In another study, Sun *et al.* [34] applied the same PCA/CCA strategy to Generalized CCA (GCCA) which incorporates the class information of the training samples to improve the discriminative ability. They firstly extracted two groups of feature sets on ORL and Yale face image datasets, then used PCA to reduce the dimension of two groups of feature sets, finally used their proposed GCCA algorithm to extract discriminative features in the transformed low-dimensional feature spaces. In the same year, He *et al.* [33] proposed to use a very similar PCA plus CCA strategy, and extended the

case to KPCA plus CCA to extract nonlinear features which is equivalent to Kernel Fisher Discriminant Analysis in nature. They tested the PCA/KPCA+CCA methods on ORL face database which contains images from 40 individuals, each providing 10 different images. They fed the extracted features with PCA/KPCA+CCA to k-nearest neighbor classifier, and showed that KPCA/PCA+CCA significantly outperform Fisherface and KPCA+CCA is a little better than PCA+CCA. Yang *et al.* [19] proposed a learning based $2D$-$3D$ face matching method using CCA to learn the mapping between 2D face image and 3D face data, and they also applied PCA on both $2D$ face image and $3D$ face data before the feature extraction step to avoid the curse of dimensionality and reduce noise. Although it has been shown that PCA can be applied before CCA as a preprocessing step to avoid the overfitting and singularities of covariance matrices, some of the important information regarding the correlations between the views might be lost in this "blind" dimensionality reduction process since, unlike CCA, the PCA method does not have the same ultimate goal of preserving the interrelations between the views. Besides, how much the variance of the views must be preserved is also another parameter that must be determined to apply PCA.

3.1.2. Alternating Regression

The alternating regression (AR) approach is firstly described by Wold [81]. The motivation of alternating regression approach is implementing CCA without the use of sample covariance matrices, which are sensitive to outliers and nosiy samples. Wold has already mentioned that AR can be used to estimate the canonical variates as a solution to CCA [25]. Given two multi-dimensional datasets,

$$X = [x_1, x_2, \cdots, x_N] \in \mathbb{R}^{m \times N}$$
$$Y = [y_1, y_2, \cdots, y_N] \in \mathbb{R}^{n \times N}$$

where m and n are the number of features of X and Y, respectively, N is the number of instances, the maximization problem of CCA is

$$(w_x, w_y) = \arg\max_{a,b} corr(a^T X, b^T Y). \tag{3.1}$$

Here, w_x and w_y are called canonical vectors of X and Y, respectively. The AR approach starts by obtaining an initial value for one of these canonical vectors. Suppose that we have an initial value for w_x. This value can be obtained by assigning random values to the vector elements or by applying PCA on X and getting the principal component that corresponds to the largest eigenvalue [25]. Now, the maximization problem given in Equation 3.1 can be written as:

$$w_y = \arg\max_{b} corr(w_x^T X, b^T Y). \tag{3.2}$$

Then the first value of w_y canonical vector, w_y^1 is obtained according to

$$w_x^T X = w_y^T Y + \gamma_1 + \varepsilon_1 \tag{3.3}$$

by regressing the univariate $(w_x^0)^T X$ on Y where w_x^0 is the initial value of w_x. In the next iteration, w_x^1 is obtained by regressing $(w_y^1)^T Y$ on X:

$$w_y^T Y = w_x^T X + \gamma_2 + \varepsilon_2. \tag{3.4}$$

This alternating procedure is iterated until convergence. After each iteration, the estimated regression coefficients are normalized. It must be noted that the regression estimators in the above scheme need to be robust. Branco *et al.* [25] suggested to use a weighted L_1 estimator for this purpose which has already been used in an application of AR by Croux and Filzmoser [82]. Branco *et al.* [25] also extended least squares alternating regression scheme to obtain higher order canonical variates.

In 1998, Lai and Fyfe [30] investigated a neural network implementation of CCA.

The inputs of their neural model is X and Y views and the objective function that will be maximized is $\mathbb{E}[(w_x^T X)(w_y^T Y)]$ where $\mathbb{E}[]$ denotes the expectation which will be taken over the joint distribution of X and Y. Using gradient ascent, they extract the update rules for the weights of their neural model. They tested their method on an artificial dataset which has been generated by drawing the samples of the views from zero-mean Gaussian distribution and introducing correlations between the views. They also used a real data that comprises exam grades of 88 students. In 1999, Lai and Fyfe [83] extended their neural implementation of CCA by maximising the correlation between outputs when such outputs are a nonlinear function of the inputs. They investigated a particular case of maximisation of $\mathbb{E}[(w_x^T X)(w_y^T Y)]$ when the samples of view Y are a nonlinear function of the view X's inputs. They also demonstrated the success of their method where the correlation extends to more than two input data sets.

Hsieh [31] proposed a nonlinear canonical correlation analysis method which is constructed using three feedforward neural networks. He designed the first network as a double-barreled architecture with an unconventional cost function, which maximizes the correlation between the two output neurons (the canonical variates). He used hyperbolic tangent function (or the sigmoidal function) to provide the nonlinearity as transfer function in the hidden layer of this network. The other two networks map from the outputs of the first network back to the original two sets of variables. A different algorithm other than neural networks can also be used in the implementation of AR. For example, Ryder and Favorov [74, 75, 78, 84] suggested an unsupervised machine learning tool, SINBAD (Set of INteracting BAckpropagating Dendrites), which is a basic computational strategy for finding the functions of dependencies between variables as similar to CCA. In [40], we used Support Vector Machines (SVMs) in the SINBAD model to learn the nonlinear relations among two sets of variables. In its implementation of SINBAD in [40], the learnt functions between the sets of variables, f and g, are kernel functions as we used kernel-SVMs for SINBAD dendrites. We acquired nonlinear dependences using RBF-kernel.

3.1.3. Robust Estimation of Covariance Matrix

The robust estimation of sample covariance matrices in CCA computation is a straightforward technique that can be used to overcome the sensitivity problem of CCA to outliers and noisy samples. As shown in Section 2.1, the canonical vectors between views X and Y are obtained using the eigenvectors corresponding to the largest eigenvalues of $C_{XX}^{-1}C_{XY}C_{YY}^{-1}C_{YX}$ and $C_{YY}^{-1}C_{YX}C_{XX}^{-1}C_{XY}$. The robust estimation of covariance matrix idea is based on computing the within-set covariance matrices, C_{XX} and C_{YY}, and between-set covariance matrices, C_{XY} and C_{YX}, using a robust covariance estimator.

M-estimators and Minimum Covariance Determinant (MCD) estimator methods are two alternative robust covariance estimators which can be used in CCA computation. Karnel [27] proposed a robust CCA method by computing the covariance matrices using M-estimators [85]. M-estimators are a maximum likelihood type estimator which aim to reduce the effect of outliers by replacing the residuals, r, with a function of the residuals, ρ, which must satisfy the following properties:

- $\rho(r) \geqslant 0$, $\forall r$ and has a minimum at 0,
- $\rho(r) = \rho(-r)$, $\forall r$,
- $\rho(r)$ increases with the increasing values of r, but does not get too large as r increases.

The standard least squares method where $\rho(r) = r^2$ is not a robust M-estimator since outliers have strong effects in the minimization of the function which results in overfitted parameters. Because of the poor robustness properties of M-estimators in higher dimensions [25], Croux and Dehon [28] proposed to use the Minimum Covariance Determinant (MCD) estimator of Rousseeuw [86] to robustly estimate the population covariance matrices in CCA computation. The MCD estimator is based on choosing a subset of the data which minimizes the determinant of the sample covariance matrix. The number of samples in this subset is typically determined as $\lfloor 0.5N \rfloor$ or $\lfloor 0.75N \rfloor$ where N is the total number of samples [25] in the original dataset. Rousseeuw and

Van Driessen [87] proposed a fast algorithm to find the optimal subset of samples to compute the MCD estimator, but they did not provide a large sample theory for the Fast-MCD (FMCD) estimator. In a more recent study, Zhang *et al.* [88] proposed three robust estimators of multivariate location and dispersion, and used one of these to create a robust method of CCA. They conducted two simulation studies to compare eight different CCA methods which are based on different robust sample covariance matrix estimations.

3.2. Ensemble Idea

The ensemble idea is based on combining multiple models (classifiers, regressors, or clusterings) to obtain an improved final model [89]. The final model is expected to have higher prediction accuracy in classification problems [50] and more robust cluster solutions in clustering problems [90] than could be obtained from any of the individual models.

3.2.1. Ensemble Learning

Ensemble learning is a recently popular multi-learner system, in which multiple individual learners are trained on the same task and a better predictive model than all of the individual learners is aimed to be obtained by combining the predictions of the individual learners. The set of individual learners may be constituted using:

- different learning algorithms on the training set,
- an algorithm with different hyper-parameters,
- different feature representations of the same data which can be obtained naturally from different sensors (e.g. chemical and biological views of data in drug discovery, or acoustic features and motion of lip region in speech recognition) or by applying different feature extraction algorithms on the original data or by artificially dividing the original data into groups in order for utilizing multiple predictors on each view,
- different subsamples of the dataset by drawing random training sets [50].

Early works on ensemble learning aimed at improving the accuracy and robustness of classifiers and regressors [59, 91–94]. In 1990, Hansen and Salamon [91] proved that the likelihood of an error using majority voting strategy to combine predictions will monotonically decrease with the increasing number of ensemble members if each member of the ensemble, i.e. feature subset, can get the right answer more than half the time, and the responses of members are independent. In other words, both theoretical [91, 95], and empirical [96–98] research on ensemble learning proved that in order to obtain a better predictive final model, the individual learners must be accurate and diverse enough. Diversity of the ensemble can be increased by creating individual members which make their errors on different parts of the input space. Many diversity measures have been proposed due to its important characteristic in classifier combination. However, diversity has still no strict definition or generally accepted formal measure [99]. As the accuracy of the members tend to decrease with the increasing diversity, the studies that incorporate the accuracy and diversity within a single measure are popular [100]. Fundamental machine learning tasks for ensemble learning are harder than traditional single view learning because not only the accuracy of the system but the diversity among the members of the ensemble must also be taken into account. In one of those studies, Opitz [101] proposed a genetic algorithm based feature selection method that aims at finding a set of feature subsets that will promote diversity among the ensemble's classifiers. They compared their method with the traditional and powerful ensemble approaches of AdaBoost and Bagging on various datasets and showed the utility of feature selection for ensembles.

In 2003, Bryll *et al.* [102] presented an ensemble construction technique, called attribute bagging. This method constitutes the individual members of the ensemble by partitioning the features instead of the samples in order to increase the diversity among the individual classifiers. In the attribute bagging method, after determining an appropriate attribute subset size, subsets of features are selected randomly. In this way, the projections of the training set are obtained on which the ensemble classifiers are built. They compared the performance of attribute bagging method with sample based ensemble methods such as bagging on a hand-pose recognition dataset, and showed that it gives consistently better results than bagging both in terms of accuracy

and stability.

Ensemble learning is based on employing separate classifiers on each feature subset and combining the predictions of the views using techniques such as voting and stacking [103]. The final prediction, y, of an ensemble is given by

$$y = \sum_{i=1}^{M} w_i d_i \tag{3.5}$$

satisfying

$$w_i \geq 0, \forall i \text{ and } \sum_{i=1}^{M} w_i = 1, \tag{3.6}$$

where w_i is the weight of the prediction of ith ensemble member, d_i is the prediction of ith ensemble member, and M is the total number of individual members in the ensemble [50]. The weight of the vote of each ensemble member, i.e. w_i, is equal in the simple voting scheme ($w_i = \frac{1}{M}$). The class with the maximum number of votes is the final prediction of the ensemble. This voting strategy is called majority voting for two class classification problems. Instead of using the hard label predictions, more sophisticated predictions as the measure of how much confident the ensemble member is for its prediction can be used to obtain the final predictions. In the stacking approach, the weights of the individual member predictions are learned by another learner which does not need to be linear. In Figure 3.1, the training scheme of an ensemble with two individual members is shown. In the individual learning step, firstly, the classifiers are trained independently, and the obtained models for each member (i.e. view) are applied on the training set which gives the class posterior probability estimates for the training examples. This mapping can be defined as:

$$\overrightarrow{f}_{v,0} : X_v \mapsto P^c \tag{3.7}$$

where $P = [0, 1] \subseteq \mathbb{R}$ and $\overrightarrow{f}_{v,0}$ denotes the function of the initial model of view v defined from X_v, i.e. the original features of view v, to the class posterior probability

estimates (a p dimensional vector, where p is the number of classes). In the combination step, the probability estimates of the views are combined using a combination technique such as voting or stacking.

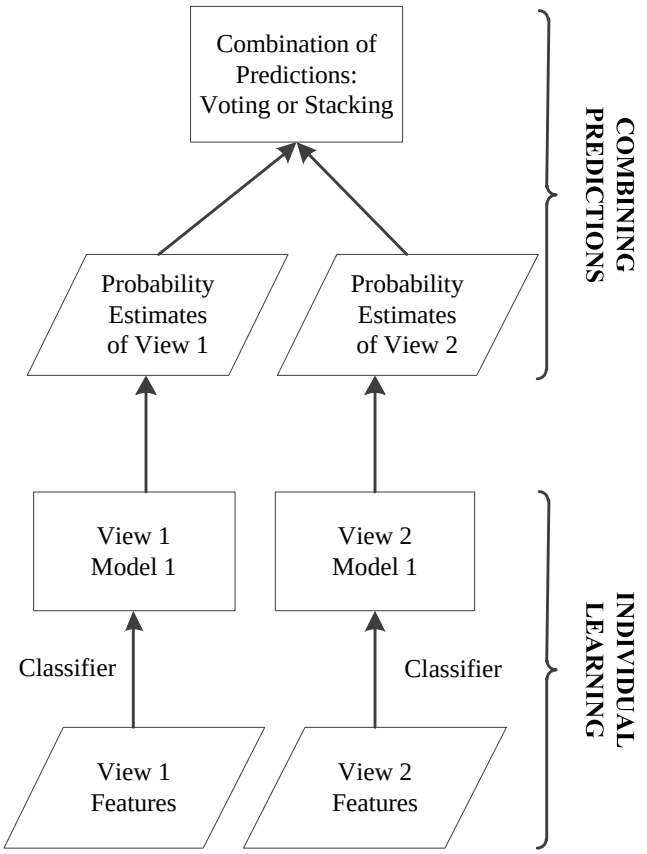


Figure 3.1. Ensemble learning with two individual members.

3.2.2. Proposed Parallel Interacting Multi-view Learning Method

In recent years, ensemble learning has gained considerable interest in predictive tasks regarding high dimensional datasets which are available in many fields such as biomedical engineering and bioinformatics [50, 104–106]. As a preliminary study of this thesis, considering that the classical ensemble approach does not take into account conditional interdependences among the views, we present a two stage supervised multi-view learning technique called Parallel Interacting Multi-view Learning (PIML) [42, 43]. The classical ensemble techniques are expected to work well when there are no conditional dependences (given the class-label) among the views since the views do

not interact during their individual learning processes. The proposed method PIML enables view interaction (in parallel) the training process, not only for avoiding the curse of dimensionality but also for modeling at least some of the interdependences among the views [43]. In other words, PIML addresses the following drawbacks of the existing multi-view learning approaches: (i) the lack of training phase interaction problem of the classical ensemble learning approach, (ii) the curse of dimensionality problem of the approach which merges the views of the dataset and treats it as if it consists of a single view. The proposed PIML method can also be used to combine the covariates extracted with our proposed ensemble CCA or discriminative alternating regression methods.

In the architecture of PIML, the views interact in parallel during the training process for modeling the interdependences among the views, and also the curse of dimensionality is avoided since only the probability estimates of other views as a summary information are used instead of high-dimensional original input space. The main idea is as follows: the classifier of each view uses the input variables from its own view along with the predictions (outputs) of the classifiers of the other views. In other words, each view uses the summary of information in the other views and evaluates its own input features (again) this time also by taking into account the predictions obtained from classifiers trained, in a similar fashion, on the other views. This technique increases the individual accuracies (sufficiency) of the views by taking the class posterior probability estimates of the other views during its second training phase, and also aims to preserve the diversity of the views by merging the original features of the individual views only with the summary information of the other views. Therefore, PIML approach is expected to reach higher accuracy than its counterparts that merge all the variables of all the views or combine their predictions after the individual learning process, i.e. ensemble methods.

If we used the probability estimates of all the views directly, we can only create a single stacking network. Thus, our PIML strategy resembles blocking [49], which is an experimental design strategy which produces similar experimental conditions to compare alternative stochastic configurations in order to be confident that observed

differences in accuracy are due to actual differences rather than to fluctuations and noise effects. Using each view's raw features along with the probability estimates of other views at least corresponds to creating multiple versions of stackings (yet possibly still using the same classifier model), thus increasing the number of blocking configurations.

The PIML architecture, which can be implemented in batch and in online learning form, is shown in Figure 3.2. According to the simplified two-part batch implementation of PIML, firstly, each view is trained independently using a classifier (support vector machines in this thesis). Then, the obtained models for each view are applied on the training set and the class posterior probability estimates of each view for the training examples are obtained. This mapping is defined as given in Equation 3.7.

The interaction of the views is then accomplished by combining the original features of each view with the class posterior probability estimates of the other views on the training set. In other words, the input of the second-pass classifier for each view consists of the original features of the view plus the class posterior probability estimates of the other views on those samples. Augmenting the features of a view by the class posterior probability estimates as 'the summary of the information that the other views possess about the target', a new model for each view is obtained. These second (or higher order) models can be described recursively as below:

$$\overrightarrow{f}_{v,i+1} : \left(X_v, \left[\overrightarrow{f}_{w,i} \right]_{\forall w \neq v} \right) \longrightarrow P^c \tag{3.8}$$

where $i+1$th model of view v is a function defined from X_v augmented with the class posterior probability estimates of the ith models of the other views to the improved estimates of class posterior probabilities. Thus, PIML can be seen, at least, as an ensemble of stacking networks, in which rather than creating a single stacking that uses all the estimates from all the views, the estimate of each view is utilized multiple times, thus leading to multiple stackings to be ensembled, which in turn leading to higher number of blocking configurations and more confident posterior probability estimates.

When PIML is used in batch mode, we observed that generally, models show

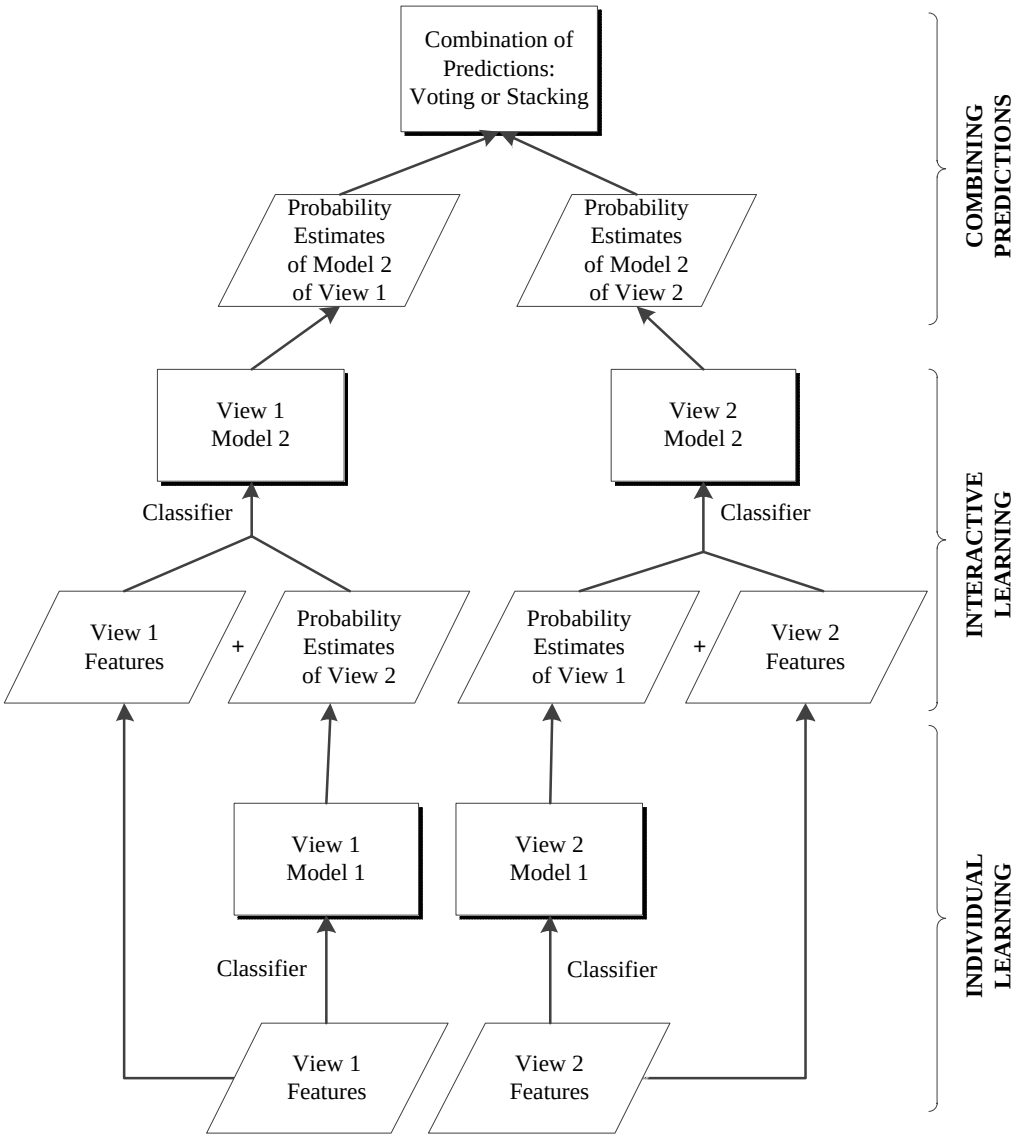


Figure 3.2. Two pass PIML architecture with two views. Each view has a classifier that feeds its output into the classifiers of the other views.

substantial improvement over the initial ones. However, the third pass results in approximately the same accuracy. Therefore, in our PIML experiments given in Section 5.5, we use two pass implementation of PIML in our experiments.

3.2.3. Cluster Ensembles

In the latter times, the ensemble approaches have been applied to combining multiple clusterings, also known as the cluster ensemble problem. Cluster ensemble problem was first addressed by Strehl and Ghosh [90]. The cluster ensemble approach is based on combining the cluster labels which are obtained by applying a single clustering approach [50, 107] to each ensemble set. While simple techniques such as majority voting can be used to combine the predictions of multiple classifiers, for cluster ensembles we need more sophisticated tools since cluster labels assigned by each single clustering are only symbolic, known as the cluster correspondence problem [108–110].

3.2.3.1. Single Clustering Algorithms. The clustering solutions of the convergent clustering methods such as k-means depend on the starting cluster seeds. The k-means [111] clustering algorithm is a non-hierarchical clustering procedure which aims to partition the observations into k clusters such that each observation belongs to the cluster with the nearest mean. In k-means clustering algorithm, the aim is to partition the observations into k clusters so that the within-cluster sum of squares is minimized. The within-cluster sum of squares (WCSS) is computed as:

$$WCSS = \sum_{i=1}^{k} \sum_{x_j \in S_i} \| x_j - \mu_i \|^2, \tag{3.9}$$

where S_i is the set of observations of cluster i, and μ_i is the mean of points in set S_i. The standard algorithm of k-means identifies the clusters using an iterative refinement technique consisting of two steps: firstly the mean of the clusters are computed and each observation is assigned to the cluster with the nearest mean, and secondly the means of the clusters are updated as the centroid of the observations in the cluster. The algorithm is terminated when the assignments of the observations to the clusters remain same between two iterations. The Forgy Method [112] can used for the initialization of the means in which the means are randomly selected among observations. The k-means clustering algorithm is heuristic, so the global optimum solution is not guaranteed. Because of the built-in randomness of the algorithm, a different cluster solution is

obtained in each run of the algorithm.

On the other hand, the hierarchical clustering methods are not convergent. In general terms, these methods aim to build a hierarchy of clusters. In the agglomerative type hierarchical clustering strategy, which is a 'bottom-up' approach, each observation starts in its own cluster, and the most similar pairs of clusters are merged into a single parent cluster. This process is repeated until the maximum number of clusters that will be kept in the hierarchical tree is formed. There is no need for iterative optimization procedure for the convergence of hierarchical clustering, so different runs of hierarchical clustering techniques with the same observations and parameters result in the same clustering indices. However, these methods are very sensitive to outliers [113].

3.2.3.2. Co-association Matrix based Cluster Ensembles. To overcome the variations of the single clustering methods and obtain robust cluster solutions, co-association matrix based cluster ensembles [90] approach is employed. The aim of the cluster ensemble approaches is to obtain a final robust and reproducible cluster solution by combining the solutions of many single clusterings. These single clusterings may be found using different initialization conditions, constituting different subsets of observations with resampling techniques such as bootstrapping, or applying different clustering algorithms to the original input data.

The cluster ensembles method based on co-association matrix [90, 114–116] is briefly summarized as follows: Suppose that a clustering algorithm is run B times on a dataset X with different starting conditions, and partitions $S = S_1, S_2, \ldots, S_B$ are obtained. Each element of the co-association matrix, i.e. the similarity between two objects x_1 and x_2, is defined as:

$$\mathrm{sim}(x_1, x_2) = \frac{1}{B} \sum_{i=1}^{B} \theta(S_i(x_1), S_i(x_2)), \tag{3.10}$$

where $\theta(a, b) = 1$ if $a = b$, $\theta(a, b) = 0$ if $a \neq b$, and $S_i(x_1)$ is the partition of x_1. Similarity between a pair of objects is simply represented by number of clusters shared

by these objects in the partitions $S_1, \ldots, S_B$ [117–119]. The agglomerative types of hierarchical clustering techniques such as single-link, average-link or complete-link are applied to the co-association matrix to obtain the final mostly representative partition [118, 119].

To obtain multiple clusterings, the k-means clustering algorithm can be run multiple times on the whole training set with different cluster seeds, and the hierarchical clustering can be run multiple times on the selected subset of data points using bootstrapping data resampling technique to create partitions. The cluster labels for each partition are obtained by applying a clustering algorithm, then the cluster indices of multiple clusterings are combined into the co-association matrix, and finally the consensus partition is obtained applying a clustering algorithm on the co-association matrix.

3.2.3.3. Proposed Cluster Stacking Method. We propose a cluster ensembles method for multi-view datasets, called cluster stacking, to combine the multiple clustering solutions of the views. Cluster stacking approach [39] is based on augmenting the clustering indices of all the clustering trials into a consensus matrix and using this augmented consensus partition as the final partition.

We construct an augmented cluster index matrix to combine the multiple clusterings of the views. This matrix is defined as:

$$\mathbf{C} = \begin{bmatrix} c_1^1 & c_2^1 & \cdots & c_V^1 \\ c_1^2 & c_2^2 & \cdots & c_V^2 \\ \vdots & \vdots & \vdots & \vdots \\ c_1^B & c_2^B & \cdots & c_V^B \end{bmatrix} \in \mathbb{R}^{(N \cdot B) \times V}, \tag{3.11}$$

where $c_i^j \in \mathbb{R}^{N \times 1}$ represents the cluster indices that belong to jth clustering run of view i, B is the total number of clustering runs, N is the number of samples, and V is the total number of views in the multi-view dataset. In order to avoid the label correspondence problem between different clusterings of a view, we used completely different

clustering indices in each partition. For this purpose, the c_i^j vector is constructed by assigning cluster indices between $(j-1)\times k+1$ and $j\times k$, so that the cluster indices get different values for different runs of the same view. Thus, each view of the multi-view dataset is reduced to a single variable. This mapping can be defined as:

$$V_i \in \mathbb{R}^{N\times d_i} \mapsto c_i \in \mathbb{R}^{(N\cdot B)\times 1}, \tag{3.12}$$

where V_i denotes the original data matrix of ith view, and d_i is the number of features in V_i. The obtained augmented cluster index matrix, $\mathbf{C}$ can be used as input to a clustering algorithm, and final partition of the objects can be obtained. Alternatively, it can be used as input to a feature selection algorithm to identify the most relevant views with the class/target variable.

The augmented cluster index matrix given in Equation 3.11 resembles the covariate correspondence matrix (see Section 3.3) that we propose to solve the covariate correspondence problem encountered in ensemble CCA approach. In fact, the cluster index matrix that we use to combine multiple clusterings of multiple views provided a basis for solving the covariate correspondence problem of ensemble CCA using the covariate correspondence matrix given in Equation 3.14.

3.2.4. Existing Ensemble Approaches to CCA

The existing studies that combine the ensemble idea and CCA are based on separately extracting features from each subsample pair with CCA, then training a classifier with the covariates, and finally combining the predictions of separate classifiers. In other words, they do not result in a final set of covariates by combining multiple sets of covariates. Lau *et al.* [120] developed such an ensemble canonical correlation prediction method. In his study, the term ensemble has been used to represent different forecasts obtained by applying CCA to multiple sea surface temperature data obtained from different ocean basis. The aim of this study was estimating summer precipitation over the United States. Mo and Thiaw [121] applied the same ensemble canonical correlation prediction idea in a meteorological study to predict summer

rainfall over the Sahel. As predictors, they used the global sea surface temperature, 200-hPa streamfunction with zonal means removed, and forecasts or simulations from the climate models. They performed CCA for each of these variables separately, so obtained an ensemble of predicted precipitation fields, and combined these individual CCA results by getting the equal weighted average of its members. Their observations showed that each member has forecast skill over the different parts of the Sahel, and therefore the ensemble mean of these members has higher skills than each of its individual members. After this study, Mo [122] used the same ensemble CCA method to predict summer (July–September) and winter (January–March) seasonal mean surface temperature (Tsurf) with different predictors over the United States. He compared the simple ensemble forecast and the superensemble forecast. While the simple ensemble mean is the equally weighted average of the CCA results, the weighting function for the superensemble forecast was determined by linear regression analysis. He concluded that both ensemble forecasts improve skill of the individual members, and on average, the superensemble gives the best performance.

In a more recent study, Zhang and Zhang [61] used ensemble idea for CCA with the aim of extracting discriminative features. They proposed a discriminative CCA method similar to the DCCA method of Sun *et al.* [36]. However, instead of using all cross-view correlations between within-class examples, Zhang and Zhang [61] used random cross-view correlations between within-class examples to integrate the class information into CCA. This random process enabled them to construct multiple feature extractors based on CCA. They fused those feature extractors and proposed a method called random correlation ensemble (RCE) for multi-view ensemble learning. While in classical CCA approach only pairwise correlation terms are considered, DCCA of Sun *et al.* [36] takes all within-class correlation terms into account. Unlike these methods, RCE method includes only a random subset of within-class correlation terms to extract diverse correlated features. Then, in RCE method, the obtained individual sets of correlated features are fed to classifiers separately, and the predictions of these classifiers are combined using a combination technique such as majority voting or stacking. The superiority of RCE method is expected to be originated from the fact that the diversity of the ensemble members have been increased by creating them through including

random within-class correlation terms. They validated the effectiveness of RCE by comparing the discriminative ability of its features with those of CCA, discriminant CCA (DCCA) of Sun *et al.* [36], trivial ensembles of CCA and DCCA which adopt standard bagging and boosting strategies for ensemble learning. However, RCE, as the former existing ensemble CCA approaches, does not combine multiple sets of covariates to obtain a final set of covariates. It only resamples the two views to create an ensemble of subsamples, applies CCA between each subsample pair individually to obtain a set of covariates from each subsample, then feeds the covariates of each subsample pair separately into classifiers, and finally combine the predictions of the classifiers.

3.3. Proposed Ensemble Canonical Correlation Analysis Method

Ensemble approaches have been effectively used in the field of machine learning to improve the generalization capacity of classifiers and clusterings as mentioned in Section 3.2.1 and Section 3.2.3. However, the existing studies embracing both CCA and ensembles merely utilize CCA to extract covariates on each subsample separately, train a classifier on this reduced set of dimensions defined by the covariates, and only then combine the predictions of such separate classifiers [61, 121]. In other words, combining multiple sets of covariates into a final set of covariates has not yet been addressed in the literature. In this section, we first define the problem of combining multiple sets of covariates, which in its simplest terms refers to the problem of finding a final set of covariates by integrating sets of covariates extracted from CCA analyses on various subsamples. Then, we propose an ensemble CCA (ECCA) method to deal with this problem.

3.3.1. Ensemble Construction

Suppose that dataset D is composed of two views with N pairs of feature vectors as

$$D = d_i = \{(x_i, y_i), i = 1, 2, \cdots, N\},$$

where $d_i = (x_i, y_i)$, $x_i \in \mathbb{R}^m$, $y_i \in \mathbb{R}^n$, and N is the total number of instances in the whole sample. We can consider the dataset D as a block matrix of centered datasets

$$X = [x_1, x_2, \cdots, x_N] \in \mathbb{R}^{m \times N}$$
$$Y = [y_1, y_2, \cdots, y_N] \in \mathbb{R}^{n \times N}.$$

In the ensemble CCA approach, firstly, $D = (X, Y)$ is resampled to construct subsamples, D_*^i; thus producing X_*^i and Y_*^i pairs of subsamples of X and Y datasets. In the context of this thesis, we consider three ensemble construction methods for generating the ensemble from a given dataset D: bootstrap aggregating (BAGGING), delete-a-group jackknife (JACKKNIFE), and partitioning the dataset (PARTITION) but any other method can be used for this purpose. In bagging [92], a number of subsamples, $D_*^i, i = 1, 2, \ldots, B$, with N number of samples are generated by drawing random samples with replacement uniformly from the original training set D. Jackknife [123] method works by leaving out some portion of training data samples at a time and using the rest subset as a subsample of the ensemble. In partitioning method, the original training set D is randomly partitioned into B subsets, and thus B number of subsamples is constituted, $D_*^i, i = 1, 2, \ldots, B$, each with $\frac{N}{B}$ number of training samples.

3.3.2. Individual Sets of Covariates

The aim of ensemble CCA approach is to combine the individual sets of covariates into a final consensus set of covariates. For this purpose, we need to obtain the individual sets of covariates from the subsample pairs of the ensemble, which are constructed with one of the ensemble construction methods as given in Section 3.3.1.

The individual set of canonical vectors extracted from the ith subsample pair, X_*^i

and Y_*^i, can be denoted as

$$\begin{aligned} \mathbf{W}_x &= \{W_x^i\}_{i=1,2,\cdots,B} \\ \mathbf{W}_y &= \{W_y^i\}_{i=1,2,\cdots,B} \end{aligned} \tag{3.13}$$

where B is the total number of independent subsamples used, and W_x^i denotes the set of canonical vectors extracted using the ith ensemble set of view X, X_*^i. The matrices $\mathbf{W}_x$ and $\mathbf{W}_y$ of size $m \times (k \cdot B)$ and $n \times (k \cdot B)$ contain the canonical vectors of X and Y, respectively, in their columns where k is the total number of canonical vectors extracted from each ensemble set.

3.3.3. Covariate Correspondence Problem

After obtaining k canonical vectors of each ensemble set (subsample), the aim of ECCA method is to extract a final, consolidated k canonical vectors. However, we cannot simply merge the covariate sets horizontally into matrices as

$$\begin{bmatrix} (W_x^1)^T X_*^1 & (W_x^2)^T X_*^2 & \cdots & (W_x^B)^T X_*^B \end{bmatrix}$$
$$\begin{bmatrix} (W_y^1)^T Y_*^1 & (W_y^2)^T Y_*^2 & \cdots & (W_y^B)^T Y_*^B \end{bmatrix}$$

because, for example for the bootstrapping ensemble construction method, not all the samples are chosen for inclusion in each X_*^b and Y_*^b. When the bootstrapping method is applied, the size of the horizontally concatenated matrix becomes k by $(N \cdot B)$, and the set of covariates of size k by N extracted from any subsample of the ensemble belong to different resampled set of instances from the other subsamples. We call this "the sample correspondence problem". It must be noted that this problem also exists for the other construction methods such as partitioning and jackknife.

On the other hand, we cannot merge the set of covariates extracted from sub-

samples vertically either:

$$\begin{bmatrix} ((W_x^1)^T X_*^1)^T & ((W_x^2)^T X_*^2)^T & \cdots & ((W_x^B)^T X_*^B)^T \end{bmatrix}^T$$
$$\begin{bmatrix} ((W_y^1)^T Y_*^1)^T & ((W_y^2)^T Y_*^2)^T & \cdots & ((W_y^B)^T Y_*^B)^T \end{bmatrix}^T$$

because the canonical vectors extracted from the subsample pair X_*^i and Y_*^i are not guaranteed to match those extracted from the view pair X_*^j and Y_*^j, where $i \neq j$. Similar to the cluster correspondence problem [90], as the covariates extracted from different subsample pairs do not have to correspond or match, we call this phenomenon "the covariate correspondence problem".

3.3.4. Ensemble CCA

The ensemble CCA problem can be defined as combining multiple sets of covariates, which in its simplest terms refers to the problem of finding a final set of covariates by integrating sets of covariates extracted from CCA analyses on various subsamples. For this purpose, the set of covariates extracted from the subsamples of the ensemble must be combined in a common matrix which does not suffer sample and covariate correspondence problems. The algorithm of the proposed ECCA method to extract first pair of canonical vectors is shown in Figure 3.3.

Our proposed ECCA method is based on using the following covariate correspondence matrix for view X to solve both the sample and the covariate correspondence problems that are addressed in Section 3.3.3:

$$\mathbf{M_x} = \begin{bmatrix} (W_x^1)^T X_*^1 & (W_x^2)^T X_*^1 & \cdots & (W_x^B)^T X_*^1 \\ (W_x^1)^T X_*^2 & (W_x^2)^T X_*^2 & \cdots & (W_x^B)^T X_*^2 \\ \vdots & \vdots & \vdots & \vdots \\ (W_x^1)^T X_*^B & (W_x^2)^T X_*^B & \cdots & (W_x^B)^T X_*^B \end{bmatrix} \tag{3.14}$$

Input

N: Number of samples

X: $[x_1, x_2, \cdots, x_N] \in \mathbb{R}^{m \times N}$

Y: $[y_1, y_2, \cdots, y_N] \in \mathbb{R}^{n \times N}$

D: $d_i = \{(x_i, y_i), i = 1, 2, \cdots, N\}$

B: Number of subsamples

N_e: Number of samples in each subsample

Output

w_x: Final canonical vector of X

w_y: Final canonical vector of Y

ρ: correlation coefficient between final covariates of X and Y

for $i = 1$ to B **do**

Resample $D(X, Y)$ to construct X_*^i and Y_*^i pair of subsamples of X and Y

end for

for $i = 1$ to B **do**

$(W_x^i, W_y^i) = \arg\max_{a,b} corr(a^T X_*^i, b^T X_*^i)$

end for

$\mathbf{X}_* = \{X_*^i\}_{i=1,2,\cdots,B} \in \mathbb{R}^{m \times (N_e \cdot B)}$

$\mathbf{Y}_* = \{Y_*^i\}_{i=1,2,\cdots,B} \in \mathbb{R}^{n \times (N_e \cdot B)}$

$\mathbf{W}_x = \{W_x^i\}_{i=1,2,\cdots,B} \in \mathbb{R}^{m \times B}$

$\mathbf{W}_y = \{W_y^i\}_{i=1,2,\cdots,B} \in \mathbb{R}^{n \times B}$

$\mathbf{M_x} = \mathbf{W}_x^T \cdot \mathbf{X}_*$

$\mathbf{M_y} = \mathbf{W}_y^T \cdot \mathbf{Y}_*$

$(w_x, w_y) = \arg\max_{a,b} corr(a^T \mathbf{W_x}^T \mathbf{X}_*, b^T \mathbf{W_y}^T \mathbf{Y}_*)$

$\rho = corr(w_x^T \mathbf{W_x}^T \mathbf{X}_*, w_y^T \mathbf{W_y}^T \mathbf{Y}_*)$

Figure 3.3. Ensemble canonical correlation analysis algorithm to extract first pair of canonical vectors.

which can also be denoted as

$$\begin{aligned}\mathbf{M_x} &= \begin{bmatrix} \mathbf{W}_x^T X_*^1 & \mathbf{W}_x^T X_*^2 & \cdots & \mathbf{W}_x^T X_*^B \end{bmatrix} \\ &= \mathbf{W}_x^T \cdot \begin{bmatrix} X_*^1 & X_*^2 & \cdots & X_*^B \end{bmatrix} \\ &= \left[\mathbf{W}_x^T \cdot \mathbf{X}_*\right]. \end{aligned} \tag{3.15}$$

$\mathbf{X}_*$ is m by $(N^e \cdot B)$ matrix containing all the ensemble sets of view X where N_e is the number of samples in each $X_*^i, i = 1, 2, \cdots, B$.

The covariate correspondence matrix for view Y can also be written as:

$$\begin{aligned}\mathbf{M_y} &= \begin{bmatrix} (W_y^1)^T Y_*^1 & (W_y^2)^T Y_*^1 & \cdots & (W_y^B)^T Y_*^1 \\ (W_y^1)^T Y_*^2 & (W_y^2)^T Y_*^2 & \cdots & (W_y^B)^T Y_*^2 \\ \vdots & \vdots & \vdots & \vdots \\ (W_y^1)^T Y_*^B & (W_y^2)^T Y_*^B & \cdots & (W_y^B)^T Y_*^B \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{W}_y^T Y_*^1 & \mathbf{W}_y^T Y_*^2 & \cdots & \mathbf{W}_y^T Y_*^B \end{bmatrix} \\ &= \mathbf{W}_y^T \cdot \begin{bmatrix} Y_*^1 & Y_*^2 & \cdots & Y_*^B \end{bmatrix} \\ &= \left[\mathbf{W}_y^T \cdot \mathbf{Y}_*\right]. \end{aligned} \tag{3.16}$$

The covariate correspondence matrix idea is similar to the augmented cluster index matrix that we propose for integrating multiple clustering solutions of each view of a multi-view dataset. Each column of $\mathbf{M_x}$ is constituted by projecting all the subsamples onto the canonical vector which has been explored between the subsample pair that corresponds to that column. Thus, the covariate correspondence problem addressed in Section 3.3.3 is solved, and the final set of covariates of the ECCA method can be explored between $\mathbf{M_x}$ and $\mathbf{M_y}$. The architecture of ECCA is shown in Figure 3.4.

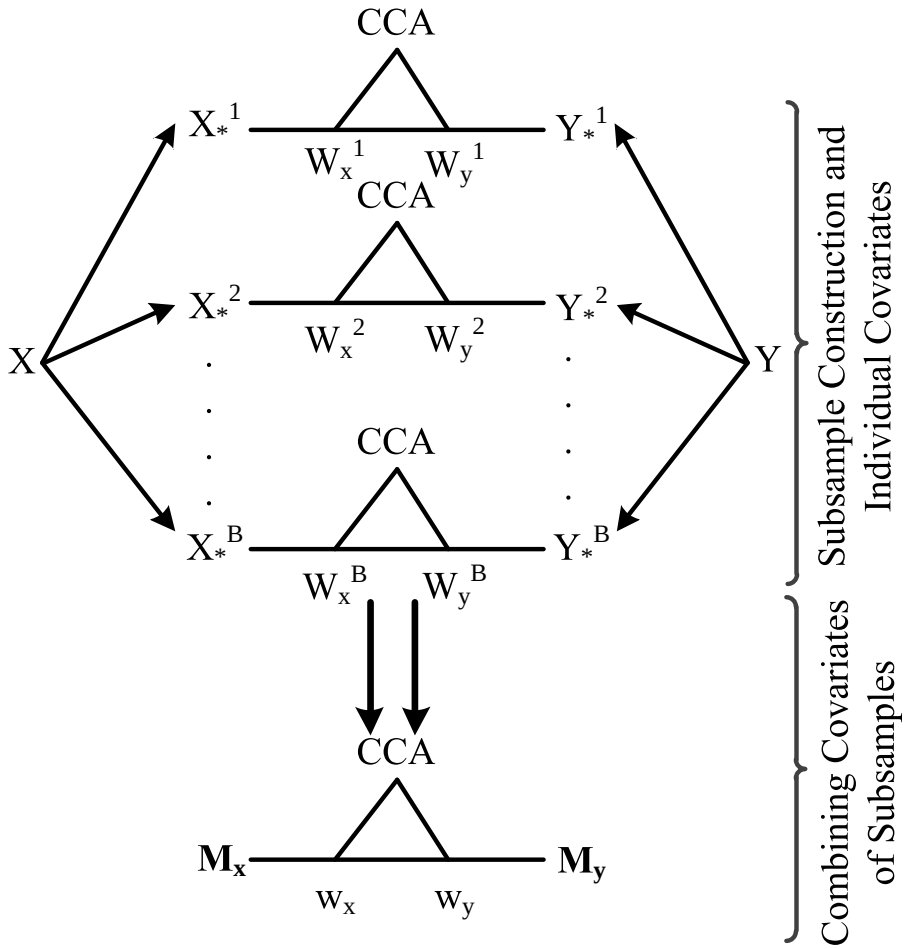


Figure 3.4. ECCA architecture.

Now, the training data of X and Y views are transformed to

$$\mathbf{W}_x^T \cdot \mathbf{X}_* \in \mathbb{R}^{(k \cdot B) \times N_e} \text{ and } \mathbf{W}_y^T \cdot \mathbf{Y}_* \in \mathbb{R}^{(k \cdot B) \times N_e},$$

respectively. As the standard CCA problem we can find the top k most correlated covariates of the two combined sets of subsample-covariates. Thus, the CCA problem is turned into maximizing the correlation between the linear combinations of $\mathbf{W}_x^T \cdot \mathbf{X}_*$

and $\mathbf{W}_y^T \cdot \mathbf{Y}_*$:

$$\begin{aligned}\rho &= \max_{w_x,w_y} corr(w_x^T \mathbf{W_x}^T \mathbf{X}_*, w_y^T \mathbf{W_y}^T \mathbf{Y}_*) \\ &= \max_{w_x,w_y} \frac{cov(w_x^T \mathbf{W_x}^T \mathbf{X}_*, w_y^T \mathbf{W_y}^T \mathbf{Y}_*)}{\sigma_{\mathbf{W_x}^T \mathbf{X}_*} \sigma_{\mathbf{W_y}^T \mathbf{Y}_*}}\end{aligned} \tag{3.17}$$

where *cov* denotes the covariance, and σ is the standard deviation.

The correlation expression given in Equation 3.17 can be rewritten as

$$\begin{aligned}\rho &= \max_{w_x,w_y} \frac{\mathbb{E}[(w_x^T \mathbf{W_x}^T \mathbf{X}_*)(w_y^T \mathbf{W_y}^T \mathbf{Y}_*)^T]}{\sqrt{\mathbb{E}[(w_x^T \mathbf{W_x}^T \mathbf{X}_*)(w_x^T \mathbf{W_x}^T \mathbf{X}_*)^T]\mathbb{E}[(w_y^T \mathbf{W_y}^T \mathbf{Y}_*)(w_y^T \mathbf{W_y}^T \mathbf{Y}_*)^T]}} \\ &= \max_{w_x,w_y} \frac{w_x^T \mathbb{E}[\mathbf{W_x}^T \mathbf{X}_* \mathbf{W_y}^T \mathbf{Y_*}^T] w_y}{\sqrt{w_x^T \mathbb{E}[\mathbf{W_x}^T \mathbf{X}_* \mathbf{W_x}^T \mathbf{X_*}^T] w_x w_y^T \mathbb{E}[\mathbf{W_y}^T \mathbf{Y}_* \mathbf{W_y}^T \mathbf{Y_*}^T] w_y}}\end{aligned} \tag{3.18}$$

in which $\mathbb{E}$ denotes the expectation.

Denoting the between-set sample covariance matrix by $\mathbf{C_{XY}}$, and within-set sample covariance matrices of $\mathbf{W_x}^T \mathbf{X}_*$ and $\mathbf{W_y}^T \mathbf{Y}_*$ by $\mathbf{C_{XX}}$ and $\mathbf{C_{YY}}$, respectively, we obtain

$$\rho = \max_{w_x,w_y} \frac{w_x^T \mathbf{C_{XY}} w_y}{\sqrt{w_x^T \mathbf{C_{XX}} w_x w_y^T \mathbf{C_{YY}} w_y}}. \tag{3.19}$$

Since re-scaling of the canonical vectors w_x or w_y does not affect the solution of the maximization problem given in Equation 3.19, the ECCA optimization problem given in Equation 3.17 can be rewritten as:

$$\begin{aligned}\rho &= \max_{w_x,w_y} \frac{w_x^T \mathbf{C_{XY}} w_y}{\sqrt{w_x^T \mathbf{C_{XX}} w_x w_y^T \mathbf{C_{YY}} w_y}} \\ \text{subject to } & w_x^T \mathbf{C_{XX}} w_x = 1 \\ & w_y^T \mathbf{C_{YY}} w_y = 1.\end{aligned} \tag{3.20}$$

Using the Lagrangian relaxation method, the above problem is reduced to:

$$\mathbf{C_{XY}C_{YY}^{-1}C_{YX}}w_x = \delta^2\mathbf{C_{XX}}w_x \tag{3.21}$$

which is an eigenvalue problem of the form $Ax = \delta Bx$. The canonical vectors, w_x and w_y, are obtained using the eigenvectors corresponding to the largest eigenvalues of $\mathbf{C_{XX}^{-1}C_{XY}C_{YY}^{-1}C_{YX}}$ and $\mathbf{C_{YY}^{-1}C_{YX}C_{XX}^{-1}C_{XY}}$. The projections of $\mathbf{W_x^T X}_*$ and $\mathbf{W_y^T Y}_*$ onto these canonical vectors, i.e. $w_x^T\mathbf{W_x^T X}_*$ and $w_y^T\mathbf{W_y^T Y}_*$, are called canonical variables (covariates) of ECCA.

4. DISCRIMINATIVE ALTERNATING REGRESSION

In principle, CCA, itself, is a dimensionality reduction technique which aims at finding maximally correlated projections of the views with each other rather than preserving the discriminative information of the views. In other words, although CCA features are used for classification, unlike supervised learning algorithms such as Linear Discriminant Analysis (LDA), it does not utilize the class labels in its formulation while fusing the views. On the other hand, the use of sensitive sample covariance matrices in the traditional formulation of CCA results in ungeneralizable dependencies which do not hold on unseen test examples.

In this chapter, we propose a method, D-AR (Discriminative Alternating Regression), which aims at exploring discriminative and robust features by incorporating the class labels into the view fusion framework. Besides, using a multilayer perceptron based alternating regression algorithm avoids the use of sensitive sample covariance matrices, so increases the generalization capacity of the discriminative features.

4.1. Discriminative Canonical Correlation Analysis

The class information of the samples is not exploited in the traditional formulation of CCA. The existing studies that aim to improve the discriminative ability of CCA features are based on incorporating the class labels into feature fusion and extraction framework. The straightforward approach to discriminative CCA (DCCA) has been proposed by Sun *et al.* [36]. DCCA method employs the class information by maximizing the correlation between feature vectors in the same class and minimizing the correlation between feature vectors belonging to different classes. Suppose that we have two different but related views,

$$X = [x_1, x_2, \cdots, x_N] \in \mathbb{R}^{m \times N}$$
$$Y = [y_1, y_2, \cdots, y_N] \in \mathbb{R}^{n \times N},$$

where N is the number of instances. Then, the optimization problem of DCCA can be written as:

$$\begin{aligned} \rho = \max_{w_x, w_y} (w_x^T K_w w_y - \eta \cdot w_x^T K_b w_y) \\ \text{subject to } w_x^T X X^T w_x = 1 \\ w_y^T Y Y^T w_y = 1, \end{aligned} \tag{4.1}$$

where K_w and K_b denote the correlations between the same classes and between different classes, respectively, such that:

$$\begin{aligned} K_w &= \sum_{i=1}^{p} \sum_{k=1}^{N_i} \sum_{l=1}^{N_i} (x_k^i - \overline{x})(y_l^i - \overline{y})^T \\ K_b &= \sum_{i=1}^{p} \sum_{j=1, j\neq i}^{p} \sum_{k=1}^{N_i} \sum_{l=1}^{N_j} (x_k^i - \overline{x})(y_l^j - \overline{y})^T, \end{aligned} \tag{4.2}$$

where p is the number of classes, N_i is the number of instances that belong to class i, x_j^i denotes the jth example in class i, and $\overline{x}$ and $\overline{y}$ denotes the global means of X and Y, respectively. The global means $\overline{x}$ and $\overline{y}$ can be computed as:

$$\begin{aligned} \overline{x} &= \frac{1}{N} \sum_{i=1}^{p} \sum_{j=1}^{N_i} x_j^i \\ \overline{y} &= \frac{1}{N} \sum_{i=1}^{p} \sum_{j=1}^{N_i} y_j^i. \end{aligned} \tag{4.3}$$

Representing the mean centered data, X and Y, and class label vector, e_{N_i} as

$$\begin{aligned} X &= [\underbrace{x_1^1, x_2^1, \cdots, x_{N_1}^1}_{\text{instances of class 1}}, \cdots, \underbrace{x_1^p, x_2^p, \cdots, x_{N_p}^p}_{\text{instances of class p}}] \\ Y &= [\underbrace{y_1^1, y_2^1, \cdots, y_{N_1}^1}_{\text{instances of class 1}}, \cdots, \underbrace{y_1^p, y_2^p, \cdots, y_{N_p}^p}_{\text{instances of class p}}] \\ e_{N_i} &= [\underbrace{0, \cdots, 0}_{\sum_{j=1}^{i-1} N_j}, \underbrace{1, \cdots, 1}_{N_i}, \underbrace{0, \cdots, 0}_{N - \sum_{j=1}^{i} N_j}]^T \in \mathbb{R}^{\mathbb{N}} \end{aligned} \tag{4.4}$$

the within-class correlation matrix, K_w, can be written as:

$$\begin{aligned} K_w &= \sum_{i=1}^{p} \sum_{k=1}^{N_i} \sum_{l=1}^{N_i} x_k^i y_l^{iT} \\ &= \sum_{i=1}^{p} (X e_{N_i})(Y e_{N_i})^T \\ &= X A Y^T \end{aligned} \tag{4.5}$$

where

$$\mathbf{A} = \begin{bmatrix} 1_{N_1 \times N_1} & & & & \\ & \ddots & & & \\ & & 1_{N_i \times N_i} & & \\ & & & \ddots & \\ & & & & 1_{N_p \times N_p} \end{bmatrix} \in \mathbb{R}^{N \times N} \tag{4.6}$$

is a blocked diagonal matrix, and

$$1_N = [1, \cdots, 1]^T \in R^N. \tag{4.7}$$

The between-class correlation matrix can be defined as:

$$\begin{aligned} K_b &= \sum_{i=1}^{p} \sum_{j=1, j \neq i}^{p} \sum_{k=1}^{N_i} \sum_{l=1}^{N_j} x_k^i y_l^{jT} \\ &= \sum_{i=1}^{p} \sum_{j=1}^{p} \sum_{k=1}^{N_i} \sum_{l=1}^{N_j} x_k^i y_l^{jT} - \sum_{i=1}^{p} \sum_{k=1}^{N_i} \sum_{l=1}^{N_i} x_k^i y_l^{iT} \\ &= (X 1_N)(Y 1_N)^T - X A Y^T \\ &= -X A Y^T \end{aligned} \tag{4.8}$$

where $(X 1_N) = 0$ and $Y 1_N = 0$ because both X and Y are centered data. Thus, we obtain the following relation between within-class and between-class correlation

matrices:

$$K_w = -K_b. \tag{4.9}$$

Now we can rewrite the optimization problem of DCCA given in Equation 4.1 as:

$$\begin{aligned} \rho &= \max_{w_x, w_y} w_x^T K_w w_y \\ \text{subject to } & w_x^T X X^T w_x = 1 \\ & w_y^T Y Y^T w_y = 1. \end{aligned} \tag{4.10}$$

The above problem is reduced to the following eigenvalue problems similar to CCA:

$$\begin{aligned} K_w (Y Y^T)^{-1} K_w^T w_x &= \lambda^2 X X^T w_x \\ K_w^T (X X^T)^{-1} K_w w_Y &= \lambda^2 Y Y^T w_y. \end{aligned} \tag{4.11}$$

The eigenvectors that correspond to the largest eigenvalues of the above problem are the extracted features of DCCA. Similar to LDA, DCCA has the limitation of extracting less number of features than the number of classes because the rank of K_w is $p-1$ for high dimensional data where the number of features in X and Y are greater than p.

Sun *et al.* [36] fed the DCCA features to k-Nearest Neighbor and Naive Bayes classifiers, and compared the discriminative ability of DCCA with CCA and Partial Least Squares (PLS) on text categorization, face recognition, and handwritten digit recognition datasets. They observed that DCCA features achieved higher classification accuracy than those of the CCA and Partial Least Squares (PLS) methods. However, it has recently been shown that [37] the projective directions explored by DCCA are equal to the ones obtained from LDA with respect to an orthogonal transformation. These relations between LDA and DCCA are elaborated by Shin and Park [37] utilizing the relation between CCA and LDA which is given in Section 2.2. Therefore, in our numerical experiments, we compared the discriminative ability of the D-AR features with that of LDA in the D-AR experiments given in Section 5.2.

4.2. Proposed Discriminative Alternating Regression Method

In this section, we propose a method, called Discriminative Alternating Regression (D-AR), to explore correlated and also discriminative features. D-AR utilizes two (alternating) multi-layer perceptrons, each with a linear hidden layer, learning to predict both the class-labels and the outputs of each other.

4.2.1. Architecture of D-AR Method

The architecture of our proposed D-AR method is based on the alternating regression approach [81] implemented by a multi-layer neural network with a "linear" hidden layer. The block diagram and MLP architecture of the D-AR method are shown in Figure 4.1 and Figure 4.2, respectively. The input layer of the proposed MLP based D-AR Network (D-ARNet) consists of the view features. The input layer of each D-ARNet is transformed into a low-dimensional subspace (hidden layer) such that the hidden layer is in turn transformable to the output layer containing output units that can mutually maximize their match with their corresponding counterparts of the other D-ARNet and also the predictor output units have maximum classification accuracy. Output units consist of class labels and correlated features. The correlated features are alternated between the two views whereas the original class labels are used in each iteration of the algorithm.

The aim of the method is obtaining a subspace of the input features (k hidden neurons where $k < m$ and $k < n$) in the hidden layer which preserves both the common information with the other view and the class information. In other words, the high-dimensional input space is reduced to a low-dimensional subspace in the bottleneck hidden layer [124] which is expected to possess the "correlated discriminative" information. The overall algorithm of the method is shown in Figure 4.3.

The network is forced to explore the discriminative information which is contained by both of the views. This correlated discriminative information is expected to be more reliable and generalizable especially on datasets with small sample size and high-

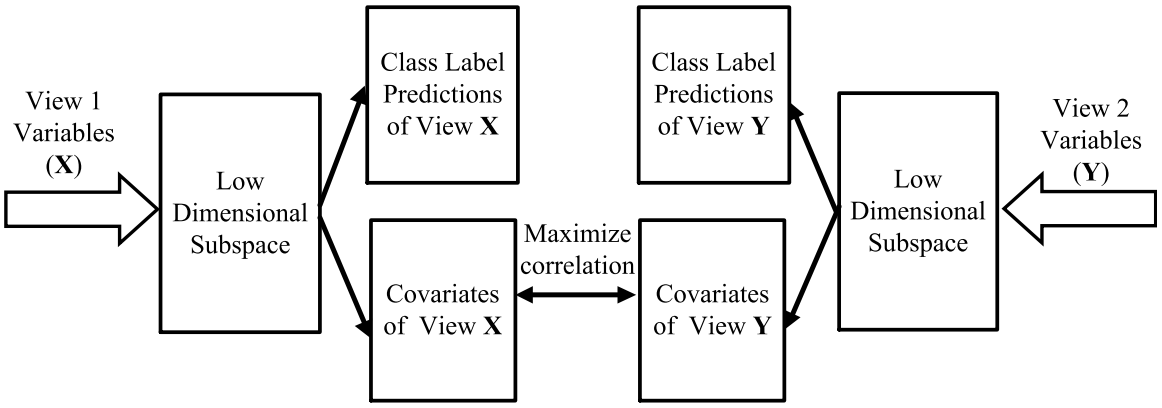


Figure 4.1. Block diagram of the proposed D-AR method. Dimensionality of $X \in \mathbb{R}^{\mathbf{m}\times\mathbf{N}}$ and $Y \in \mathbb{R}^{\mathbf{n}\times\mathbf{N}}$ are reduced from m and n to k, respectively, where $m > p$ and $n > p$. Correlated outputs (covariates) are alternated between the two views to maximize their correlations during training.

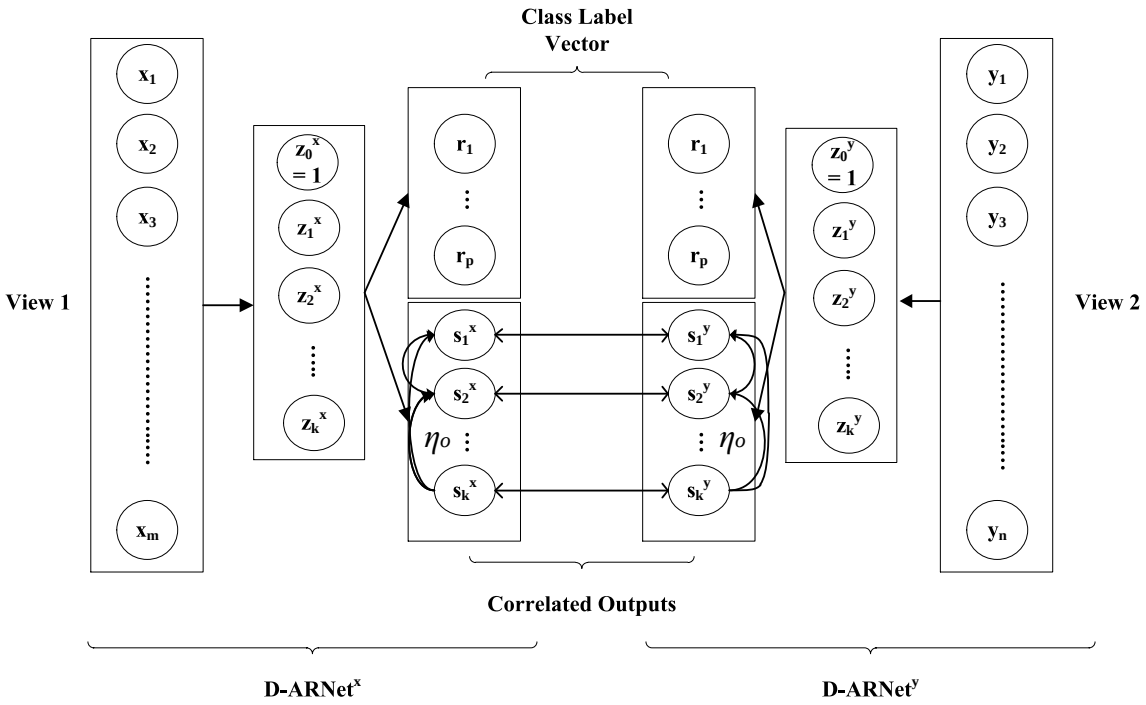


Figure 4.2. Architecture of the proposed D-AR method.

Input

k: number of outputs

p: number of classes

N: number of instances

X: $\mathbf{N} \times \mathbf{m}$ training set of view 1

Y: $\mathbf{N} \times \mathbf{n}$ training set of view 2

$\eta_{\mathbf{0}}$: inhibition coefficient of output layer

Output

$\mathbf{w^x}$: input layer weights of view **X**

$\mathbf{v^x}$: hidden layer weights of view **X**

$\mathbf{s^x}$: correlated outputs of view **X**

$\mathbf{r^x}$: predicted class label vector of view **X**

$\mathbf{z^x}$: covariates of view **X** of size $\mathbf{N} \times \mathbf{k}$

d: D-ARNet correlations of view **X**

Randomly initialize $\mathbf{s^y}$

repeat

Train D-AR network of view **X**

$[\mathbf{w^x}, \mathbf{v^x}, \mathbf{d}] = \text{D-ARNet}^x\ (\mathbf{X}, \mathbf{k}, \mathbf{p}, \mathbf{s^y}, \eta_{\mathbf{0}})$

$\mathbf{z^x} = \mathbf{X} \times \mathbf{w^x}$

$[\mathbf{s^x}, \mathbf{r^x}] = \mathbf{z^x} \times \mathbf{v^x} + \mathbf{v_0^x}$

Scale $\mathbf{s^x}$: zero mean, unit standard deviation

Train D-AR network of view **Y**

$[\mathbf{w^y}, \mathbf{v^y}, \mathbf{d}] = \text{D-ARNet}^y\ (\mathbf{Y}, \mathbf{k}, \mathbf{p}, \mathbf{s^x}, \eta_{\mathbf{0}})$

$\mathbf{z^y} = \mathbf{Y} \times \mathbf{w^y}$

$[\mathbf{s^y}, \mathbf{r^y}] = \mathbf{z^y} \times \mathbf{v^y} + \mathbf{v_0^y}$

Scale $\mathbf{s^y}$: zero mean, unit standard deviation

until convergence

Figure 4.3. Algorithm of the proposed D-AR method.

dimensional input space. If the hidden layer is removed, the architecture reduces to a traditional neural CCA model which is augmented by an independent perceptron that performs classification. The hidden layer is forced to implement both LDA and CCA simultaneously. The unusual use of "linear" hidden units, rather than the traditional nonlinear ones such as sigmoidal ones, in the hidden layer is to preserve the linearity of the transformation while performing the dimensionality reduction task. It must also be noted that LDA, and consequently DCCA, have the limitation of extracting less than number of classes dimensions (orthogonal projective directions) due to the rank deficiency of the between-class scatter matrix [125] whereas the proposed D-AR method does not suffer from this limitation.

4.2.2. Alternating Regression Procedure of D-AR

The training signals used for the outputs of the two perceptrons are evolved through an alternating regression procedure. In the alternating regression procedure, we minimize the differences in output covariates computed by the two perceptrons in response to their coincident input vectors from view 1 and view 2. The procedure is iterative and decreases the difference at each step by alternately re-training one perceptron on the outputs of the other (while holding the other one fixed at its previous evolution) and vice versa [74, 78, 126].

The total error function of the system can be generalized and written as:

$$E^x(w^x, v^x|X) = E_s^x + \lambda E_r^x \tag{4.12}$$

where X is the set of instances of the view; E_s^x and E_r^x are the errors of correlated output units and class label units of the output layer, respectively; and, λ is the discrimination factor that can be used to tune the trade-off between the prediction of the class labels and correlation of the outputs in the proposed architecture. Clearly, when $\lambda = 0$, we have D-AR equal to neural implementation of CCA (Neural implementation of alternating regression). In this case, the class labels will not be incorporated into our framework and the D-AR architecture will be essentially equivalent to the case where

there is no hidden layer at all. With the increasing values of λ, the network will be forced to give more weight to the error term of the class labels, E_r^x, and at the extreme values of λ, each perceptron will behave like a simple perceptron that aims at learning the class labels by using only its own features without interacting with the other view (similar to LDA). The value of the discrimination factor λ can be optimized on the training set to obtain the desired balance between the discrimination accuracy and the correlations among the covariates.

The total error function of the system can be calculated as:

$$E^x(w^x, v^x|X) = \frac{1}{2}\Bigg(\sum_{t=1}^{N}\sum_{i=1}^{k}\left(s_{it}^y - s_{it}^x\right)^2\Bigg) - \lambda\sum_{t=1}^{N}\sum_{i=1}^{p}\left(l_{it}\log r_{it}^x\right) \tag{4.13}$$

where N is the number of instances, k is the number of features that will be extracted, p is the number of classes, w^x and v^x are the hidden and output layer weights of view X, l_{it} is 1 if sample x^t belongs to class i and 0 otherwise, r_{it}^x is the predicted value of ith class for sample t. The correlated output value of view X for sample t of ith output is denoted with s_{it}^x.

The update rules of the hidden layer weights, w^x, of the D-ARNet multi-layer perceptron architecture are extracted using the backpropagation algorithm [127]:

$$\frac{\partial E^x}{\partial w_{hj}^x} = \frac{\partial E_s^x}{\partial s_i^x}\frac{\partial s_i^x}{\partial z_h^x}\frac{\partial z_h^x}{\partial w_{hj}^x} + \lambda\frac{\partial E_r^x}{\partial r_i^x}\frac{\partial r_i^x}{\partial z_h^x}\frac{\partial z_h^x}{\partial w_{hj}^x} \tag{4.14}$$

where w_{hj}^x is the hidden layer weight between the jth input feature and h^{th} hidden unit of view X, and z_h^x is the h^{th} hidden unit of view X. The correlated output units and predicted class values are:

$$\begin{aligned} s_{it}^x &= \sum_{h=1}^{k} v_{ih}^x z_{ht}^x + v_{i0}^x \\ r_{it}^x &= \frac{\exp(v_{ih}^x z_{ht}^x + v_{i0}^x)}{\sum_{k=1}^{p}\exp(v_{ih}^x z_{ht}^x + v_{i0}^x)} \end{aligned} \tag{4.15}$$

where v_{ih}^x is the output layer weight between the h^{th} hidden and ith correlated output unit of view X. Using gradient descent, we get the following update rules for the hidden and output layer weights:

$$\Delta w_{hj}^x = \eta_1 \sum_{t=1}^{N} \left[\sum_{i=1}^{k} (s_{it}^y - s_{it}^x) v_{ih}^x \right] x_{jt} + \lambda \eta_2 \sum_{t=1}^{N} \left[\sum_{i=1}^{k} (l_{it} - r_{it}^x) v_{ih}^x \right] x_{jt} \tag{4.16}$$

$$\Delta v_{ih}^x = \eta_1 \sum_{t=1}^{N} (s_{it}^y - s_{it}^x) z_{ht}^x + \lambda \eta_2 \sum_{t=1}^{N} (l_{it} - r_{it}^x) z_{ht}^x \tag{4.17}$$

where η_1 and η_2 denote the learning factors of the covariates and class labels, respectively.

4.2.3. Decorrelation of Output Units

The alternated output units which provide the correlation between the hidden layer units of the views in D-AR algorithm must be decorrelated. Otherwise, they tend to tune to the same direction which gives the minimum mean-square error. In [128], Foldiak showed that a layer of simple Hebbian units connected by modifiable anti-Hebbian feed-back connections can learn to discriminate the patterns of different classes in such a way that statistical dependency between the elements of the representation is reduced, while information is preserved. He obtained a sparse resulting code, which is favourable if it is to be used as input to a subsequent supervised associative layer. He showed the usefulness of the network on two simple problems.

Thus, we use the cascading anti-Hebbian inhibition algorithm for decorrelating the outputs of each perceptron and force them tune to different, ideally orthogonal, covariates. The cascading anti-Hebbian inhibition rule to decorrelate output s_i^x with the already produced outputs is applied as follows:

$$s_{it}^x = s_{it}^x - \sum_{j=1}^{i-1} \eta_0 \times \rho(s_i^x, s_j^x) \times s_{jt}^x, \forall t \in X \tag{4.18}$$

where η_0 is the inhibition coefficient of outputs, and $\rho(s_i^x, s_j^x)$ is the correlation coeffi-

cient between ith and jth outputs, s_i^x and s_j^x, of view X.

5. EXPERIMENTS AND RESULTS

In this chapter, experimental results on various datasets are given in order to show the superiority of our proposed methods over the alternative methods.

5.1. Ensemble Canonical Correlation Analysis Experiments

5.1.1. Methodology

In our ECCA experiments, we use bagging (ECCA-B), jack-knife (ECCA-J), and partitioning (ECCA-P) ensemble construction methods for creating sufficient and diverse subsamples. Unless otherwise specified, the methodology used in the evaluation of ECCA experiments is as follows: The test set correlations of the variations of ECCA are evaluated and compared against traditional CCA under various training set sizes and number of subsamples in the ensemble. Besides, we present the box-plot figures for statistical evaluation using a representative setting for each one of the ensemble construction method: the number of subsamples for ECCA-B are taken as 10, the number of folds for ECCA-J and the number of partitions for ECCA-P are set to 4. We provide the box-plot figures for training sets with 100 and 200 samples which are chosen randomly from the whole data, and the rest of the data is used as the test set. The train-test splits are repeated 10 times for statistical significance.

We present the experimental results on emotion recognition, object recognition, content-based retrieval, and multiple view object recognition datasets. We also evaluate the robustness of ECCA covariates on a toy dataset.

Although canonical correlation based features are extracted without any supervision for class discrimination, as they are known to tune to prominent features that are also useful for classification, we compare the discriminative power of the covariates extracted by CCA and the proposed ECCA methods on the emotion recognition, handwritten digit recognition, and content-based retrieval experimental datasets. We

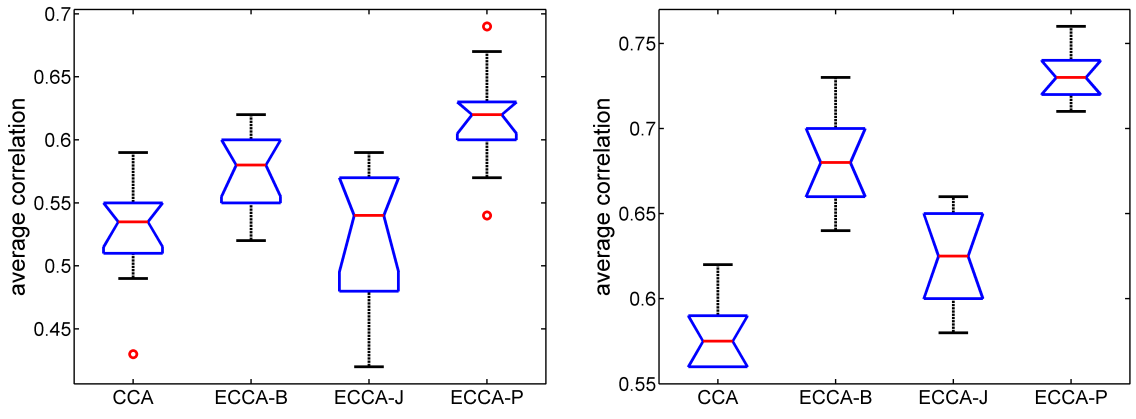
Figure 5.1. An exemplary video from CK+ dataset [129–131] (a) Acting begins with neutral expression (b) An example frame between neutral and peak expression (c) Happiness target expression.

use Support Vector Machines (SVMs) as the classifier to evaluate the discriminative performances of the extracted covariates.

5.1.2. Emotion Recognition

The Cohn-Kanade Facial Expression Database (CK+) [129] consists of 320 video clips each along with an emotion label recorded from 118 subjects. The video clips in CK+ dataset belong to 7 different emotions which are anger, contempt, disgust, fear, happiness, sadness, and surprise. Each video clip begins with a neutral expression and proceeds to a peak expression where the emotion is most significant (see Figure 5.1).

Two types of feature representations extracted from CK+ dataset are used as views in our experimental studies. The first view consists of appearance-based features [132, 133] which is obtained using the difference image between the first frame of the video clip (the neutral facial expression) and the corresponding last frame (the peak frame of the emotion). Each sample in the appearance based-view has 4096 (64×64) features (pixels). The second view consists of the geometric features [130, 131] such as the positions of the specific landmark points on the face and the shape of the components of the face. They are obtained by subtracting the coordinates of landmark points of the neutral face expression from the coordinates of the landmark points of the target expression. Each sample is represented with 134 features in the geometric-based view.

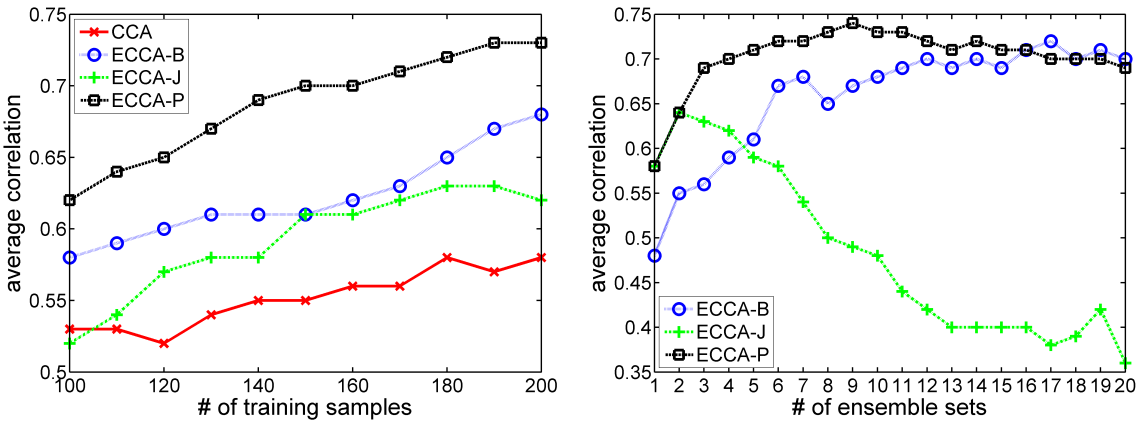


(a) (b)

Figure 5.2. Average correlation of top 5 covariates extracted between appearance and geometric based views of face dataset using (a) 100 training samples (b) 200 training samples.

The average correlation of the top 5 covariates extracted between appearance and geometric based views of CK+ dataset using 100 and 200 training samples are shown in Figure 5.2. The central mark of each box in Figure 5.2 represents the median and the box edges represent the 25th and the 75th percentiles. The whiskers extend to the most extreme data points which are not considered as outliers. Outliers are visualized by circles. As seen in Figure 5.2, with both 100 and 200 training samples, the notches of ECCA-P do not overlap with the notches of any other method which shows that at the 5% significance level, the true median of ECCA-P is significantly higher than CCA and also the other ECCA methods. ECCA-B gives significantly higher correlations than CCA for both 100 and 200 training samples, whereas ECCA-J is significantly better than CCA only for 200 training samples. As the subsamples of ECCA-J have more overlap with the original training set than those of the other ECCA variations, ECCA-J shows the closest performance to CCA.

In Figure 5.3a, average correlation of the top 5 covariates are shown with increasing number of training samples. As it is seen, the correlations show an increasing trend for all the methods using more training samples. However, increasing the number of training samples improves the generalization of the ECCA methods more than that of CCA on the test set.



(a) (b)

Figure 5.3. Average correlation of top 5 covariates extracted between appearance and geometric based views of CK+ face dataset versus number of (a) training samples (b) subsamples.

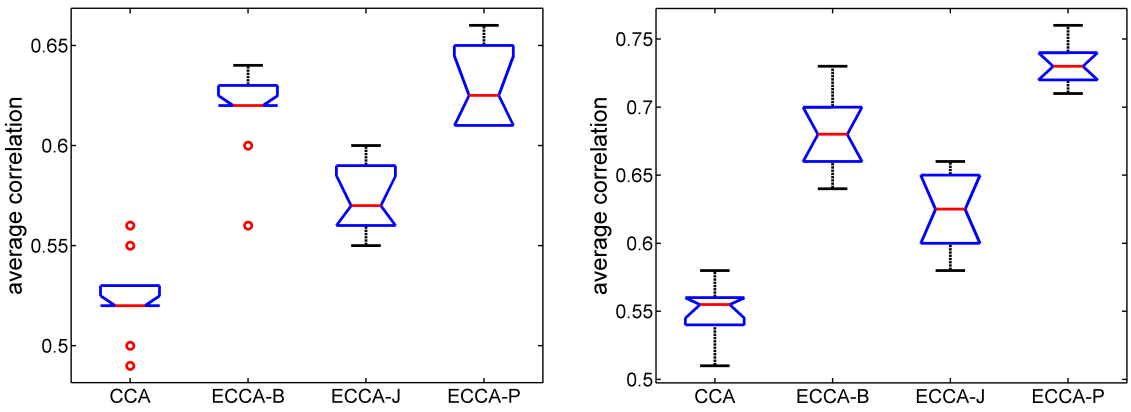
Figure 5.3b shows the number of ensemble sets (subsamples) versus average correlation of the top 5 covariates using 200 training samples. It is seen that the correlations of ECCA-J are in a decreasing trend with the increasing number of subsamples, whereas correlations of ECCA-B and ECCA-P increase. This reveals that the performance of ECCA-J is getting closer to CCA with more overlapping training samples in its subsamples and tends to tune dummy dependencies that gives very high correlations on the training set but does not hold on the test set. The highest correlation is obtained with ECCA-P by partitioning the training dataset into 9 subsets. However, the correlation has a decreasing trend after 9 partitions because after this point the number of samples in the partitions becomes insufficient to learn covariates with good generalization on the test set. The performance of ECCA-B increases until 12 ensemble sets; after this point, a stable performance is observed, which is because most of the samples are already chosen in subsamples, and so neither accuracy nor diversity of the subsamples improves.

5.1.3. Handwritten Digit Recognition

The handwritten digit recognition dataset is available in the UCI machine learning data repository [47]. The dataset consists of features of handwritten numerals ('0'–'9') extracted from a collection of Dutch utility maps. There are 200 patterns per

class (for a total of 2,000 patterns). In the dataset, there are six views. The CCA experiments are performed on all view pairs. As we obtained similar results on all pairs, as representative examples, only the experiments on the Fourier (76 Fourier coefficients of the character shapes)-Profile (216 profile correlations) and Profile-Karhunen coefficients (64 Karhunen-Love coefficients) view pairs are elaborated.

As seen in Figure 5.4, all of the ECCA variations give significantly higher correlations than CCA. For 100 training samples, ECCA-B and ECCA-P show similar performances. However, in 2 out of 10 runs, ECCA-B produces outliers, showing that ECCA-P is more stable than ECCA-B when the training sample size is comparably small. When the number of training samples is increased to 200, ECCA-P gives significantly higher correlations than CCA and the other ECCA variations.

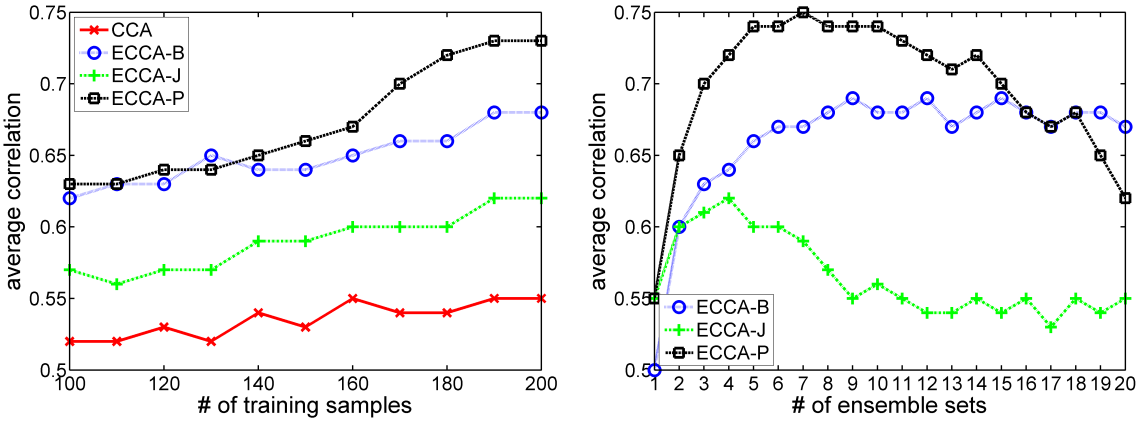


(a) (b)

Figure 5.4. Average correlation of top 5 covariates extracted between Fourier coefficients and profile correlations views of handwritten digit dataset using (a) 100 training samples (b) 200 training samples.

It can be seen in Figure 5.5a that the correlations of ECCA-B and ECCA-P are close to each other up to 140 training samples; after this point, using more training samples, ECCA-P outperforms ECCA-B because the number of samples in each partition of the ECCA-P method becomes increasingly more sufficient (better generalization) with more training samples. In Figure 5.5b, it is clearly seen that the correlations obtained with ECCA-J method increases up to 4 subsamples, then worsens with the increasing number of subsamples and gives almost the same correlations with CCA after 12 subsamples. This shows that the performance of ECCA-J is getting closer to

CCA with more subsamples. It is also observed in Figure 5.5b that the performance of ECCA-B increases until 10 subsamples, and then again, as it is the case for the emotion recognition dataset, a stable performance is observed. The highest correlation is obtained with ECCA-P by partitioning the training set into 7 subsets, and partitioning the dataset into more subsamples with insufficient number of instances results in obtaining covariates with lower correlations.

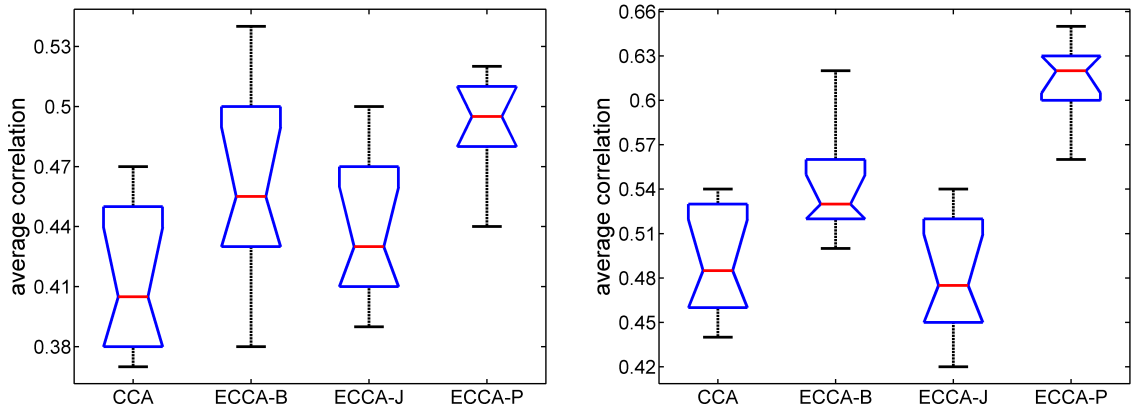


(a) (b)

Figure 5.5. Average correlation of top 5 covariates extracted between Fourier coefficients and profile correlations views of handwritten digit dataset versus the number of (a) training samples (b) subsamples.

Experiments between Profile correlations and Karhunen-Love coefficients views of handwritten digit recognition dataset give similar results to those obtained between Fourier coefficients and Profile correlations views. As it is seen in Figure 5.6, ECCA-B and ECCA-P methods give significantly higher correlations with 200 training samples than CCA whereas only ECCA-P is significantly better than CCA with 100 training samples.

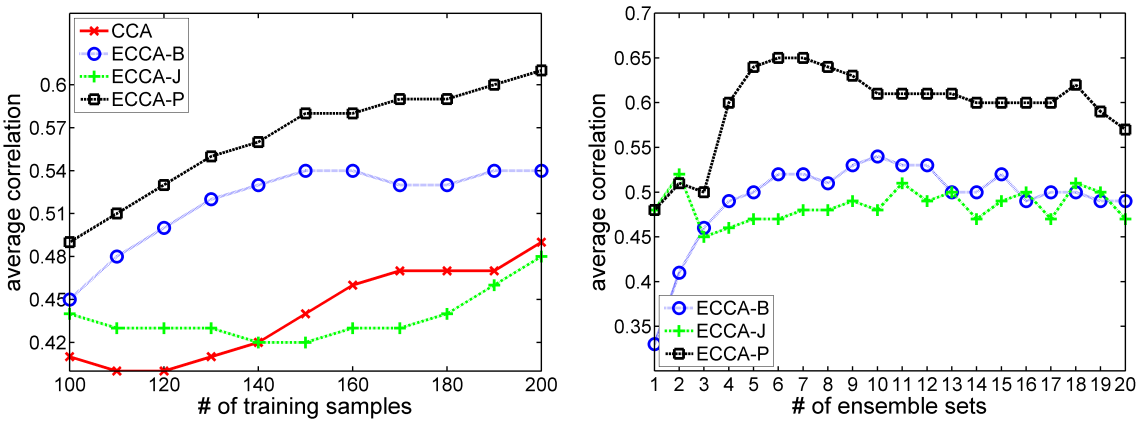
Figure 5.7 shows the comparative performance of ECCA variations on profile correlations - Karhunen-Love coefficients view pair with various training set sizes and number of subsamples. As it can be seen, ECCA-P gave the highest average correlations for all number of training samples from 100 to 200. Figure 5.7 also shows that the performance of ECCA-B increases until 10 ensemble sets, and then a stable performance is observed. The optimal number of partitions is 7 for ECCA-P. The average correlation shows a decreasing trend after 7 partitions because the number of samples in the



(a) (b)

Figure 5.6. Average correlation of top 5 covariates extracted between profile correlations and Karhunen-Love coefficients views of handwritten digit dataset using (a) 100 training samples (b) 200 training samples.

partitions are not enough to learn generalizable covariates.

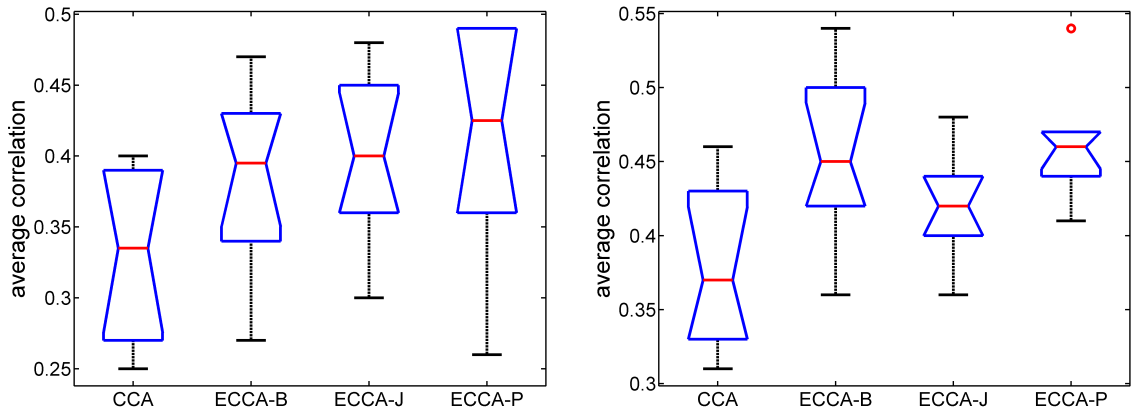


(a) (b)

Figure 5.7. Average correlation of top 5 covariates extracted between profile correlations and Karhunen-Love coefficients views of handwritten digit dataset versus the number of (a) training samples (b) subsamples.

5.1.4. Video Retrieval Evaluation

The TREC Video Retrieval Evaluation (TRECVID) 2003 [134] dataset consists of 1078 manually labeled video shots that belong to 5 categories. The TRECVID dataset is used with the purposes of content-based analysis and retrieval from digital video. Each shot of the dataset is represented with two views: 1894-dimensional binary vector



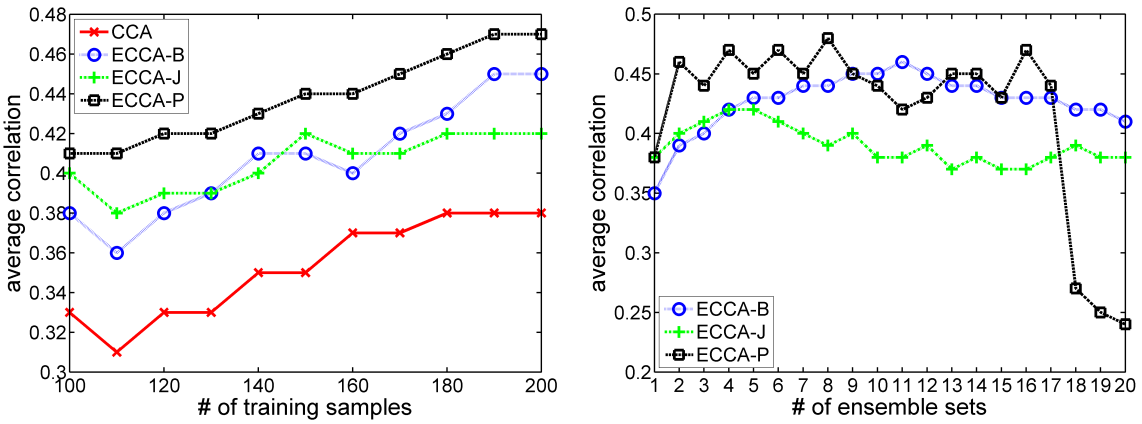
(a) (b)

Figure 5.8. Average correlation of top 5 covariates extracted between text and color histogram based views of TRECVID 2003 dataset using (a) 100 training samples (b) 200 training samples.

of text features and a 165-dimensional vector of HSV color histogram.

The correlations of CCA and the ECCA methods with 100 and 200 training samples are shown in Figure 5.8. As it can be seen, ECCA-P and ECCA-J produce significantly higher correlations than CCA for both 100 and 200 training samples, whereas ECCA-B is significantly better than CCA only for 200 training samples. It must also be noted the correlations obtained on TRECVID dataset is comparably lower than those on emotion and handwritten datasets. One of the reasons for these low correlations is the high dimensionality of the views of TRECVID dataset. Although we have applied PCA as a preprocessing step to reduce the dimension of the first view by preserving the 98% of the variance, further improvements may be obtained using a more sophisticated dimensionality reduction technique.

As seen in Figure 5.9a, ECCA-P is again the most successful method on TRECVID 2003 dataset in extracting correlated features. While ECCA-B and ECCA-J perform similarly with various number of training samples, both of them are superior over CCA in terms of average correlation. Figure 5.9b points out that the performance of ECCA-P variation increases up to 8 partitions, whereas it decreases after this point since the partitions fall short of sufficient number of training samples to learn generalizable covariates. The optimal numbers of ensemble sets for ECCA-J and ECCA-B are observed



(a) (b)

Figure 5.9. Average correlation of top 5 covariates extracted between text and color histogram based views of TRECVID 2003 dataset versus the number of (a) training samples (b) subsamples.

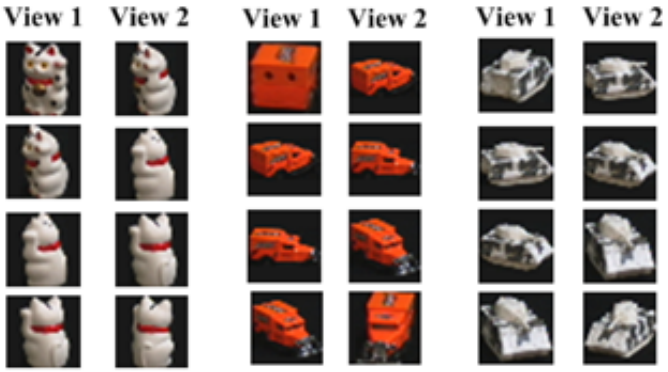
as 4 and 10, respectively, on TRECVID 2003 dataset. Also, the results obtained on TRECVID dataset with various number of subsamples validate the results obtained on emotion and handwritten datasets.

5.1.5. Object Recognition

The Columbia Object Image Library (COIL-100) [135] dataset consists of images of natural 100 objects taken under different viewing angles (see Figure 5.10). The objects were placed on a motorized turntable against a black background and the turntable was rotated through 360 degrees to vary object pose with respect to a fixed color camera. The images of the objects were taken at pose intervals of 5 degrees, so each object has 72 poses.

We have generated a two-view dataset by taking poses separated by 45 degrees. We pick a random starting point in the interval from 0 to 360 degrees and generate 8 pairs of poses for the training set. For each training pose, there are two poses separated by 5 degrees (to the left and right), we use those as the starting points and generate 16 test examples (again pairing them up with poses that are 45 degrees apart). This test set is called test set 1. We have repeated the same procedure for poses separated from the training examples by 10 degrees for generating test set 2 (a slightly harder

Figure 5.10. Exemplary objects from the COIL-100 object dataset.



(a) (b) (c)

Figure 5.11. Exemplary training samples of multi-view COIL-100 dataset.

test set). For test set 3 and 4, we have used 15 and 20 degree separation, respectively; thus, exhausting all COIL images into a training set of 800 samples and for test sets, each one having 1600 samples. The features are the pixel values of 32×32 image of the corresponding object. Some training set samples of View 1 and 2 are shown in Figure 5.11.

The canonical vectors extracted using the whole training set of COIL-100 dataset are applied on each test set. The correlations of the top 5 covariates can be seen in Table 5.1. It is seen that the correlations of CCA covariates on test set 1 are comparable with the ensemble methods. However, the canonical vectors of CCA explored on the training set do not generalize well on the other test sets when compared with the ECCA variations. This shows that CCA overfits on the training set and gives high correlations on test set 1 that consists of only 5 degrees rotated poses of training samples. However, these training correlations do not hold so well on the other three harder test sets which consists of 10, 15, and 20 degrees rotated poses of training samples. On the other hand, ECCA methods give more stable correlations. While ECCA-B is the most successful

method for test sets 1 and 2, ECCA-P yields higher correlations on test sets 3 and 4. Besides, ECCA-P gives the most consistent correlations on the test sets which reveals that the canonical vectors explored by ECCA-P generalize better on the test sets.

Table 5.1. Test set correlations of top 5 covariates extracted between views of COIL-100 object dataset.

	Test set 1				**Test set 2**			
ρ	**CCA**	**ECCA-B**	**ECCA-J**	**ECCA-P**	**CCA**	**ECCA-B**	**ECCA-J**	**ECCA-P**
ρ_1	0.93	0.95	0.95	0.95	0.90	0.92	0.91	0.95
ρ_2	0.94	0.96	0.96	0.93	0.87	0.91	0.92	0.93
ρ_3	0.93	0.95	0.94	0.92	0.87	0.92	0.90	0.91
ρ_4	0.92	0.93	0.93	0.89	0.85	0.89	0.88	0.89
ρ_5	0.91	0.93	0.92	0.84	0.81	0.86	0.85	0.80
avg	0.93	0.94	0.94	0.91	0.86	0.90	0.89	0.89
	Test set 3				**Test set 4**			
ρ	**CCA**	**ECCA-B**	**ECCA-J**	**ECCA-P**	**CCA**	**ECCA-B**	**ECCA-J**	**ECCA-P**
ρ_1	0.85	0.89	0.87	0.94	0.83	0.87	0.84	0.93
ρ_2	0.83	0.89	0.90	0.92	0.81	0.87	0.90	0.91
ρ_3	0.85	0.91	0.88	0.89	0.82	0.89	0.86	0.90
ρ_4	0.82	0.88	0.86	0.88	0.81	0.88	0.86	0.87
ρ_5	0.79	0.84	0.82	0.80	0.78	0.84	0.82	0.82
avg	0.83	0.88	0.87	0.89	0.81	0.87	0.85	0.89

As it is seen in Figure 5.12, on test set 1 the correlations of ECCA-P reveal an increasing trend with the increase in number of subsamples. However, on the other test sets the trend is decreasing, which reveals that the small partitions are not sufficient to learn canonical vectors that can generalize. In other words, the individual sets of covariates extracted on each subsample overfit the training set when the training sample size is small. It can also be observed in Table 5.1 that ECCA-J method shows the most similar performance to CCA as it is also observed for the other experimental datasets. ECCA-B is again the most robust method to the number of subsamples.

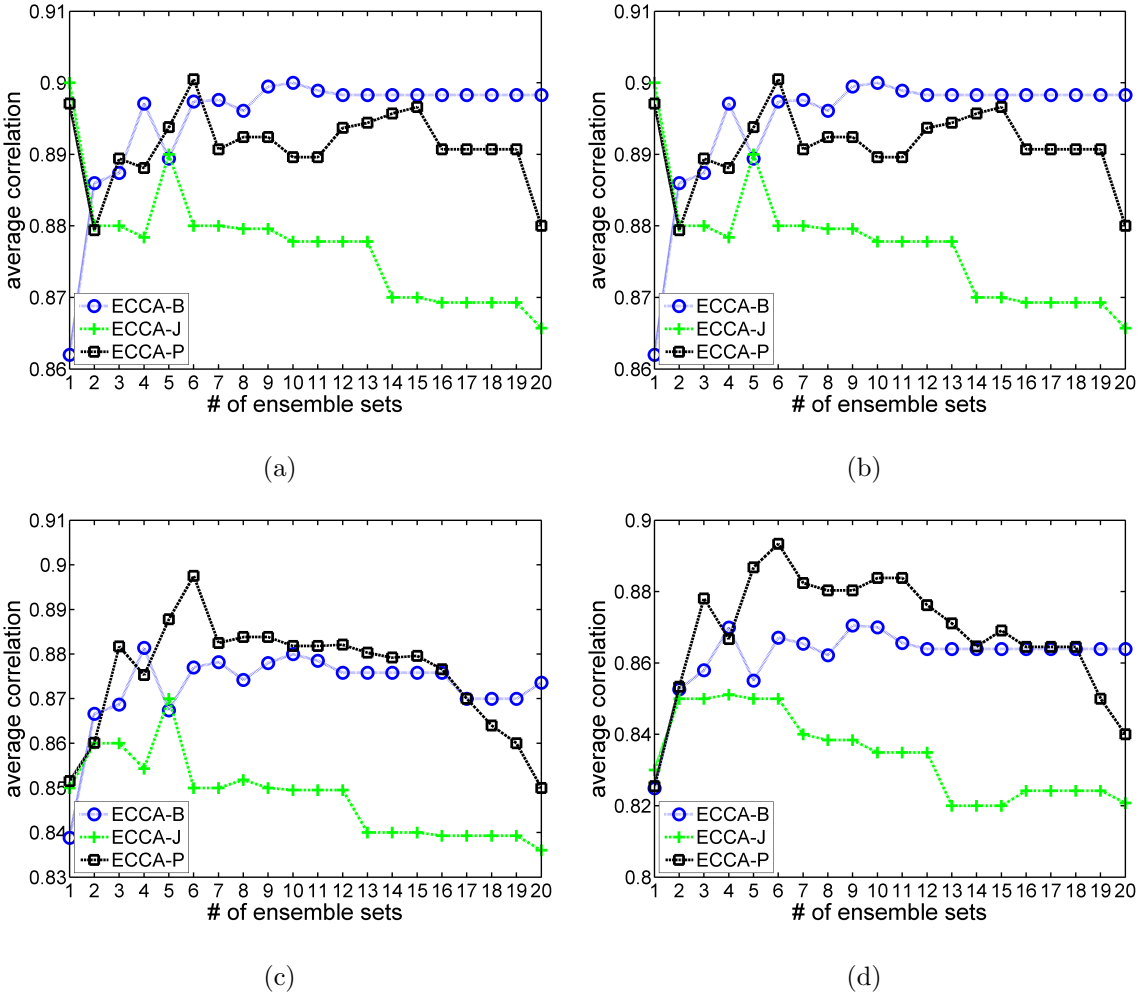


Figure 5.12. Number of ensemble sets versus average correlation of top 5 covariates extracted between rotated views of COIL object dataset. (a) test set 1 (b) test set 2 (c) test set 3 (d) test set 4.

5.1.6. Toy Dataset

The toy dataset consists of two views, each of which has two features. The common information of the views is the radius of a circle. In order to generate a sample of the two views, first, the common radius value, r, is randomly generated in unit interval. Then, for each view, a random angle value α in the interval of $[0, 2\pi]$ is generated, and the features of the view are computed as $(r\cos\alpha)^2$ and $(r\sin\alpha)^2$. The feature values of the samples of the views are shown in Figure 5.13. The algorithm used to generate the toy dataset is summarized in Figure 5.14.

In realistic settings, some of the samples in each view are corrupted due to an

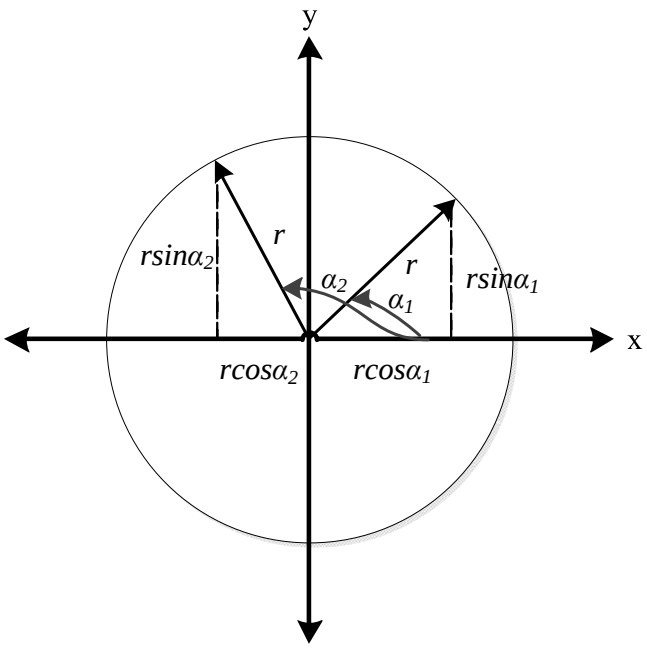


Figure 5.13. Feature values of the samples of the toy dataset. $r\cos\alpha_1(x_1)$ and $r\sin\alpha_1(x_2)$ are features of view 1(X) whereas $r\cos\alpha_2(y_1)$ and $r\sin\alpha_2(y_2)$ are features view 2 (Y).

independent noise process or view corruption [56]. This is because one of the multiple sensors used in collection of multi-view data can be temporarily in an erroneous condition or may have temporarily calibration problems. Due to the sensitivity of covariance matrix, CCA is effected by such noisy observations and can tune to dependencies with poor generalization on the test set. With the aim of comparing the generalization capability of the CCA and ECCA methods in a similar realistic setting, after generating 100 training samples, a few samples of each view are corrupted. Then, the CCA and ECCA methods are used to extract the covariate pair between the views of the toy dataset. The test set with 1000 clean samples is generated using the same procedure, and the test set correlations of the covariates are obtained for each method. This process is repeated for 100 times for statistical significance and the change in average correlation of the CCA and ECCA covariates with increasing number of corrupted samples is shown in Figure 5.15. It can be seen that ECCA-P and ECCA-J methods are more robust than CCA against outliers even on such a small dataset. ECCA-B cannot

Input

N: number of samples

Output

X: view 1 dataset of size $\mathbf{N} \times \mathbf{2}$

Y: view 2 dataset of size $\mathbf{N} \times \mathbf{2}$

for $i = 1$ to N **do**

Randomly generate radius (r) value in unit interval

Generate view 1 features:

Randomly generate angle value (α_1) in interval $[0, 2\pi]$

$X(i,1) = (r\cos\alpha_1)^2$

$X(i,2) = (r\sin\alpha_1)^2$

Generate view 2 features:

Randomly generate angle value (α_2) in interval $[0, 2\pi]$

$Y(i,1) = (r\cos\alpha_2)^2$

$Y(i,2) = (r\sin\alpha_2)^2$

end for

Figure 5.14. Toy dataset generation algorithm.

significantly outperform CCA because the subsamples generated with bootstrapping ensemble construction method are not sufficiently diverse on this small dataset. It is also observed that ECCA-P is the most robust method against number of corrupted samples on the toy dataset.

5.1.7. Usefulness of Covariates for Classification

CCA based features are extracted without any supervision for class discrimination, but as they are known to tune to prominent features that are also useful for

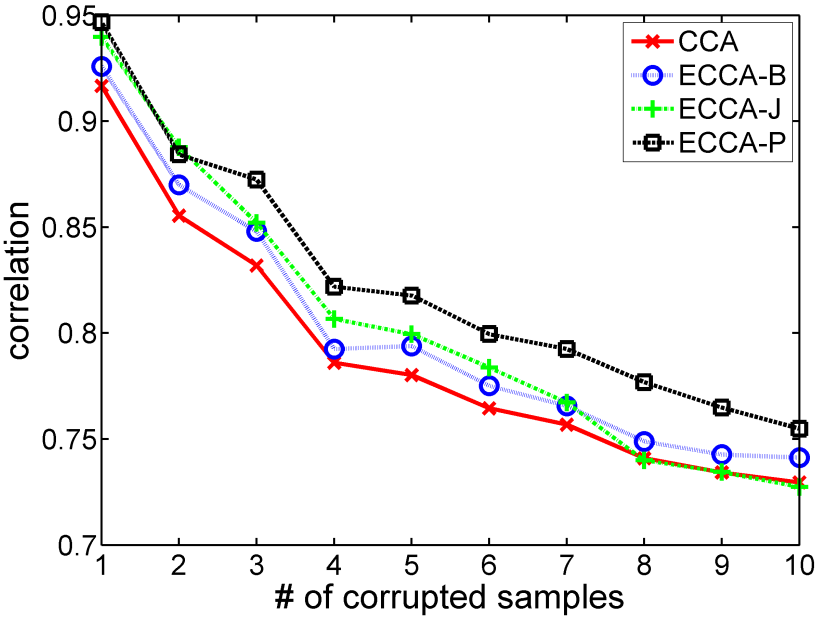


Figure 5.15. Correlation of the covariate extracted between the views of toy dataset versus number of corrupted samples.

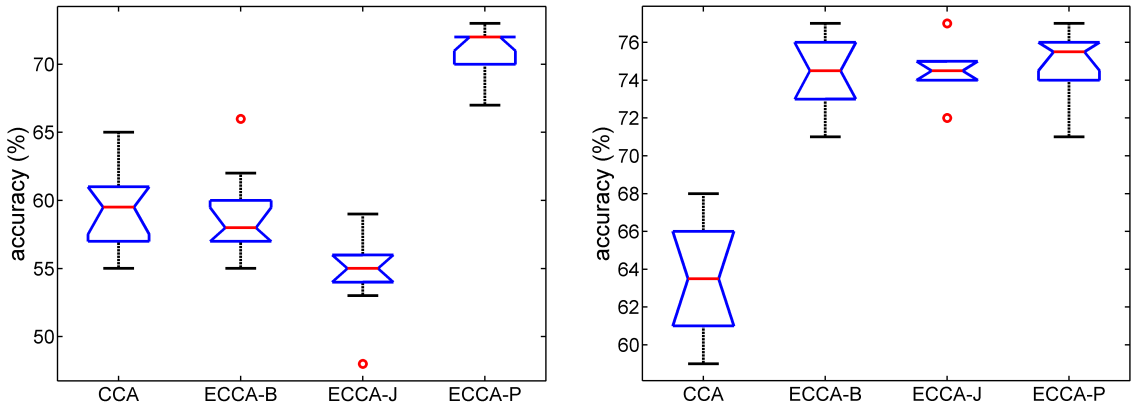
classification, we evaluate the discriminative power of CCA and ECCA covariates on the emotion recognition, handwritten digit recognition, and content-based retrieval experimental datasets. As the classifier, we use Support Vector Machines (SVMs) which is a very popular machine learning algorithm [136]. SVMs aims to find the optimally placed hyperplanes to discriminate the classes from each other. The closest samples to these hyperplanes are called support vectors, and the solution is defined in terms of this subset of samples which limits the complexity of the problem. We use the linear kernel LIBSVM [137] implementation of SVMs which supports multiclass classification. The complexity of the linear kernel SVMs is controlled by the Cost (C) parameter. Higher values of C may result in overfitting to the training set. In order to avoid overfitting, we used the default value of LIBSVM ($C = 1$) in our experiments.

As seen in Figure 5.16, for 200 training samples, view 1 covariates of ECCA-P have best classification performance whereas view 2 covariates of all the ECCA variations have significantly better classification performance than those of CCA. The view 2 covariates of ECCA variations achieved similar accuracies. However, it must be

noted that the difference between the lower and upper quartiles of ECCA-J is less than the other ECCA variations, showing that the covariates of ECCA-J method give more stable accuracies with different training examples than those of the other methods. This originates from the fact that the diversity among the subsamples of the ECCA-J is less than the diversity of the other ECCA methods. Therefore, the SVMs accuracies achieved by the individual set of covariates of ECCA-J deviate less than those of the other ECCA variations. It is also seen in Figure 5.16 that the difference between the lower and upper quartiles of the CCA accuracies is more than those of the ECCA variations. This reveals that the classification performances of the CCA covariates extracted from different training sets have greater variability when compared with those of the ECCA variations.

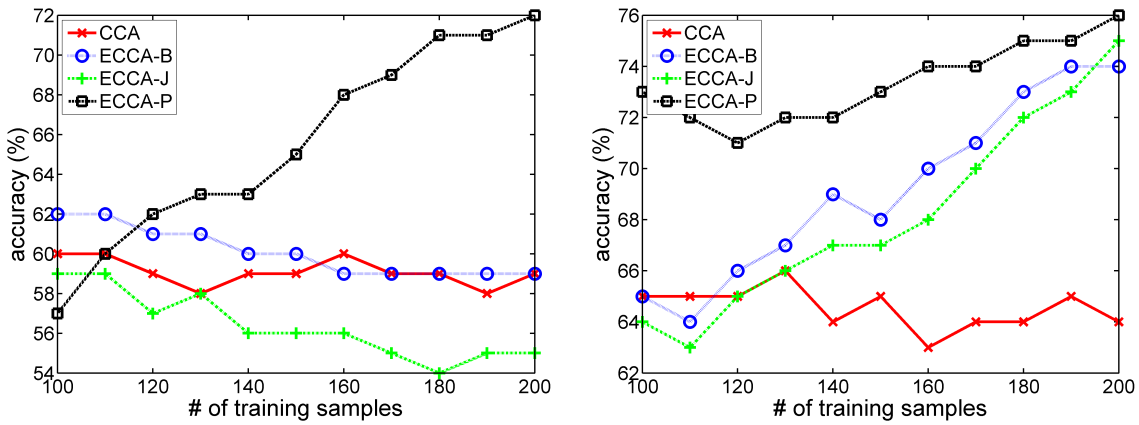
Figure 5.17 presents the trend of SVMs accuracy on emotion recognition dataset with the increasing number of training samples. It can be seen that the discriminative power of both view 1 and view 2 covariates of ECCA-P rise using more training samples. However, for ECCA-J and ECCA-B, while view 1 covariates of these methods tend to have less discriminative power, view 2 covariates of these methods perform higher accuracies. It is also seen that increasing the number of training examples do not significantly enhance the performance of CCA, which is because CCA tends to learn very small relations with high correlation between the views and such relations do not have generalizable discriminative information.

Figure 5.18 shows SVM accuracies on the test set obtained using the top 5 covariates of the handwritten digit dataset. To sum up, ECCA-P achieved the highest classification accuracy with both of the view covariates. While for Fourier coefficients view ECCA-B and ECC-J methods are not significantly better than CCA, for profile correlations view both of them outperform CCA. Again, similar to the results obtained on emotion recognition dataset, the most stable performance is shown by ECCA-J since its subsamples are less diverse when compared with those of the other ECCA methods. In general, it can be said that the classification performances of the ECCA methods are equal or significantly better than CCA. Figure 5.19 shows the SVM accuracies on the test set as a function of the number of training samples. The SVM



(a) (b)

Figure 5.16. SVM accuracy obtained using top 5 covariates extracted between appearance and geometric based views of CK+ dataset using 200 training samples. (a) appearance-based (b) geometric based.

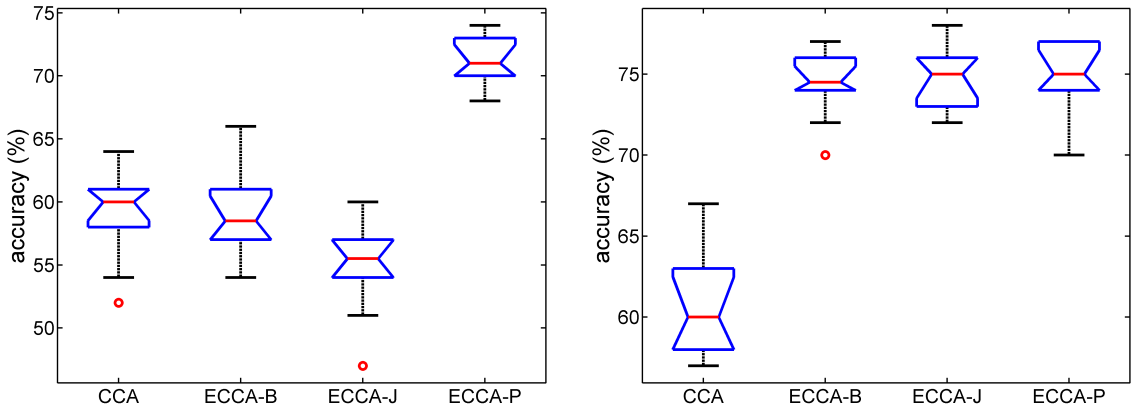


(a) (b)

Figure 5.17. Number of training samples versus SVM accuracy obtained using top 5 covariates extracted between appearance and geometric based views of CK+ dataset. (a) appearance-based (b) geometric based.

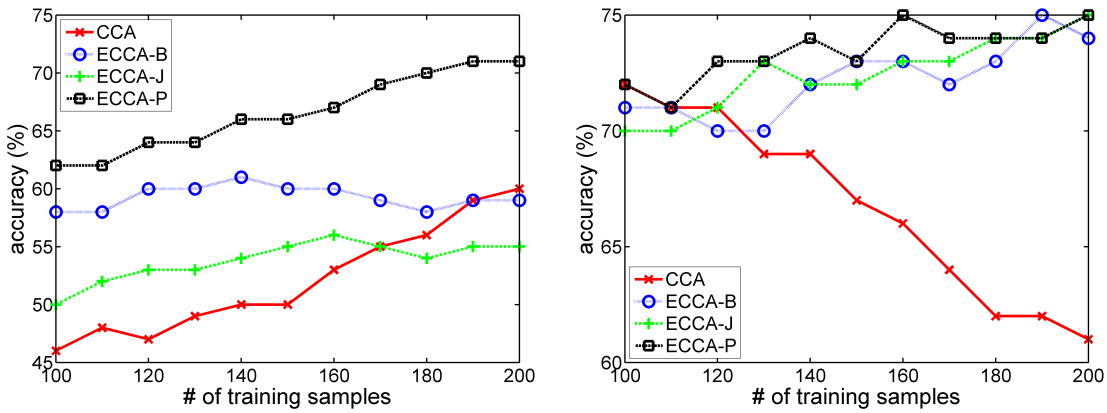
model of ECCA-P achieves the best classification performance with the covariates of the Fourier view. On the other hand, with profile correlations, the ECCA variations yield similar performances. The SVMs accuracies of ECCA variations are increasing or show a stable performance for both of the views. However, while the accuracies obtained with the view 1 CCA covariates are increasing, the accuracies of view 2 CCA covariates are in a decreasing trend. This is because the CCA correlations between view 1 and view 2 covariates are increasing with more training examples, and the classification performances of the highly correlated view 1 and view 2 covariates converge

to each other.



(a) (b)

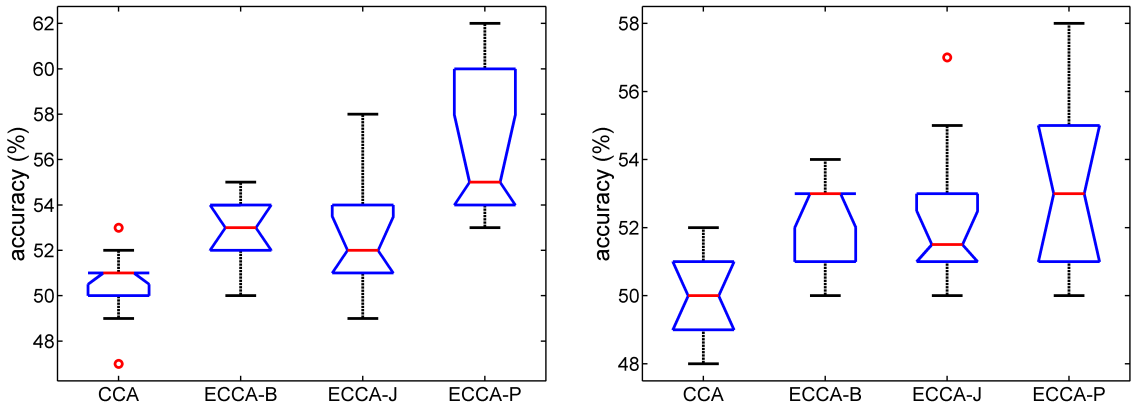
Figure 5.18. SVM accuracy obtained using top 5 covariates extracted between Fourier coefficients and profile correlations views of handwritten digit dataset using 200 training samples. (a) Fourier coefficients (b) profile correlations.



(a) (b)

Figure 5.19. Number of training samples versus SVM accuracy obtained using top 5 covariates extracted between Fourier coefficients and profile correlations views of handwritten digit dataset (a) Fourier coefficients (b) profile correlations.

Figure 5.20 shows the SVM accuracies obtained on TRECVID dataset. The results in Figure 5.20 reveal that view 1 covariates of all ECCA methods extracted using 200 training samples have more discriminative power than CCA. However, it must also be noted that only the superiority of ECCA-B and ECCA-P accuracies are significant. For view 2 covariates, it is seen that all the ECCA methods have significantly more discriminative power than CCA.



(a) (b)

Figure 5.20. SVM accuracy obtained using top 5 covariates extracted between text and color histogram based views of TRECVID dataset using 200 training samples. (a) text-based (b) histogram-based.

5.1.8. Discussion

We observe that using resampling we can generate a number of CCA trials, when combined resulting in covariates with higher correlations on the test set. As a representative example, the CCA trials on the individual subsamples of the emotion dataset (see Section 5.1.2) produce average test set correlation of around 0.4; when combined by the proposed method we obtain the test correlation of 0.7. Similar to the combination of weak classifiers to obtain an ensemble with better classification accuracy and generalization, the proposed ensemble CCA also takes advantage of weak correlations obtained from subsamples.

In order to show that weak correlations obtained on the subsamples contribute to the production of higher ECCA correlations, principal component analysis (PCA) [50] can be applied on the subsamples instead of CCA. However, combining the principal components in a similar fashion to the proposed combination approach (Equation 3.15) does not yield correlations as high as those obtained with ECCA. Again as a representative example, on the emotion dataset, using the PCA as the first step gives 0.6 average correlation whereas the ECCA covariates (using CCA as the first step) have significantly higher test set correlations of around 0.7. Note that even the use of PCA prior to the final combination step that utilize CCA outperforms the results obtained

by the classical CCA applied on the whole dataset. This result is promising in that the proposed approach may be giving us a framework that can be used to obtain ensemble dimensionality reduction; i.e. if PCA is used to combine PCA projections it may give us a robust set of dimensions with high-variance dimensions. Another future direction is to apply the proposed ensemble CCA approach to the other implementations of CCA, for example, the neural implementation of CCA is known to improve the generalization and its ensemble version can further improve its generalization.

5.2. Discriminative Alternating Regression Experiments

5.2.1. Methodology

In our Discriminative Alternating Regression (D-AR) experiments, the discriminative power of the covariates extracted with the proposed D-AR method are evaluated and compared against of the traditional CCA, PCA+CCA, AR, and LDA methods under various training set sizes. In the application of the proposed D-AR algorithm, we use 0.05 as the learning factor for the weight updates. The number of epochs and the total number of alternation iterations are experimentally found to be optimal around 30 and 80, respectively. For the sake of simplicity, we have used the discrimination factor, λ, as 1. The coefficient of inhibition among the output units is set to 1.0. The obtained covariates are fed to linear kernel Support Vector Machine (SVM) and k-Nearest Neighbor (k-NN) classifiers.

5.2.2. Emotion Recognition

The Cohn-Kanade Facial Expression Database (CK+) [129] consists of 327 video clips each along with an emotion label recorded from 118 subjects. The video clips in CK+ dataset belong to 7 different emotions which are anger, contempt, disgust, fear, happiness, sadness, and surprise. The details of the dataset is given in Section 5.1.2.

For statistical significance, the dataset is shuffled 10 times, and different training and test sets are generated in each run. In Figure 5.21, the obtained average SVM and

k-NN (k=3) classifier test set accuracies are shown with respect to increasing number of training samples per class. As seen, both view 1 and view 2 covariates of D-AR give significantly higher accuracies with SVM as well as k-NN (paired t-test, $p < 0.05$). While LDA as a supervised dimensionality reduction has the closest performance to D-AR, the CCA, PCA+CCA, and AR methods showed no significant superiority over each other. The results also point out that both SVM and k-NN classifiers accomplished 85% classification accuracy with view 2 covariates of D-AR extracted with only 10 samples per class.

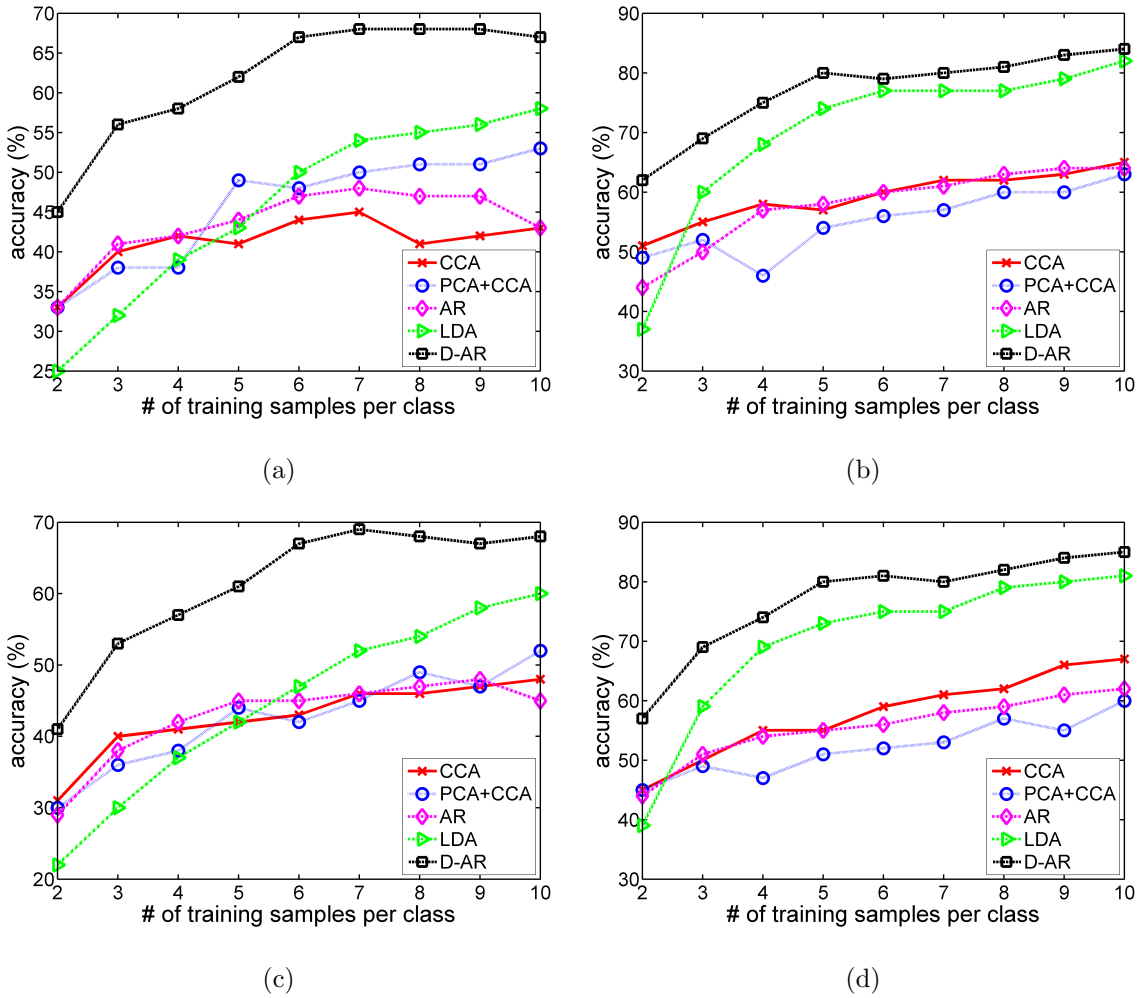


Figure 5.21. Number of training samples per class versus accuracies of covariates of (a) view 1 (SVM) (b) view 2 (SVM) (c) view 1 (3-NN) (d) view 2 (3-NN).

Figure 5.22 shows the classification accuracies obtained using 5 training samples per class versus the number of features extracted. As LDA can extract only $C-1$ features when applied to a dataset with C classes, in Figure 5.22, the classification accuracies are shown up to 6 covariates on the 7-class CK+ dataset. Again, D-AR

features give significantly higher accuracies than those of its alternatives for all number of outputs.

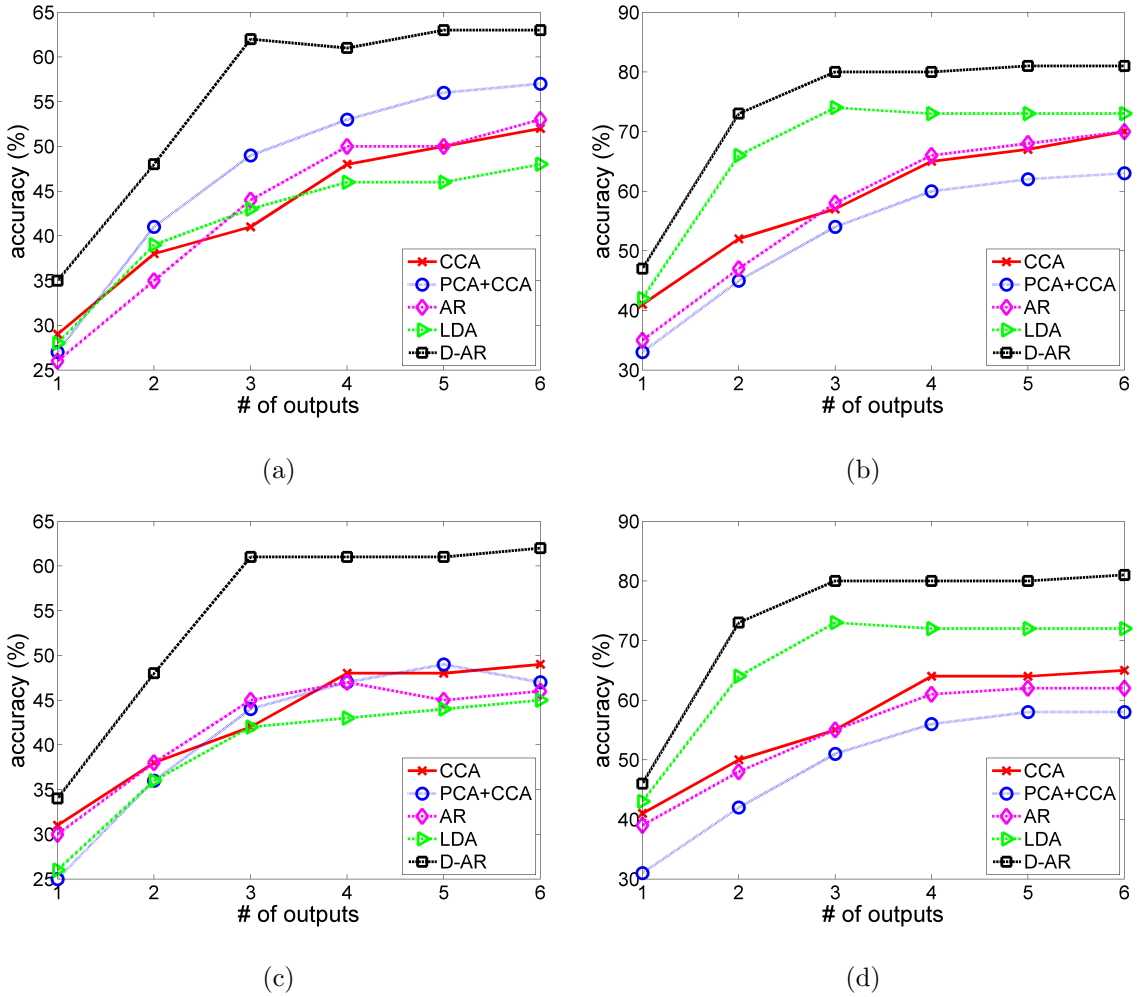


Figure 5.22. Number of outputs (covariates) versus accuracies obtained on emotion recognition dataset using 5 training samples per class with covariates of (a) view 1 (SVM) (b) view 2 (SVM) (c) view 1 (3-NN) (d) view 2 (3-NN).

In Figure 5.23, the relationship between discriminative power and correlation of the obtained outputs with D-AR method is shown with increasing value of discrimination factor, λ. In fact, when λ=0, we obtain the covariates of AR, in which case the class labels are not incorporated and the D-AR architecture is equivalent to the case where simple perceptrons (without the hidden layer) are alternated. Both the accuracy and the correlation of the outputs increase as the value of the discrimination factor goes up to around 1.0, whereas after this point they show a slow decreasing trend. This shows that utilizing the class labels in CCA increases its discriminative power up to a point. However, with much higher values of λ, as more weight is given to the prediction

of the class labels, the correlations of the extracted outputs decrease and the system converges to a single view LDA. Thus, along with the covariate correlations, the overall classification accuracy of a two view system is also compromised because each view end up using its own single view features without interaction with the other view.

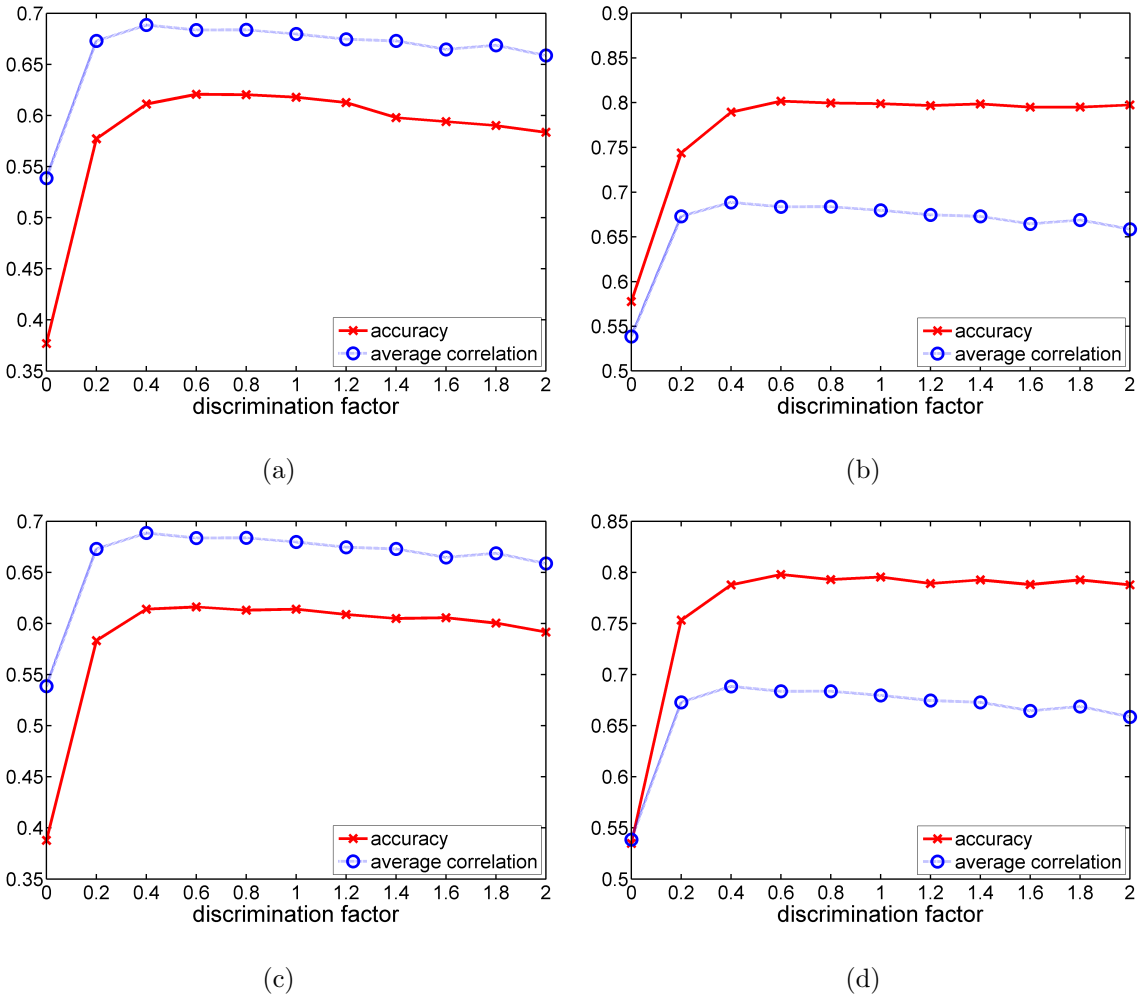
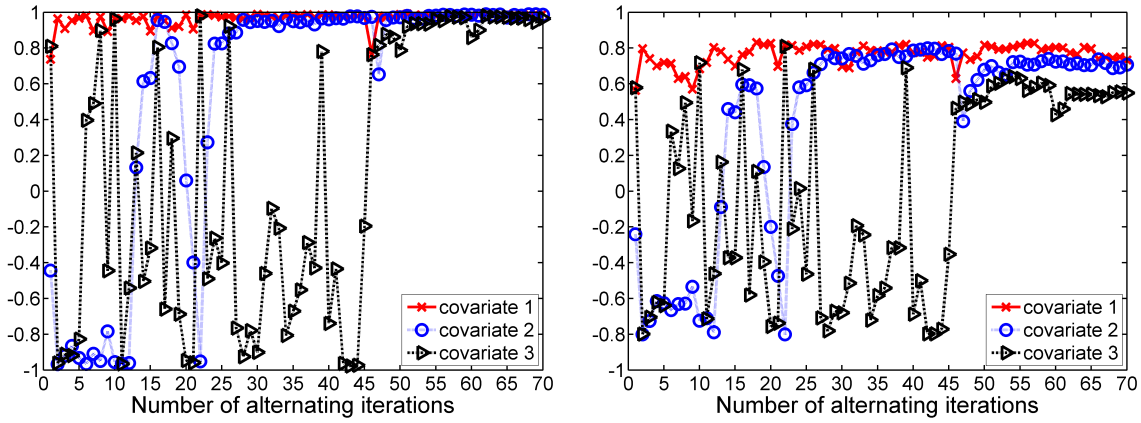


Figure 5.23. Discrimination factor versus average correlation and accuracy obtained on emotion recognition dataset with (a) SVM (view 1) (b) SVM (view 2) (c) 3-NN (view 1) (d) 3-NN (view 2).

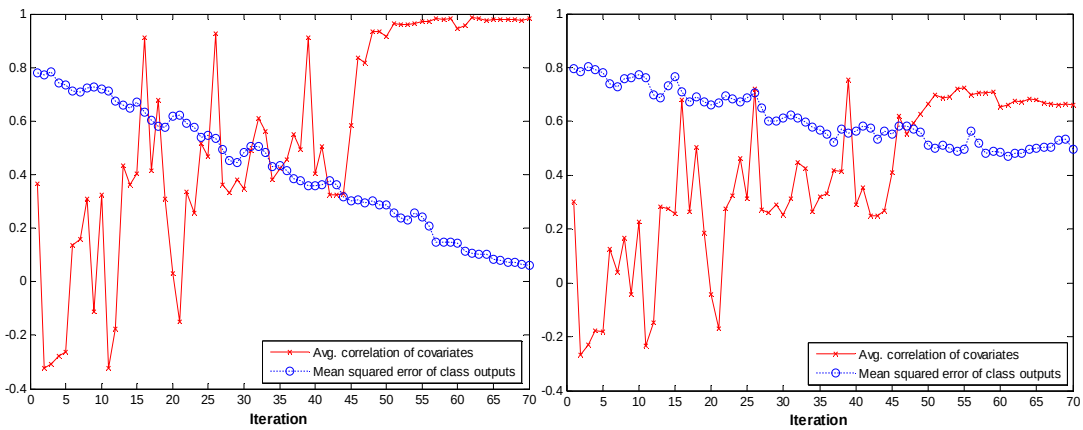
The convergence of the top 3 output units obtained with 5 training samples per class is shown in Figure 5.24. It can be seen that after 50 alternation iterations the output units converge on both training and test sets. In Figure 5.25, the average correlation of top 3 covariates and average mean squared errors (MSEs) of view 1 and view 2 class outputs are shown with respect to the number of alternation iterations of D-AR method. To demonstrate the relationship between the classification performance and mutual maximization of covariate correlations, in Figure 5.25, it is seen that the MSEs

of classification (MSEs between the class outputs and actual class labels) decreases as the average covariate correlation increases.



(a) (b)

Figure 5.24. Convergence of D-AR covariates on (a) training set and (b) test set of emotion recogntion dataset.



(a) (b)

Figure 5.25. Average correlation of top 3 covariates and average mean squared errors of view 1 and view 2 class outputs on (a) training set and (b) test set of CK+ dataset.

5.2.3. Object Recognition

The Columbia Object Image Library [135] dataset consist of images of natural 100 objects taken under different viewing angles. We have chosen 58 objects and categorized them into 6 classes which are vehicle, cup, animal knick-knack, fruit, round-shaped box, and rectangular-shaped box. The details of the dataset and its views are

given in Section 5.1.5.

Figure 5.26 shows SVM accuracies obtained with view 1 covariates on the test sets. It is seen that D-AR features have higher discriminative power on unseen test examples than those of the other methods (paired t-test, $p < 0.05$). In Figure 5.26, it is also seen that after D-AR, the supervised LDA features are the second most accurate method showing that feature extraction should benefit from the class-label information if available as D-AR does. While CCA performs better up to first 3 covariates than PCA+CCA and AR, AR gave significantly higher accuracies than CCA with 4 and 5 covariates. This indicates that CCA is successful at exploring the prominent correlated features; but it tunes to dummy dependencies on training set which do not hold on test set while exploring the less correlated features due to its sensitivity to outliers and noisy samples.

The k-NN accuracies obtained on test sets are shown in Figure 5.27. It can be seen that D-AR yields significantly highest accuracies (paired t-test, $p < 0.05$) on test sets 1, 2, 4, whereas the superiority of D-AR over LDA is not significant on test set 2. The accuracies obtained with view 2 covariates on object dataset are not shown as the views see the same training examples in different orders and thereby produce similar features.

5.3. KCCAmRMR Experiments

5.3.1. Methodology

In our KCCAmRMR experiments, we use a plain CCA implementation [72] for the KCCA implementation (as shown in Figure 2.4). Therefore, due to the computational time and space such kernels take when facing with large number of data samples, we select medium-sized datasets from UCI machine learning repository [47]. However, as KCCA is subjected to single variables for both sets, other fast approximations to kernels can be easily adopted for large datasets.

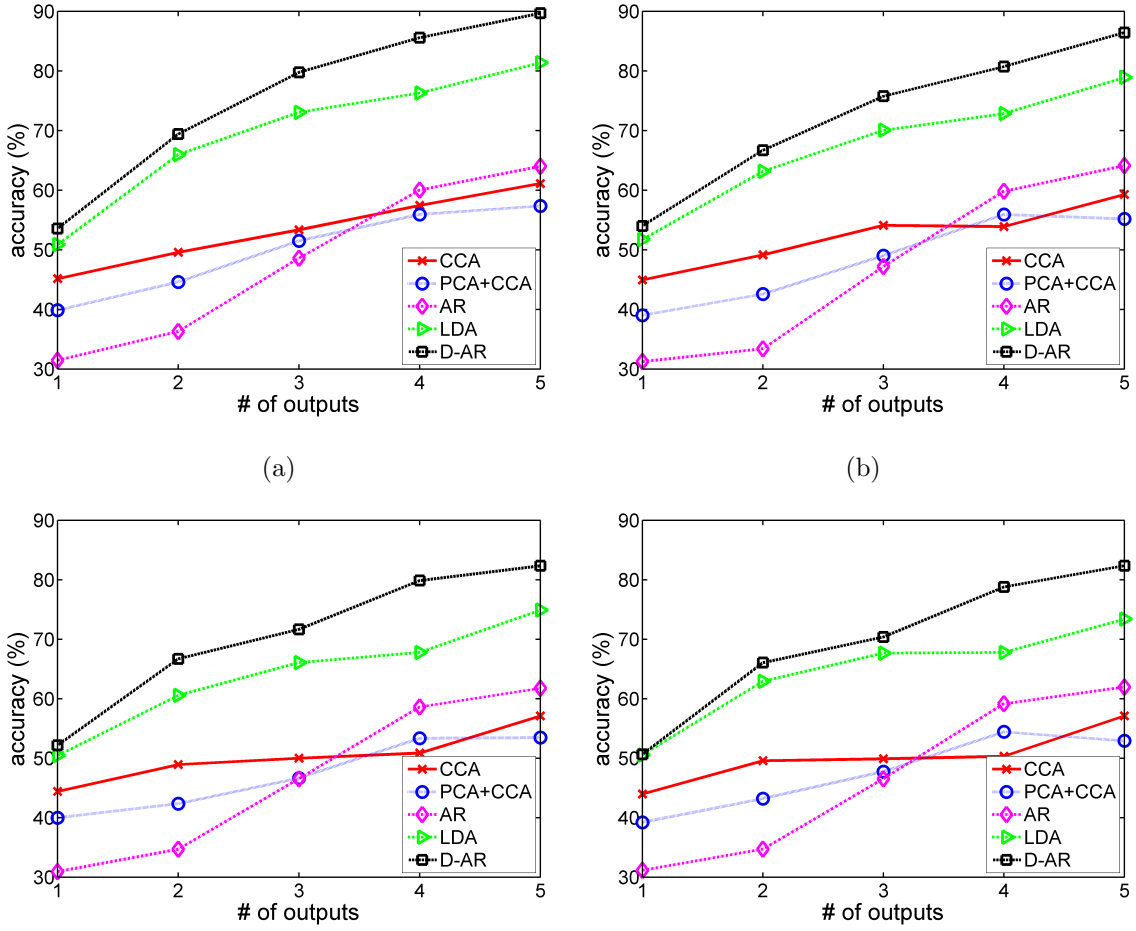


Figure 5.26. Number of outputs versus SVM accuracies obtained with view 1 covariates on (a) test set 1 (b) test set 2 (c) test set 3 (d) test set 4 of object recognition dataset.

We firstly revisit the toy problem on which we have shown that using mRMR can lead to inaccurate orderings of the variables because it does not deal with the type of the dependency, but only with its quantity. Then we give the experimental results on UCI datasets. For all the datasets, we normalize linear-valued features to zero-mean and unit-variance before feeding the selected features into KCCA for covariate extraction and also into SVMs for final classification; however, in mutual information computations during the selection phase, we discretize them to 9 discrete levels as in [40, 48] by converting the feature values between $\mu - \sigma/2$ and $\mu + \sigma/2$ to 0, the four intervals of size σ to the right of $\mu + \sigma/2$ to discrete levels from 1 to 4, and the four intervals of size σ to the left of $\mu - \sigma/2$ to discrete levels from -1 to -4 whereas very large positive or negative feature values are truncated and

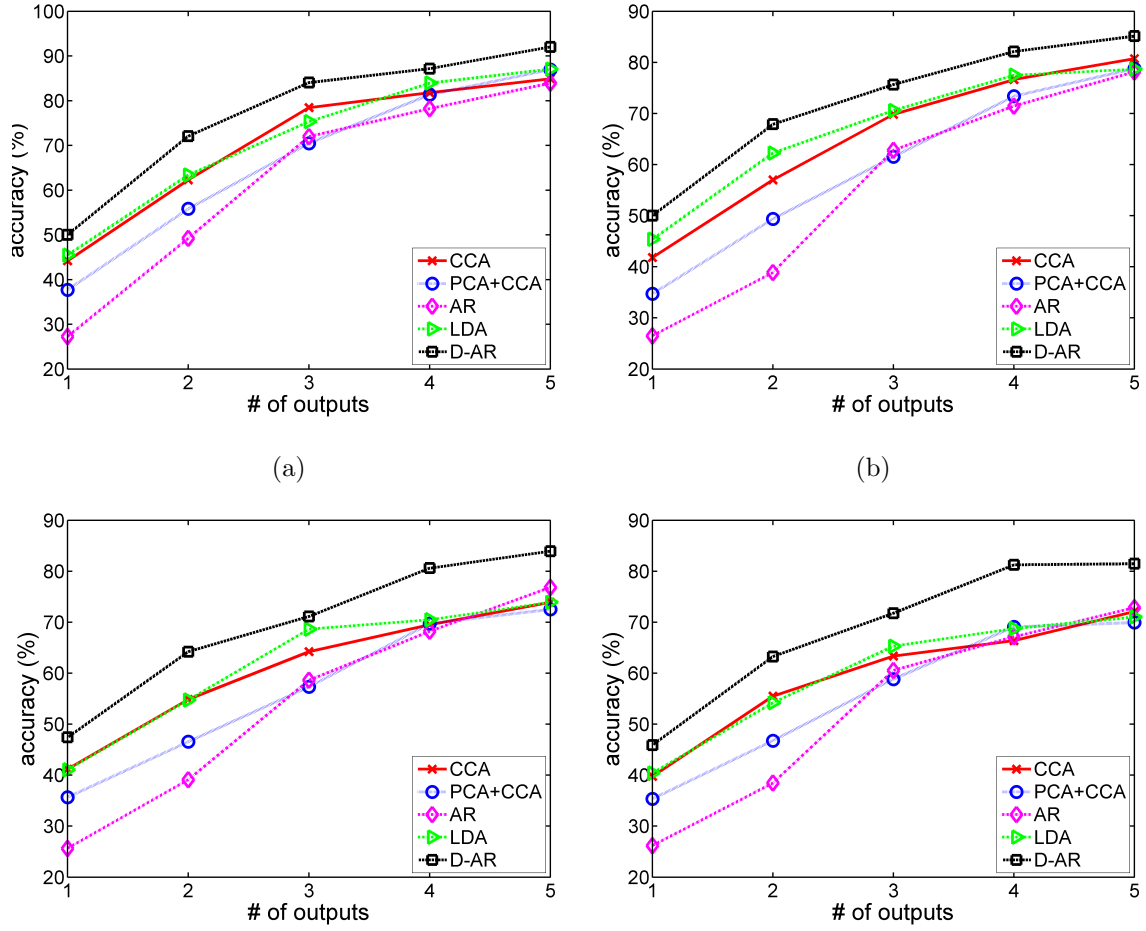


Figure 5.27. Number of outputs versus k-NN accuracies obtained with view 1 covariates on (a) test set 1 (b) test set 2 (c) test set 3 (d) test set 4 of object recognition dataset.

discretized to ± 4 appropriately. For KCCA computations, we set the g (kernel width) parameter of the kernel of the KCCAmRMR to 10.0 in all experiments. Since KCCA explores multiple correlated functions between the features and the target, we use the explored functions with correlation coefficient equal or greater than 0.10 for the UCI datasets (simply discarding the very low correlations for improving the time and space complexity of the algorithm).

5.3.2. Toy Problem from Section 2.4.2 (Revisited)

We reconsider the artificial example given in Section 2.4.2 (see Figure 2.3). In the first iteration, our method KCCAmRMR chooses X_1 as it has the highest mutual

information with the target variable t. As seen in Table 5.2, the highest correlation coefficient, ρ, with 0.6545 is naturally obtained between $f_1(X_1)$ and $g_1(t)$ (correlated functions between X_1 and t). In the second iteration, the relevance term is computed as the mutual information between t and the correlated function $f_j(X_j)$ of X_j for all the remaining features; that is, for simplicity we use a single covariate for each feature and denote it $f_j(X_j)$, for $j = 2, 3, 4$. As for the redundancy term, we calculate the mutual information between $f_1(X_1)$ and $f_j(X_j)$. This redundancy term gives us an approximation to the relevant redundancy (RR) because the correlated functions $f_1(X_1)$ and $f_j(X_j)$ are presumably filtered out of the irrelevant information to t.

Table 5.2. Mutual information among correlated functions and correlation coefficient ρ with the respective functions of the target variable t.

	$f_1(\mathbf{X}_1)$	$f_2(\mathbf{X}_2)$	$f_3(\mathbf{X}_3)$	$f_4(\mathbf{X}_4)$
$f_1(\mathbf{X}_1)$	2.1233	1.2416	0.1043	0.0068
$f_2(\mathbf{X}_2)$	1.2416	2.0876	0.1086	0.0062
$f_3(\mathbf{X}_3)$	0.1043	0.1086	1.8307	0.0053
$f_4(\mathbf{X}_4)$	0.0068	0.0062	0.0053	1.4720
ρ	0.6545	0.6353	0.4469	0.1734

It can be seen from Figure 2.3 that X_3 possesses r_7 and r_8 as common information with the target t, and only r_8 is redundant with the already selected variable X_1. Consequently, r_7 is the unique information that X_3 possess about t, and X_3 has a KCCAmRMR score of 0.0042. As it can be easily seen in Figure 2.3, X_4 has no redundancy with the selected variable X_1, and r_2 is the common information that it possess about t, and has a KCCAmRMR score of 0.0004 in this second iteration. Therefore, as a correct choice, X_3 will be the second feature selected by our method as it is the variable with the highest KCCAmRMR score. However, as explained in Section 2.4.2, in the second step, mRMR chooses X_4 as a mistake since it cancels out r_7 of X_3 with r_9. Table 5.3 shows KCCAmRMR scores of each variable during all iterations (first iteration being their true mutual information with the target).

Table 5.3. KCCAmRMR scores of variables in each iteration.

	$\mathbf{X}_1$	$\mathbf{X}_2$	$\mathbf{X}_3$	$\mathbf{X}_4$	**Selected Variable**
Iteration 1	0.4826	0.4536	0.2218	0.0399	$\mathbf{X}_1$
Iteration 2	-	-0.3672	0.0042	0.0004	$\mathbf{X}_3$
Iteration 3	-	-0.1245	-	0.0006	$\mathbf{X}_4$
Iteration 4	-	-0.0336	-	-	$\mathbf{X}_2$

5.3.3. UCI Datasets

We rank the top m features of the benchmark datasets with mRMR and KCCAmRMR and feed into SVMs. The number of features, samples, and classes of the benchmark datasets are shown in Table 5.4. We show the number of selected features with mRMR and KCCAmRMR versus classification accuracies of SVMs in Figure 5.28 for all datasets. The means and standard deviations of the top classification accuracies on the datasets are also shown in Table 5.4.

Table 5.4. SVMs average classification accuracies on UCI datasets with various number of features selected by mRMR and KCCAmRMR as input.

	# of features	# of **samples**	**# of classes**	**mRMR Accuracy (%)**	**KCCAmRMR Accuracy (%)**
Arrhythmia	270	452	16	69.94±0.97	70.49±0.98
Control Chart	99	600	6	78.31±5.27	83.41±8.69
Ionosphere	34	351	2	92.24±3.03	92.51±2.92
Libras	90	360	15	67.49±16.84	69.67±17.26
Sonar	60	208	2	77.57±2.34	80.58±3.47
Soybean	35	307	19	77.93±15.27	77.79±15.14

The average of SVMs accuracies with KCCAmRMR features is higher than with mRMR features for all the datasets in Table 5.4 except of soybean dataset for which the results are very close. It is also seen in Figure 5.28 that the superiority among mRMR and KCCAmRMR on soybean dataset varies with selected number of features. The maximum accuracy (approximately 90%) is obtained with 14 features by KCCAmRMR

which is reached by mRMR with 20 features. For sonar and control chart datasets, SVMs using features selected by KCCAmRMR is clearly superior to those using mRMR features; whereas the difference in accuracies for libras and ionosphere datasets is not obvious. For arrhythmia dataset, SVMs with KCCAmRMR features achieves 72% accuracy with only 25 features whereas the one with mRMR features achieves the same accuracy with 40 features. This probably originates from the mRMR method's problem of computation of relevant redundancy as we discussed in Section 2.4.2. Actually, we also see in Figure 5.28 that mRMR starts picking up 'good' features and catches the accuracy of KCCAmRMR but with more features which contradicts with the claim of mRMR method of selecting a minimal and compact subset of features.

5.4. Cluster Stacking Experiments

We present the experimental results of cluster stacking method (see Section 3.2.3.3) on a protein dataset with multiple views that are used to predict protein structure. In this section, we firstly give the description of the dataset and outline the methodology we followed. Then, we present the experimental results on this dataset.

5.4.1. Methodology

As described in Section 5.4.2, the protein dataset consists of 51 views. In this thesis, we evaluate the success of our cluster stacking approach with the aim of ranking the views of this multi-view protein dataset according to their discriminative power and selecting a minimal subset of views for the prediction of the protein structures.

As known, the use of individual feature selection methods such as mRMR described in Section 2.4.1 for selecting a minimal subset of views is not suitable since they ignore the presence of the views, dismantle them, and treat their variables intermixed along with those of others at best results in a complex uninterpretable predictive system for such multi-view datasets. For example, the multi-view protein dataset is treated as a single view dataset consisting of 1447 features. As the selected feature subset will comprise individual features from most of the views, this approach requires

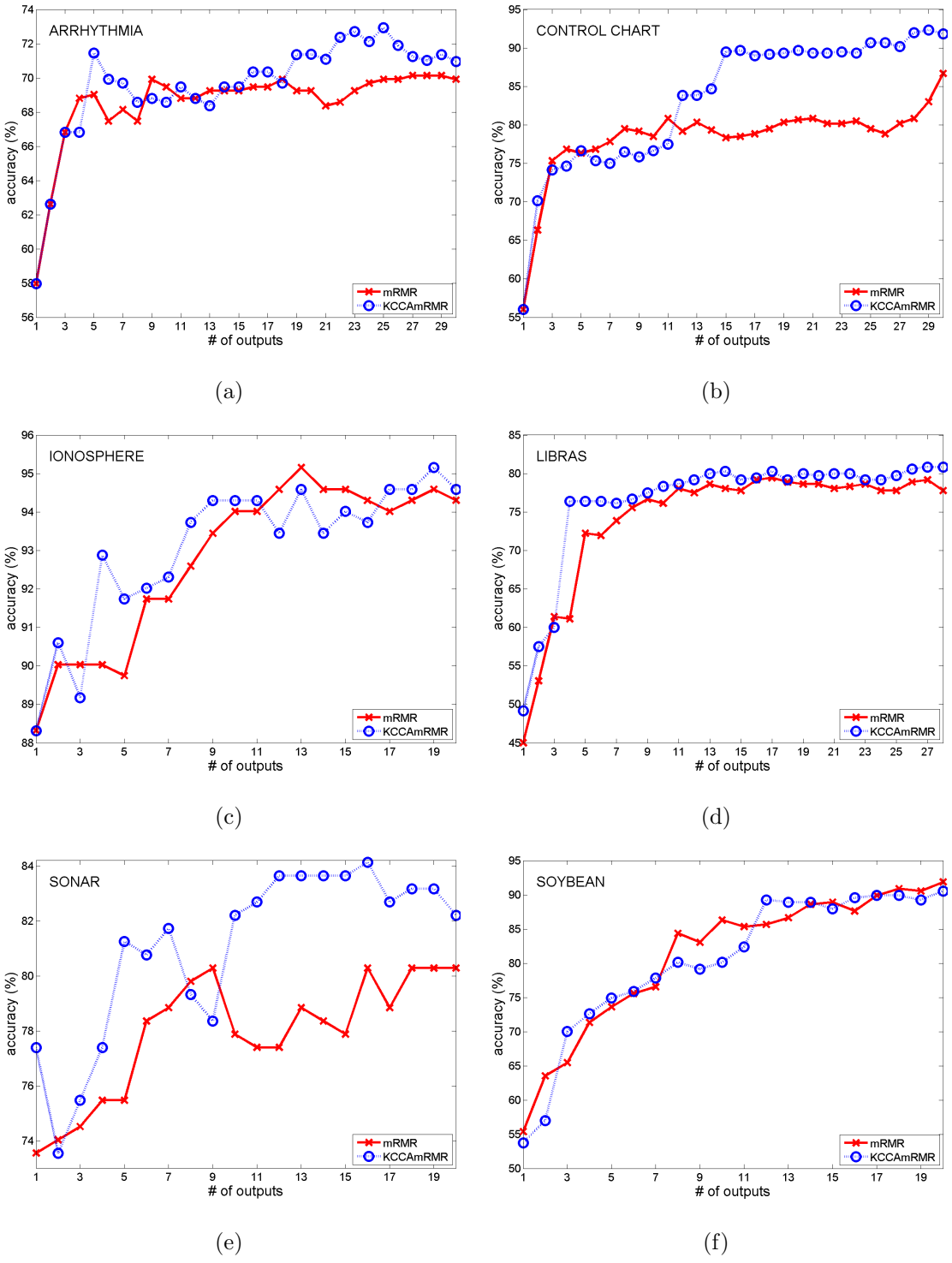


Figure 5.28. Number of selected features with mRMR and KCCAmRMR versus classification accuracies using SVMs.

measuring or computing majority of the views and results in a complex uninterpretable predictive system for researchers in these fields. Considering this, in order to choose a minimal subset of views without dismantling the views into individual features, rather

than presenting an $N \times n$ data matrix to the feature selection algorithm (where N denotes the number of samples and n the number of features), we firstly apply clustering to each view and for each sample, the feature vector of that view is replaced by the index of the cluster it belongs to. Thus, the dimensionality of each view is reduced to one and the whole dataset becomes $N \times V$, where V denotes the number of views. Due to the presence of randomness of clustering outcomes and availability of many different clustering methods, a mechanism for combining multiple clusterings is required to produce a more robust representation that avoids variations from one clustering to another. On this basis, we use our proposed cluster stacking approach and also co-association matrix based cluster ensembles given in Section 3.2.3.2) methods while reducing each view into a single variable, then apply mRMR to select a minimal set of views, and compare the discriminative power of the cluster ensembles methods by feeding the selected views to a classifier.

After reducing each view down to a single variable using the clustering of the views, we feed the dataset to mRMR for individual feature selection. As a preprocessing step, variables are discretized similar to [48] but using $\sigma/3$ step-size to obtain more discrete levels, i.e. the values between $\nu - \sigma/6$ and $\nu + \sigma/6$ are converted to 0, and bins of size $\sigma/3$ are created both on left and right of the bin 0; however, it must be noted that similar results are obtained with other discretization parameters. For statistical significance, the single clustering is repeated 30 times and the average accuracy of the multiple runs is computed.

There are several methods to create multiple partitions for cluster ensembles methods. In our experiments, for k-means clustering, we create 30 different partitions (i.e. the number of runs of k-means is set to 30) using all of the data observations with different random cluster seeds since k-means clustering algorithm has already a built-in randomness due to the starting cluster seeds. The number of clusters, k, is set to an initial value of 15 with the option of removing empty clusters during iterations. On the other hand, hierarchical clustering techniques do not have built-in randomness and they do not need any iterative optimization procedure to converge, so different runs of hierarchical clustering techniques with the same observations and parameters result in

the same clustering indices. Therefore, in order to create multiple partitions to obtain a final more representative partition, multiple subsets of observations are generated using bootstrapping data resampling technique [138]. Number of partitions is set to 30 as in k-means clustering and maximum number of clusters to keep in the hierarchical tree is set to 15. The single, average, and complete linkage criterions are tried and it is seen that the best results are obtained with complete linkage.

The features of the selected views by mRMR are fed into Support Vector Machine (SVMs) classifier. The kernel type of SVMs is determined as gaussian. The cost and kernel width parameters of the SVMs model are fixed to 50 and $1/n$, respectively, where n is the number of features. We bipartitioned the dataset randomly for train and test sets. The clustering algorithms and selection of views with mRMR are applied on the training set, and SVMs is trained on the training set using the features of the selected views. Finally, the obtained model is applied on the yet unseen test set and the unbiased prediction accuracy is obtained.

5.4.2. Description of Protein Dataset

Prediction of numerous functions and structures has a significant role in the comprehension of proteins. The multi-view dataset used in this thesis to evaluate the performance of the proposed cluster stacking approach is related to the structures of the proteins. These predictions are not only important in terms of their biological and medical functions but also one of the most challenging areas in bioinformatics due to high number of features and highly unbalanced class-prior distributions. The structure prediction dataset used contains 1086 proteins that are classified into four classes (with the number of samples in each class): alpha (223), beta (292), alpha and beta (331), and alpha + beta (240). Further details can be obtained from [70,139,140]. The definitions of these protein structural classes, which are based on the standard developed by Levitt and Chothia [141], are summarized as follows:

- All-Alpha - proteins with only small number of strands
- All-Beta - proteins with only small amount of helices

- Alpha and Beta - proteins that include both helices and strands and where strands are mostly parallel
- Alpha + Beta - proteins with both helices and strands and where strands are mostly anti-parallel

The proteins in this dataset are characterized by using a vector of 1447 features obtained from their sequence information, which have been shown to be very effective descriptors in previous studies [70, 139]. The dataset has seven main sets of views (with the number of features in each view): view 1 - amino acid composition (20), view 2 - dipeptide composition (400), view 3 - Moreau-broto autocorrelation (240), view 4 - Moran autocorrelation (240), view 5 - Geary autocorrelation (240), view 6 - composition-transition-distribution (147), and view 7 - sequence-order (160). Further there are 51 sub-subsets of these main views, each with varying number of features from 3 to 400 on which view selection methods are applied.

5.4.3. Results

The prediction accuracies obtained with individual feature selection (IFS), selected views using k-means as a single clustering (SC) algorithm, selected views with multiple clusterings using k-means clustering as the base clustering algorithm for our cluster stacking (CS) and co-association matrix based (CO-ASM) cluster ensembles methods, and all the features of the views as a single view (SV) into SVMs are shown in Table 5.5. The significance of the difference in accuracy is assessed by the two-sample t-test. For individual feature selection, we show the results from 100 to 500 selected features. For view selection, we show the results from 3 to 7 selected views. As it is seen from the results, the first 100 individually selected features with mRMR belong to 8 distinct views. On the other hand, the first 3 selected views using cluster stacking consist of only 63 features totally and perform higher accuracy than selected 100 individual features. We also see that the views selected with cluster stacking and co-association cluster ensembles give better results than the views selected with single clustering.

Table 5.5. Prediction accuracies of SVMs with inputs of mRMR selected features and views (k-means clustering).

IFS		**SV**		**SC (1)**	**CS (2)**	**CO-ASM**	
# of features (# of views)	**Accu.**	**Accu.**	**# of views**	**Accuracy**			**Level of significance (1-2)**
100 (8)	44.1		**3**	48.2	50.0	44.4	NS
200 (15)	51.4		**4**	53.4	**55.7**	45.1	*
300 (25)	53.3	54.5	**5**	53.6	**56.9**	51.9	*
400 (31)	55.5		**6**	54.1	57.5	**58.6**	*
500 (42)	56.5		**7**	56.5	**58.4**	**58.4**	**
****, p <0.01; *, p <0.05; NS, Not Significant**							

The obtained results show that using the consensus partitions of each view for view selection with mRMR by combining the solutions of many single clusterings results in more predictive SVMs classification models. The difference in accuracy between the single clustering mRMR and cluster stacking mRMR are statistically significant (two-sample t-test, Table 5.5) for the number of 4, 5, 6, and 7 selected views. The highest accuracy is achieved using the top-6 ranking views of co-association cluster ensembles method. However, with smaller number of selected views, i.e. with top-3, top-4, top-5 mRMR ranking views, the prediction models of the cluster stacking approach are better than the prediction models of co-association cluster ensembles. It is also seen that after selecting 400 individual features with mRMR, the accuracy of individual feature selection is getting closer to the accuracies of view selection methods. However, these 400 features belong to 31 distinct views meaning high computational cost to measure them. Besides, the interpretation and analysis of the system with 31 views is much more difficult than the one with 5 views.

The prediction accuracies obtained by feeding the selected views which are reduced to a single variable using hierarchical clustering are shown in Table 5.6. The individual feature selection and single view results are also included in Table 5.6 for comparison purposes. The results are similar to those obtained with k-means clustering (see Table 5.5). The highest accuracy is achieved using the top-7 ranking views of

cluster stacking. Also for other selected number of views, cluster stacking has higher accuracies than the other methods. A statistically significant difference in accuracy between the single clustering mRMR and cluster stacking mRMR is found for all selected number of views.

Table 5.6. Prediction accuracies of SVMs with inputs of mRMR selected features and views (hierarchical clustering).

IFS		**SV**		**SC (1)**	**CS (2)**	**CO-ASM**	
# of features (# of views)	**Accu.**	**Accu.**	**# of views**	**Accuracy**			**Level of significance (1-2)**
100 (8)	44.1		**3**	48.2	**53.8**	50.8	*
200 (15)	51.4		**4**	51.1	**55.6**	51.6	*
300 (25)	53.3	54.5	**5**	52.8	**55.8**	51.8	*
400 (31)	55.5		**6**	53.5	**55.8**	52.3	*
500 (42)	56.5		**7**	54.4	**57.1**	55.8	*
***, p <0.01**							

Cluster stacking approach can be best explained as follows. The mutual information matrices obtained in multiple clustering runs can be averaged, and then this matrix can be given to mRMR for rankings the views. Based on this averaged mutual information scores, the rankings will be more robust since it is computed over many clusterings instead of a single one. The mRMR code now needs to be modified so that it uses a $V \times V$ mutual information matrix instead of the original $N \times n$ dataset, where n is the number of features. However, in [39], we play the role of the plain-user of the mRMR feature selection package. Without any need for performing any modification of its source code, our approach stacks the outputs of multiple clusterings and then gives them to mRMR. Using the augmented clustering indices is the alternative way of the aforementioned averaging of the mutual information matrices of many clustering runs. The resulting augmented matrix of size $(N \times B) \times V$ is fed into mRMR directly to get the robust view rankings.

The simulation results on a multi-view protein structure prediction dataset showed that using the consensus partitions of each view for view selection with mRMR by

combining the solutions of many single clusterings results in more predictive SVMs classification models. We have also compared the proposed cluster stacking approach with co-association cluster ensembles method. We observed that while the highest accuracy is achieved using the top-6 ranking views of co-association cluster ensembles method, with smaller number of selected views, i.e. with top-3, top-4, top-5 mRMR ranking views, the prediction models of the cluster stacking approach are better than the prediction models of co-association cluster ensembles.

5.4.4. Discussion

The single clustering approach is based on firstly reducing the dimensionality of each view to one by applying the clustering algorithm for once. Then, we use the cluster indices of the views as input to mRMR. Therefore, the internal mechanisms of the mRMR method operate on the $V \times V$ (where V stands for the number of views) mutual information matrix storing the pairwise mutual information scores of the clustered views. Cluster stacking approach, on the other hand, can be best explained as follows. The mutual information matrices obtained in multiple clustering runs can be averaged, and then this matrix may be given to mRMR to rank the views. The rankings of the views will be more robust when their mutual dependence is computed based on this mutual information matrix since the dependence is computed over many clusterings instead of a single one. However, the mRMR code needs to be modified to implement this algorithm so that it uses a $V \times V$ mutual information matrix instead of the original $N \times n$ dataset. In other words, the columns of this matrix are not feature vectors but the averaged pairwise mutual information scores between the clusterings of the views. However, in the application of our cluster stacking approach together with mRMR, we play the role of the plain-user of the mRMR feature selection package. Thus, mRMR can be directly applied using the columns of augmented cluster index matrix $\mathbf{C}$ (given in Equation 3.11) as regular variables to get the consensus top-m views. This simply corresponds to modifying the dependency (Equation 2.29) and

redundancy (Equation 2.30) terms of mRMR as follows:

$$D = \frac{1}{|S|} \sum_{C_i \in S} I(C_i; t), \tag{5.1}$$

$$R = \frac{1}{|S|^2} \sum_{C_i, C_j \in S} I(C_i; C_j), \tag{5.2}$$

where C_i represents the ith column of $\mathbf{C}$, S is the selected set of views, and t is the target variable such as the structure of the protein in this thesis. Without any need for performing any modification of its source code, our approach stacks the outputs of multiple clusterings and then gives them to mRMR. Using the augmented clustering indices is the alternative way of the aforementioned averaging of the mutual information matrices of many clustering runs. The resulting augmented matrix of size $(N \times B) \times V$ is fed to mRMR directly to get the robust view rankings where B is the number of clustering runs.

The co-association matrix based cluster ensembles method is based on combining the multiple clusterings of a view into a matrix, and then finding the consensus partition by applying the clustering algorithm to this matrix. This final partition is used to compute the mRMR score of the view for view selection. The computational complexity of constructing the co-association based matrix is $O(B \times N^2)$ for each view. So, the computational complexity of mutual information computation is $O(V \times B \times N^2)$; that is we have to apply a clustering of co-association matrix which takes $O(B \times N^2)$ complexity for each one of the V views. On the other hand, our cluster stacking approach only stacks the cluster indices of the multiple clusterings of each view into a column vector, i.e., unlike co-association based matrix method, constructing a matrix is not required to represent the similarities between the pairs of samples by number of clusters shared by these samples in multiple clusterings. Thus, the complexity of our method is only limited with the complexity of the single clustering method we prefer to use.

5.5. Parallel Interacting Multi-view Learning Experiments

In this section, the proposed Parallel Interacting Multi-view Learning (PIML) method [42, 43] is demonstrated and compared with the classical ensemble approach on two protein datasets (secondary structure and subnuclear location prediction from sequence features) and one dataset on arrhythmia type prediction. For the protein datasets, we have the structure prediction and sub-nuclear location prediction tasks. These protein datasets are split into views using different sequence-driven protein feature extraction methods. For the arrhythmia dataset, we use random subspace method [102] to randomly split it into views. The task is to predict the type of arrhythmia.

5.5.1. Methodology

We use LIBSVM [137] implementation of Support Vector Machines (SVMs) as the classifier in the implementation of our PIML method. This package supports multiclass classification and can also produce the class posterior probability estimates of each one of the class labels for (multi)classification problems. For the comparison of single view, classical ensemble, and our proposed PIML methods on the protein datasets, firstly, we find the most suitable kernel type and parameter values of SVMs for each view. For this purpose, we try linear and Radian Basis Function (RBF) kernels, and various values for SVMs parameters C (cost) (C = 1, 2, 5, 10, 25, 50, 100) and g (the spread parameter) (g = 0.5/k, 1/k, 2/k, 5/k, etc, where k is the number of features in the view) for the RBF kernel. We train each view independently with all the combinations of kernel types and parameter values given above and determine the best fitted settings using 10-fold cross validation. For each SVMs parameter setting, firstly the training set is divided into 10 folds, then SVM is trained using the training examples of all the folds except the left-out fold, and finally the obtained model is applied on the left-out fold to obtain the accuracy. Thus, only training set examples are used to find the optimum SVM settings of each view. The optimized models are tested on unseen test examples only for comparing the individual accuracies before and after the interaction of the views, in other words test set examples are not used during the optimization of

individual SVM models. For the arrhythmia dataset, we use linear kernel SVMs since it gives better results when compared with polynomial and RBF kernels, and follow the same strategy with protein datasets for the optimization of the parameters and train-test splits.

5.5.2. Protein Datasets

The description of the protein datasets and the experimental results of PIML on protein datasets are presented in this section.

5.5.2.1. Description of Protein Datasets. In PIML experiments, we are concerned with the prediction of structure and sub-nuclear locations of the proteins. Making these predictions are challenging due to high number of features, small sample size, and highly unbalanced class-prior distributions. The structure prediction dataset used contains 1086 proteins that are classified into four classes (with the number of samples in each class): alpha (223), beta (292), alpha and beta (331), and alpha + beta (240). The details of this dataset is given in Section 5.4.2. The sub-nuclear location prediction dataset contains 714 proteins that are classified into nine classes (with the number of samples in each class): Chromatin (99), Heterochromatin (22), Nuclear Envelope (61), Nuclear Matrix (29), Nuclear Pore Complex (79), Nuclear Speckle (67), Nucleolus (307), Nucleoplasm (37) and Nuclear PML Body (13). Further details about protein datasets used in this study can be obtained from [3, 70, 140].

The proteins in these datasets are characterized by using a vector of 1447 and 1497 features, respectively, obtained from their sequence information, which have been shown to be very effective descriptors in previous studies [3, 70]. The views of protein datasets are obtained using different sequence-driven protein feature extraction methods. The first dataset has seven main sets of views (with the number of features in each view): view 1 - amino acid composition (20), view 2 - dipeptide composition (400), view 3 - Moreau-broto autocorrelation (240), view 4 - Moran autocorrelation (240), view 5 - Geary autocorrelation (240), view 6 - composition-transition-distribution (147), and

view 7 - sequence-order (160). The second dataset consists of the same 7 views as the first dataset with an additional 50-feature view called pseudo amino acid composition as the 8th view. Further there are 52 sub-subsets of these main views, each with varying number of features from 3 to 400. However, it should be noted that only the 8 main subsets are used as the independent views in our PIML experiments.

5.5.2.2. Results. We firstly train each view of the protein datasets individually, then test the obtained models on the test sets to find the optimal parameters by following the strategy described in Section 5.5.1. As seen in Table 5.7 and Table 5.8, the best results for protein datasets are obtained with RBF kernel. After finding the optimal SVMs setting of each individual view, we implement the interaction among the views proposed by PIML. For this task, we firstly train a separate SVM for each view with their determined best settings. Then we test the obtained models of each view on the training set. All the views are retrained using the features of the view along with the class posterior probability estimates obtained from the classifiers of the other views. For the protein structure prediction dataset, the size of the attached feature vector for each sample is $4 \times 6 = 24$ where 4 is the number of classes and 6 is the number of (other) views. In the protein sub-nuclear prediction dataset, there are 9 classes and 8 views. Thus, the size of the attached feature vector for each sample is $9 \times 7 = 63$.

The average and standard deviation of the 10-fold cross validation test set accuracies are shown in Table 5.7 and Table 5.8 for each view. As it is seen, the individual accuracy of each view is improved using the proposed technique with both linear and Gaussian kernels. The highest accuracies (60% and 58%) on the test sets of the structure and sub-nuclear prediction datasets are obtained using view 2 both with PIML.

For the structure prediction dataset, the results of the combination of the independent view and PIML predictions with different techniques are shown in Table 5.9. The highest final accuracy with 59.7% is obtained by combining the predictions of PIML using class posterior probability estimates as input to simple voting in which equal weighted class probability estimates are used. Since the distribution of classes is

Table 5.7. Protein structure prediction dataset: Average SVMs accuracies of views obtained using 10-fold cross validation.

SVM kernel	Method	V_1	V_2	V_3	V_4	V_5	V_6	V_7	Single view
Linear	Ind.	44 ± 2.2	43 ± 2.5	39 ± 1.8	40 ± 3.1	40 ± 3.0	41 ± 2.1	43 ± 2.5	47 ± 0.4
	PIML	$\mathbf{50 \pm 2.8}$	$\mathbf{43 \pm 3.1}$	$\mathbf{45 \pm 2.1}$	$\mathbf{44 \pm 4.0}$	$\mathbf{48 \pm 3.1}$	$\mathbf{47 \pm 2.5}$	$\mathbf{48 \pm 2.0}$	
RBF	Ind.	51 ± 5.4	53 ± 3.0	51 ± 5.2	48 ± 3.3	45 ± 0.2	48 ± 3.8	49 ± 5.9	53 ± 0.5
	PIML	$\mathbf{59 \pm 3.7}$	$\mathbf{60 \pm 3.4}$	$\mathbf{58 \pm 3.6}$	$\mathbf{58 \pm 2.9}$	$\mathbf{58 \pm 3.0}$	$\mathbf{58 \pm 3.4}$	$\mathbf{57 \pm 2.3}$	

Table 5.8. Protein sub-nuclear location prediction dataset: Average SVMs accuracies of views obtained using 10-fold cross validation.

SVM kernel	Method	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8
Linear	Ind.	42 ± 3.7	45 ± 4.0	38 ± 3.5	46 ± 3.2	38 ± 3.8	37 ± 2.4	42 ± 3.4	39 ± 2.1
	PIML	$\mathbf{47 \pm 3.5}$	$\mathbf{52 \pm 4.1}$	$\mathbf{49 \pm 3.4}$	$\mathbf{50 \pm 3.6}$	$\mathbf{50 \pm 4.0}$	$\mathbf{47 \pm 2.9}$	$\mathbf{46 \pm 3.1}$	$\mathbf{45 \pm 2.7}$
RBF	Ind.	50 ± 4.9	54 ± 4.7	48 ± 5.0	48 ± 2.4	47 ± 2.7	52 ± 2.3	47 ± 3.0	52 ± 3.6
	PIML	$\mathbf{56 \pm 6.8}$	$\mathbf{58 \pm 5.3}$	$\mathbf{57 \pm 4.8}$	$\mathbf{56 \pm 4.7}$	$\mathbf{57 \pm 5.5}$	$\mathbf{57 \pm 3.6}$	$\mathbf{56 \pm 5.2}$	$\mathbf{56 \pm 6.2}$

imbalanced, for assessing the prediction performance of the methods, we present the accuracies of each class (i.e., sensitivity of the classes) and also the average accuracy which is computed as the number of samples for each class is equal. PIML with simple voting again gives the highest equally weighted accuracy (59.1%).

Table 5.9. Protein structure prediction dataset: Average ensemble SVMs accuracies obtained using 10-fold cross validation.

Class (# of samples)	Hard voting		Simple voting		Weighted voting	
	Ensemble	PIML	Ensemble	PIML	Ensemble	PIML
Alpha (223)	8.2 ± 7.0	$\mathbf{64.8 \pm 8.6}$	56.9 ± 7.0	$\mathbf{62.8 \pm 7.3}$	57.8 ± 6.8	$\mathbf{64.5 \pm 5.5}$
Beta (292)	$\mathbf{73.4 \pm 8.1}$	71.3 ± 6.0	70.2 ± 8.1	$\mathbf{72.0 \pm 7.0}$	$\mathbf{80.1 \pm 8.6}$	71.9 ± 7.7
Alpha and Beta (331)	$\mathbf{81.5 \pm 6.9}$	68.9 ± 7.9	$\mathbf{86.5 \pm 6.9}$	69.4 ± 6.5	$\mathbf{75.5 \pm 7.0}$	69.5 ± 8.1
Alpha + Beta (240)	3.8 ± 4.5	$\mathbf{28.2 \pm 9.1}$	7.9 ± 4.5	$\mathbf{29.2 \pm 9.4}$	11.3 ± 11.0	$\mathbf{26.8 \pm 13}$
Weighted average	57.2 ± 6.7	$\mathbf{59.3 \pm 8.4}$	58.4 ± 6.6	$\mathbf{59.7 \pm 8.0}$	58.9 ± 8.0	$\mathbf{59.1 \pm 8.5}$
Average	54.2 ± 6.5	$\mathbf{58.3 \pm 7.9}$	55.1 ± 6.5	$\mathbf{59.1 \pm 7.8}$	57.5 ± 1.2	$\mathbf{58.1 \pm 8.5}$

As PIML already creates an ensemble of multiple stackings, it results in a better accuracy than making single ensemble stacking. We see similar results in Table 5.10 which belongs to the sub-nuclear prediction dataset. It is seen that the highest accuracy is obtained with simple voting PIML. Classical ensemble approach performs better in

terms of classification accuracy only for the prediction nucleolus proteins class. However, this originates from the fact that ensemble method does not learn a generalizable model enough to discriminate the classes with small sample size from each other and thus labels most of the test examples as nucleolus proteins which has the highest prior probability with 307 samples.

Table 5.10. Protein sub-nuclear location prediction dataset: Average ensemble SVMs accuracies obtained using 10-fold cross validation.

Class (# of samples)	**Hard voting**		**Simple voting**		**Weighted voting**	
	Ensemble	**PIML**	**Ensemble**	**PIML**	**Ensemble**	**PIML**
Chromatin (99)	39.4 ± 1.2	$\mathbf{45.5 \pm 2.4}$	44.4 ± 2.1	$\mathbf{51.5 \pm 3.8}$	58.5 ± 2.89	$\mathbf{64.5 \pm 3.4}$
Heterochromatin (22)	0	$\mathbf{13.6 \pm 2.3}$	0	$\mathbf{13.6 \pm 2.3}$	13.6 ± 4.6	$\mathbf{27.2 \pm 4.4}$
Nucl. envelope (61)	24.6 ± 0.9	$\mathbf{49.2 \pm 3.2}$	47.2 ± 3.2	$\mathbf{50.2 \pm 3.0}$	44.3 ± 3.4	$\mathbf{49.2 \pm 2.9}$
Nuclear matrix (29)	10.3 ± 0.0	10.3 ± 0.0	10.3 ± 0.0	10.3 ± 0.0	20.7 ± 5.9	20.7 ± 5.9
Nucl. pore complex (79)	46.1 ± 3.6	$\mathbf{56.7 \pm 5.7}$	56.7 ± 4.1	$\mathbf{64.6 \pm 5.9}$	56.9 ± 4.1	$\mathbf{65.7 \pm 4.4}$
Nucl. speckle (67)	8.9 ± 0.0	$\mathbf{58.2 \pm 4.7}$	58.2 ± 3.9	$\mathbf{59.8 \pm 4.2}$	40.3 ± 3.6	$\mathbf{48.2 \pm 4.9}$
Nucleolus (307)	$\mathbf{90.9 \pm 6.7}$	68.4 ± 4.3	$\mathbf{76.1 \pm 4.6}$	73.2 ± 4.9	$\mathbf{79.0 \pm 2.0}$	70.2 ± 2.5
Nucleoplasm (37)	0	$\mathbf{16.2 \pm 1.5}$	16.2 ± 1.5	$\mathbf{24.3 \pm 2.0}$	16.2 ± 4.4	$\mathbf{24.3 \pm 4.8}$
Nucl. PML body (13)	0	0	0	0	0	0
Weighted average	52.1 ± 2.7	$\mathbf{54.2 \pm 3.0}$	53.8 ± 3.2	$\mathbf{57.0 \pm 3.3}$	56.3 ± 3.3	56.3 ± 3.7
Average	24.5 ± 1.4	$\mathbf{35.3 \pm 2.7}$	34.3 ± 2.2	$\mathbf{38.6 \pm 2.9}$	36.6 ± 3.4	$\mathbf{42.8 \pm 3.7}$

As another measure of comparison, we compute the average posterior probability estimates of the correctly classified samples for both ensemble and PIML algorithms. The results which are shown in Table 5.11 for both protein datasets reveal to what extent the classical ensemble and PIML methods are confident for the samples which are correctly classified. It is seen that PIML results in more confident predictions than those of classical ensemble approach.

Table 5.11. Average posterior probability estimates of correctly classified samples by ensemble and piml algorithms.

Method	**Dataset**	**Hard voting**	**Simple voting**	**Weighted voting**
Ensemble	**Structure**	0.68 ± 0.035	0.73 ± 0.051	0.75 ± 0.052
PIML		$\mathbf{0.68 \pm 0.035}$	$\mathbf{0.77 \pm 0.043}$	$\mathbf{0.76 \pm 0.060}$
Ensemble	**Sub-nuclear**	0.53 ± 0.060	0.58 ± 0.075	0.60 ± 0.094
PIML		$\mathbf{0.57 \pm 0.048}$	$\mathbf{0.59 \pm 0.085}$	$\mathbf{0.60 \pm 0.087}$

5.5.3. Arrhythmia Dataset

In this section, we present the comparative results of classical ensemble and PIML methods on an experimental dataset regarding arrhythmia.

5.5.3.1. Description of Arrhythmia Dataset. The arrhythmia dataset [142] which is available on the UCI machine learning archive [47] contains 452 samples from 16 classes. The aim is to classify the sample in one of the 16 groups of arrhythmia. The explanation of these groups are as follows: class 1 means 'normal', classes 2 to 15 refer to different classes of arrhythmia, and class 16 refers to one of the unclassified arrhythmia types [143]. The dataset consists of 279 features. We generated various number of feature subsets using the attribute bagging method [102] and use the obtained multi-view dataset in our experiments to evaluate the performance of ensemble and PIML methods.

5.5.3.2. Results. The overall accuracies obtained by combining the predictions of individual views in order to obtain the final predictions using hard voting, simple voting and weighted voting strategies are shown in Table 5.12. As seen, the classification performance of our proposed PIML method is better than that of the classical ensemble with all the voting strategies. The SVMs accuracy obtained when the arrhythmia dataset is used as single view ($67.7 \pm 2.8\%$) is also lower than the SVMs accuracy obtained with PIML. The PIML accuracies are comparable with the accuracies obtained by Peng *et al.* [48] considering that they converted the multi-class arrhythmia type prediction problem to a binary classification problem aiming to discriminate the normal class from the abnormal.

5.5.4. Discussion

The experimental results on protein and arrhythmia datasets show that PIML achieves better predictions than the classical ensemble approach because it aims at using the views in a more sophisticated way by considering the possible interdependences among the views. In PIML algorithm, features of each view are augmented by the class

Table 5.12. Arrhythmia dataset: 10-fold cross validation results with various number of views.

# of views	Hard Voting		Simple voting		Weighted voting	
	Ensemble	**PIML**	**Ensemble**	**PIML**	**Ensemble**	**PIML**
5	65.7 ± 4.2	$\mathbf{71.7 \pm 4.2}$	65.0 ± 3.8	$\mathbf{71.9 \pm 4.4}$	63.0 ± 4.7	$\mathbf{68.9 \pm 4.7}$
10	67.7 ± 3.9	$\mathbf{71.5 \pm 3.5}$	67.5 ± 3.6	$\mathbf{71.3 \pm 3.8}$	64.9 ± 4.9	$\mathbf{69.2 \pm 4.2}$
15	67.5 ± 3.7	$\mathbf{70.3 \pm 3.9}$	67.0 ± 3.3	$\mathbf{70.6 \pm 4.3}$	65.7 ± 5.6	$\mathbf{67.8 \pm 4.0}$
20	67.0 ± 4.6	$\mathbf{70.6 \pm 4.8}$	67.0 ± 3.2	$\mathbf{70.4 \pm 4.4}$	65.9 ± 5.0	$\mathbf{68.5 \pm 3.9}$
25	67.0 ± 4.2	$\mathbf{70.8 \pm 4.9}$	66.6 ± 5.5	$\mathbf{71.0 \pm 4.9}$	65.2 ± 5.3	$\mathbf{69.1 \pm 4.7}$
30	67.0 ± 3.5	$\mathbf{70.5 \pm 5.2}$	66.8 ± 3.0	$\mathbf{71.2 \pm 5.4}$	65.7 ± 5.1	$\mathbf{70.1 \pm 5.1}$
Average	67.0 ± 4.0	$\mathbf{70.9 \pm 4.4}$	66.7 ± 3.7	$\mathbf{71.1 \pm 4.4}$	65.1 ± 5.1	$\mathbf{69.0 \pm 4.5}$

posterior probability estimates of the other views. Ensemble approach to supervised multi-view learning assumes the views are conditionally independent given the class labels but views of real multi-view datasets can be interdependent. PIML algorithm models these dependencies without going into the extent of merging the features of all views (i.e. the single view approach) by interacting during learning and also prevents curse of dimensionality in contrast to single-view learning by merging only the original features of the individual view with the summary information of the other views. Thus, PIML can be seen, at least, as an ensemble of stacking networks, in which rather than creating a single stacking that uses all the estimates from all the views, the estimate of each view is utilized multiple times, thus leading to multiple stackings to be ensembled, which in turn leading to higher number of blocking configurations and more confident posterior probability estimates.

6. CONCLUSIONS AND FUTURE WORK

Canonical correlation analysis (CCA) is a well-established statistical method which maximizes the correlation between linear combinations of two sets of variables. Although CCA has been proposed by Hotelling [5] a long time ago, it has gained considerable interest in the recent machine learning studies due to the rapid rise of multi-view datasets, which consist of multiple types of data about the same underlying phenomenon. CCA has been used for many purposes in the literature but most of them utilize CCA for correlation detection and feature extraction.

Naturally, the growing interest in CCA applications has led to studies focusing on the improvement of CCA from many aspects. This thesis presents our efforts to improve the robustness and discriminative ability of CCA. Experimental results on various datasets demonstrate the usefulness of the proposed methods. Besides, we show that CCA can be used in a feature selection algorithm to quantify the relations between features and target variable. This thesis also includes our works on ensemble classification and clustering.

6.1. Contributions of the Thesis

In this thesis, we propose an ensemble CCA approach to improve the generalization capability of CCA. The generalization capability of a machine learning algorithm determines how well it will perform on a new test set. Combining diverse and sufficient models, ensemble approaches have been successfully applied for obtaining better classification accuracy than the individual learners. They have also found use in combining multiple clusterings to obtain better partitioning of the datasets. Despite the fact that the ensemble approach can be utilized effectively for classification, regression, clustering, and so on, it has not been applied to CCA based dimensionality reduction. This thesis introduces an ensemble CCA method, which aims at reaching a final set of covariates by combining sets of covariates obtained from various resamplings. The ECCA approach combines many weak correlations obtained from resampled subsets

of the views with the aim of producing a final set of stronger correlations with good generalization on the unseen test set examples.

As CCA is a technique for dimensionality reduction, we perform experiments to evaluate the generalization of ECCA on both the test set correlations of the covariates and the test set accuracy of classification performed on these reduced dimensions. We present the experimental results on emotion recognition, handwritten digit recognition, content-based retrieval, and multiple view object recognition. We also evaluate the robustness of ECCA covariates on a toy dataset.

Besides the superiority of the proposed ECCA approach over the traditional CCA, our experimental results show that when the subsamples used in the ensemble have high diversity (less overlap), ECCA yields higher correlations. Thus, when the sample size is large enough when compared with the dataset dimensionality, the Partitioning Ensemble CCA (ECCA-P) gives higher correlations and classification accuracy. Moreover, not having enough diversity among the subsamples, jackknife approach (ECCA-J) performs only slightly better than the traditional CCA. Also, bagging approach to create the subsamples has been found to be the most robust to the number of groups to consider. That is, as ECCA-B can create as many groups using bagging and may keep both the diversity and sufficiency high enough. Whereas, when the number of groups is large, ECCA-P may suffer from low sufficiency and ECCA-J may suffer from low diversity. As the diversity and sufficiency are shown to be necessary conditions for the ensemble approaches to work for classification and clustering problems, the results obtained by ensemble CCA also show that diverse and sufficient subsamples are required for the ensemble approach to outperform the CCA models based on both the individual subsamples and the whole sample. Thus, under the diversity and sufficiency conditions, ECCA has been shown to have better generalization than that of the traditional CCA.

This thesis also introduces a discriminative feature extraction method by a neural implementation of CCA, called Discriminative Alternating Regression (D-AR). Given two different representations of the same underlying phenomenon, CCA is used as a feature extraction method and the extracted features are generally used in classification

algorithms to deal with the curse of dimensionality. However, the use of CCA in classification problems mainly suffers from the fact that CCA does not utilize the class labels in its traditional analytical solution. Therefore, CCA tends to preserve highly correlated features instead of less correlated but more discriminative features in the reduced subspace. The proposed D-AR method integrates the class labels into its feature extraction framework, and so can really take advantage of the class labels in CCA computation and also avoids the use of sensitive sample covariance matrices.

D-AR is a linear dimensionality reduction method based on the alternating regression approach implemented by a multi-layer neural network with a "linear" hidden layer. In the hidden layer, feature vectors of each view is transformed into a low-dimensional subspace that preserves correlated and also discriminative information. For a comparison with the alternative linear two-view dimensionality reduction techniques CCA, PCA+CCA, alternating regression, and LDA methods are used. We have applied these methods for feature extraction on emotion recognition and object recognition datasets. The experimental results show that the covariates extracted by D-AR have higher classification accuracies.

In Chapter 2, after giving an overview of existing studies that use CCA for different tasks, we show that CCA can also be used for feature selection. As stated in feature selection literature, for obtaining a minimal yet expressive subset of variables, while maximizing the joint dependency with the target variable, the redundancy among selected variables must also be reduced to a minimum. One of the most successful studies is by Peng et. al. [48] called mRMR (minimum Redundancy – Maximum Relevance) approach which is based on choosing a subset that aims at minimizing the pairwise redundancies in the set among the selected variables while maximizing the overall relevance with the target variable. It is true that such redundancies (variables with nearly the same information content about the target variable) must be avoided in order to obtain a minimal subset that maximizes the joint inferential dependency with the target variable. However, as for the redundancy term of a candidate variable, mRMR approach computes plainly its mutual information with the already selected variables, and does not consider whether that redundancy is related to the target variable or not.

As a more effective redundancy term, in this thesis, we propose a method called Kernel Canonical Correlation Analysis-mRMR (KCCAmRMR) which deals with the part of the redundancy between the correlated functions (with the target variable) of the candidate and the selected variables. Computing the redundancies between the correlated functions quantifies the unique information that a candidate variables possesses about the target (i.e. unique in the sense that different contribution from what is already learnable from the selected variables). We utilize kernel canonical correlation analysis (KCCA) to explore the correlated functions (covariates) between features and the target variable and use these functions while computing the relevance and redundancy terms in a similar fashion to mRMR. Experimental results on benchmark data sets from the UCI Machine Learning Repository show that the proposed KCCAmRMR method can choose better set of features than mRMR and opens a promising alternative way in feature selection using the renowned CCA and kernel methods.

We also present our preliminary studies based on multi-view clustering and classification. We propose a cluster ensembles method for multi-view datasets, called cluster stacking, to combine the multiple clustering solutions of the views. Cluster stacking approach is based on augmenting the clustering indices of all the clustering trials into a consensus matrix and using this augmented consensus partition as the final partition. The augmented cluster index matrix used to combine the solutions of multiple clusterings resembles the covariate correspondence matrix that we propose to solve the covariate correspondence problem encountered in ensemble CCA approach. In fact, the cluster index matrix that we use to combine multiple clusterings of multiple views provided a basis for us to propose the covariate correspondence matrix to address the covariate correspondence problem of ensemble CCA. We present the experimental results of our cluster stacking approach on a protein dataset with multiple views that are used to predict protein structure. We also propose a multi-view ensemble learning technique called Parallel Interacting View Learning (PIML) for classification problems. PIML is shown to be suitable for predictive tasks in high dimensional biomedical/bioinformatics datasets. The experimental results on two real protein datasets (secondary structure and subnuclear location prediction from sequence features) and one real dataset on arrhythmia type prediction show that PIML achieves better predictions than the clas-

sical ensemble approach because it aims at using the views in a more sophisticated way by considering the possible interdependences among the views.

6.2. Future Work

The results obtained in Section 5.1 show that the proposed ensemble approach to CCA may be giving us a framework that can be used to obtain ensemble dimensionality reduction (see also Discussion Section, Section 5.1.8). For example, firstly an ensemble construction method such as bootstrapping can be used to obtain many subsamples of the original dataset, then PCA can be applied separately to each subsample, and the obtained PCA projections can be combined using a similar approach to ECCA. Such an approach may give us a robust set of reduced dimensions with high-variance.

The proposed ensemble CCA approach is used to improve the generalization capability of traditional CCA. As a future direction, ensemble approach can be used to further improve the generalization of existing robust CCA methods, e.g. the neural implementation of CCA is known to improve the generalization and its ensemble version can further improve its generalization.

In this thesis, we have proposed two separate methods to improve CCA: ECCA to obtain more generalizable covariates and D-AR to incorporate the class labels into the framework of CCA. The preliminary results on the emotion recognition task showed that the generalization capacity of D-AR features on unseen test set examples can be increased by combining the ECCA and D-AR methods.

The nonlinear extensions of the proposed method ECCA and D-AR are straightforward. The nonlinear version of ECCA can be implemented using kernel trick. As for D-AR, the "linear" hidden layer of its multi-layer neural network architecture can be replaced with a nonlinear activation function to obtain D-AR features that can distinguish data that are not linearly separable.

In the KCCAmRMR framework, we have used only the first pair of correlated

functions to quantify the redundancy among features and the target variable. As a future direction, more than just a single pair of correlated functions can be extracted and used possibly using to better represent the relations between features and the target variable.

As another future direction, the proposed PIML method may be improved by utilizing the multi-kernel methods for combining the original view features and the class posterior probability estimates. Thus, different kernels will be applicable to the data and posterior probabilities.

REFERENCES

1. Sakar, B. E., M. E. Isenkul, C. O. Sakar, A. Sertbas, F. Gurgen, S. Delil, H. Apaydin and O. Kursun, "Collection and Analysis of a Parkinson Speech Dataset with Multiple Types of Sound Recordings", *IEEE Journal of Biomedical and Health Informatics*, Vol. 17, No. 4, pp. 828–834, 2013.

2. Breukelen, M. V., R. P. W. Duin, D. M. J. Tax and J. E. den Hartog, "Handwritten Digit Recognition by Combined Classifiers", *Kybernetika*, Vol. 34, No. 4, pp. 381–386, 1998.

3. Nanuwa, S. S., A. Dziurla and H. Seker, "Weighted Amino Acid Composition based on Amino Acid Indices for Prediction of Protein Structural Classes", *Proceedings of the 9th Information Technology and Applications in Biomedicine*, pp. 1–4, IEEE Computer Society, Larnaka, Cyprus, 2009.

4. Thompson, B., "Canonical Correlation Analysis: Uses and Interpretation", J. L. Sullivan (Editor), *Quantitative Applications in the Social Sciences (Book 47)*, Sage University Paper Series, SAGE Publications, USA, 1984.

5. Hotelling, H., "Relations Between Two Sets of Variates", *Biometrika*, Vol. 28, pp. 312–377, 1936.

6. Kettenring, J. R., "Canonical Analysis of Several Sets of Variables", *Biometrika*, Vol. 58, pp. 433–451, 1971.

7. Hardoon, D., S. Szedmak and J. S. Taylor, "Canonical Correlation Analysis: An Overview with Application to Learning Methods", *Neural Computation*, Vol. 16, pp. 2639–2664, 2004.

8. Kursun, O., E. Alpaydin and O. Favorov, "Canonical Correlation Analysis Using Within-class Coupling", *Pattern Recognition Letters*, Vol. 32, No. 2, pp. 134–144,

2011.

9. Varis, O., "Associations Between Lake Phytoplankton Community and Growth Factors — A Canonical Correlation Analysis", *Hydrobiologia*, Vol. 210, No. 3, pp. 209–216, 1991.

10. Bartos, A., A. Fekete and B. Sarvari, "Study of Ecological Factors and Nutrient Content Variables in Wheat Applying Canonical Correlation Analysis", *Enytermeles*, Vol. 40, No. 2, pp. 111–124, 1991.

11. Wade, J. B., L. M. Dougherty, R. P. Hart, A. Rafii and D. D. Price, "A Canonical Correlation Analysis of the Influence of Neuroticism and Extraversion on Chronic Pain, Suffering, and Pain Behavior", *Pain*, Vol. 51, No. 1, pp. 67–73, 1992.

12. Finkenberg, M. E., J. M. Dinucci, S. L. Mccune and E. D. Mccune, "Personal Incentives for Exercise and Body Esteem: A Canonical Correlation Analysis", *Journal of Sports Medicine and Physical Fitness*, Vol. 34, No. 4, pp. 398–402, 1994.

13. Cserhati, T. and E. Forgacs, "Use of Canonical Correlation Analysis for the Evaluation of Chromatographic Retention Data", *Chemometrics and Intelligent Laboratory Systems*, Vol. 28, pp. 305–313, 1995.

14. Degani, A., M. Shafto and L. Olson, "Canonical Correlation Analysis: Use of Composite Heliographs for Representing Multiple Patterns", D. B. Plummer, R. Cox and N. Swoboda (Editors), *Diagrammatic Representation and Inference*, Vol. 4045 of *Lecture Notes in Computer Science*, pp. 93–97, Springer Berlin - Heidelberg, 2006.

15. Alexander, C., *The Phenomenon of Life*, Routledge, Berkeley, CA, 2002.

16. Zheng, W., X. Zhou, C. Zou and L. Zhao, "Facial Expression Recognition Using Kernel Canonical Correlation Analysis (KCCA)", *IEEE Transactions on Neural*

Networks, Vol. 17, No. 1, pp. 233–237, 2006.

17. Cao, Y., P. G. Green and P. A. Holden, "Microbial Community Composition and Denitrifying Enzyme Activities in Salt Marsh Sediments", *Applied and Environmental Microbiology*, Vol. 74, No. 24, pp. 7585–7595, 2008.

18. Ozkan, M. M., M. Adak and Z. Kocabas, "An Investigation on the Relationship Between Yield and Canopy Components in Wheat (Triticum aestivum)", *Tarım Bilimleri Dergisi*, Vol. 14, No. 2, pp. 148–153, 2008.

19. Yang, W., D. Yi, Z. Lei, J. Sang and S. Z. Li, "2D-3D Face Matching using CCA", *Proceedings of the 8th IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 1–6, IEEE Computer Society, Amsterdam, The Netherlands, 2008.

20. Parkhomenko, E., D. Tritchler and J. Beyene, "Sparse Canonical Correlation Analysis with Application to Genomic Data Integration", *Statistical Applications in Genetics and Molecular Biology*, Vol. 8, No. 1, 2009.

21. Sun, Q.-S., S.-G. Zeng, Y. Liu, P.-A. Heng and D.-S. Xia, "A New Method of Feature Fusion and Its Application in Image Recognition", *Pattern Recognition*, Vol. 38, pp. 2437–2448, 2005.

22. Sun, N., Z. hai Ji, C. rong Zou and L. Zhao, "Two-dimensional Canonical Correlation Analysis and Its Application in Small Sample Size Face Recognition", *Neural Computing and Applications*, Vol. 19, pp. 377–382, 2010.

23. Melzera, T., M. Reitera and H. Bischof, "Appearance Models Based on Kernel Canonical Correlation Analysis", *Pattern Recognition*, Vol. 36, pp. 1961–1971, 2003.

24. Romanazzi, M., "Influence in Canonical Correlation Analysis", *Psychometrika*, Vol. 57, pp. 237–259, 1992.

25. Branco, J. A., C. Croux, P. Filzmoser and M. R. Oliveira, "Robust Canonical Correlations: A Comparative Study", *Computational Statistics*, Vol. 20, pp. 203–229, 2005.

26. Sakar, C. O. and O. Kursun, "A Method for Combining Mutual Information and Canonical Correlation Analysis: Predictive Mutual Information and Its Use in Feature Selection", *Expert Systems with Applications*, Vol. 39, No. 3, pp. 3333–3344, 2012.

27. Karnel, G., "Robust Canonical Correlation and Correspondence Analysis", *In: The Frontiers of Statistical Scientific and Industrial Applications, (Volume II of the proceedings of ICOSCO-I, The First International Conference on Statistical Computing)*, pp. 335–354, American Sciences Press, 1991.

28. Croux, C. and C. Dehon, "Analyse Canonique Basee sur des Estimateurs Robustes de la Matrice de Covariance", *La Revue de Statistique Appliquge*, Vol. 2, pp. 5–26, 2002.

29. Taskinen, S., C. Croux, A. Kankainen, E. O. E and H. Oja, "Canonical Analysis based on Scatter Matrices", *Journal of Multivariate Analysis*, Vol. 97, No. 2, pp. 359–384, 2006.

30. Lai, P. L. and C. Fyfe, "Canonical Correlation Analysis Using Artificial Neural Networks", *In European Symposium on Artificial Neural Networks, ESANN98*, pp. 363–367, 1998.

31. Hsieh, W. W., "Nonlinear Canonical Correlation Analysis by Neural Networks", *Neural Networks*, Vol. 13, pp. 1095–1105, 2000.

32. Via, J., I. Santamaria and J. Perez, "A Learning Algorithm for Adaptive Canonical Correlation Analysis of Several Data Sets", *Neural Networks*, Vol. 20, pp. 139–152, 2007.

33. He, Y., L. Zhao and C. Zou, "Face Recognition Based on PCA/KPCA Plus CCA", L. Wang, K. Chen and Y. S. Ong (Editors), *Advances in Natural Computation*, Vol. 3611 of *Lecture Notes in Computer Science*, pp. 71–74, Springer Berlin - Heidelberg, 2005.

34. Sun, Q.-S., P.-A. Heng, Z. Jin and D.-S. Xia, "Face Recognition Based on Generalized Canonical Correlation Analysis", D. S. Huang and G. H. X. P. Zhang (Editors), *Advances in Intelligent Computing*, Vol. 3645 of *Lecture Notes in Computer Science*, pp. 958–967, Springer Berlin - Heidelberg, 2005.

35. Kim, T.-K., J. Kittler and R. Cipolla, "Discriminative Learning and Recognition of Image Set Classes Using Canonical Correlations", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 29, No. 6, pp. 1005–1018, 2007.

36. Sun, T., S. Chen, J. Yang and P. Shi, "A Supervised Combined Feature Extraction Method for Recognition", *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008)*, pp. 1043–1048, IEEE Computer Society, Pisa, Italy, 2008.

37. Shin, Y. J. and C. H. Park, "Analysis of Correlation Based Dimension Reduction Methods", *International Journal of Applied Mathematics and Compututer Science*, Vol. 21, No. 3, pp. 549–558, 2011.

38. Sakar, C. O., O. Kursun and F. Gurgen, "A Feature Selection Method Based on Kernel Canonical Correlation Analysis and the Minimum Redundancy Maximum Relevance Filter Method", *Expert Systems with Applications*, Vol. 39, No. 3, pp. 3432–3437, 2012.

39. Sakar, C. O., O. Kursun, H. Seker and F. Gurgen, "Combining Multiple Clusterings for Protein Structure Prediction", *International Journal of Data Mining and Bioinformatics (to appear)*, 2013.

40. Kursun, O., C. O. Sakar, O. Favorov, N. Aydin and F. Gurgen, "Using Covari-

ates for Improving the Minimum Redundancy Maximum Relevance Feature Selection Method", *Turkish Journal of Electrical Engineering and Computer Sciences*, Vol. 18, No. 6, pp. 989–1002, 2010.

41. Sakar, C. O., O. Kursun and F. Gurgen, "Ensemble Canonical Correlation Analysis", *Applied Intelligence (to appear)*, 2013.

42. Sakar, C. O., O. Kursun, H. Seker, F. Gurgen, N. Aydin and O. Favorov, "Combining Multiple Views: Case Studies on Protein and Arrhythmia Features", *Engineering Applications of Artificial Intelligence (to appear)*, 2013.

43. Sakar, C. O., O. Kursun, H. Seker, F. Gurgen, N. Aydin and O. Favorov, "Parallel Interacting Multiview Learning: An Application to Prediction of Protein Subnuclear Location", *Proceedings of 9th International Conference on Information Technology and Applications In Biomedicine*, pp. 587–590, Larnaka, Cyprus, 2009.

44. Sakar, C. O., O. Kursun and F. Gurgen, "Feature Extraction based on Discriminative Alternating Regression", *Proceedings of the 13th Mediterranean Conference on Medical and Biological Engineering and Computing (MEDICON 2013)*, pp. 819–822, Seville, Spain, 2013.

45. Sakar, C. O. and O. Kursun, "A Hybrid Method for Feature Selection Based on Mutual Information and Canonical Correlation Analysis", *Proceedings of the 20th International Conference on Pattern Recognition*, pp. 4360–4363, Istanbul, Turkey, 2010.

46. Kursun, O., B. E. Sakar, M. E. Isenkul, C. O. Sakar, F. Gurgen, S. Delil, H. Apaydin, M. Tommerdahl and O. V. Favorov, "Analysis of Effects of Parkinson's Disease on the Somatosensory System via CM-4 Tactile Stimulator", *Proceedings of International Conference on Applied Informatics and Health and Life Sciences*, pp. 57–60, Istanbul, Turkey, 2013.

47. Asuncion, A. and D. Newman, *UCI Machine Learning Repository*, Tech. rep.,

University of California, Irvine, CA: University of California, Department of Information and Computer Science, 2007.

48. Peng, H., F. Long and C. Ding, "Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 8, pp. 1226–1238, 2005.

49. Bartlett, M. S., "Further Aspects of the Theory of Multiple Regression", *Mathematical Proceedings of the Cambridge Philosophical Society*, Vol. 34, No. 1, pp. 33–40, 1938.

50. Alpaydin, E., *Introduction to Machine Learning*, The MIT Press, Cambridge, Massachusetts, 2004.

51. Sun, T. and S. Chen, "Class Label versus Sample Label-Based CCA", *Applied Mathematics and Computation*, Vol. 185, 2007.

52. Loog, M., B. V. Ginneken and R. P. W. Duin, "Dimensionality Reduction of Image Features Using the Canonical Contextual Correlation Projection", *Pattern Recognition*, Vol. 38, pp. 2409–2418, 2005.

53. Sargin, M. E., Y. Yemez, E. Erzin and A. M. Tekalp, "Audiovisual Synchronization and Fusion Using Canonical Correlation Analysis", *IEEE Transactions on Multimedia*, Vol. 9, No. 7, pp. 1396–1403, 2007.

54. Huang, H. and H. He, "Super-Resolution Method for Face Recognition Using Nonlinear Mappings on Coherent Features", *IEEE Transactions on Neural Networks*, Vol. 22, No. 1, pp. 121–130, 2011.

55. Bellman, R. E., *Adaptive Control Processes*, Princeton University Press, Princeton, New Jersey, USA, 1961.

56. Christoudias, C. M., R. Urtasun and T. Darrell, "Multi-View Learning in the Presence of View Disagreement", *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence (UAI 2008)*, pp. 88–96, AUAI Press, Helsinki, Finland, 2008.

57. Blum, A. and T. Mitchell, "Combining Labeled and Unlabeled Data with Co-training", *Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT' 98)*, pp. 92–100, ACM, Madison, WI, USA, 1998.

58. Yarowsky, D., "Unsupervised Word Sense Disambiguation Rivaling Supervised Methods", *Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics (ACL' 95)*, pp. 189–196, ACL, Cambridge, Massachusetts, USA, 1995.

59. Okun, O. and H. Priisalu, "Multiple Views in Ensembles of Nearest Neighbor Classifiers", *Proceedings of the ICML Workshop on Learning with Multiple Views (In conjunction with the 22nd International Conference on Machine Learning)*, pp. 51–58, ACM, Bonn, Germany, 2005.

60. Nigam, K. and R. Ghani, "Analyzing the Effectiveness and Applicability of Co-training", *Proceedings of the 9th International Conference on Information and Knowledge Management*, pp. 86–93, ACM, New York, USA, 2000.

61. Zhang, J. and D. Zhang, "A Novel Ensemble Construction Method for Multi-view Data Using Random Cross-view Correlation Between Within-class Examples", *Pattern Recognition*, Vol. 44, No. 6, pp. 1162–1171, 2011.

62. Liu, Y., X. Liu and Z. Su, "A New Fuzzy Approach for Handling Class Labels in Canonical Correlation Analysis", *Neurocomputing*, Vol. 71, pp. 1735–1740, 2008.

63. Peng, Y., D. Zhang and J. Zhang, "A New Canonical Correlation Analysis Algorithm with Local Discrimination", *Neural Processing Letters*, Vol. 31, pp. 1–15, 2010.

64. Kursun, O. and E. Alpaydin, "Canonical Correlation Analysis for Multiview Semisupervised Feature Extraction", L. R. *et al.* (Editor), *Artificial Intelligence and Soft Computing*, Vol. 6113 of *Lecture Notes in Computer Science*, pp. 430–436, Springer Berlin - Heidelberg, 2010.

65. Hardoon, D. R., J. S. Shawe-Taylor and O. Friman, *KCCA Feature Selection for fMRI Analysis*, Tech. rep., Faculty of Engineering, Science and Mathematics, University OF Southampton, 2004.

66. Paskaleva, B., M. M. Hayat, Z. Wang, J. S. Tyo and S. Krishna, "Canonical Correlation Feature Selection for Sensors with Overlapping Bands: Theory and Application", *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 46, No. 10, pp. 3346–3358, 2008.

67. Zheng, C., S. Schwartz, R. Chapkin, R. Carroll and I. Ivanov, "Feature Selection for High-dimensional Integrated Data", `http://arxiv.org/abs/1111.6283`, 2011, accessed at December 2013.

68. Kwak, N. and C. H. Choi, "Input Feature Selection by Mutual Information Based on Parzen Window", *IEEE Transactions Pattern Analysis and Machine Intelligence*, Vol. 24, No. 12, pp. 1667–1671, 2002.

69. Shannon, C. E., "A Mathematical Theory of Communication", *Bell System Technical Journal*, Vol. 27, pp. 379–423, 1948.

70. Li, Z. R., H. H. Lin, L. Y. Han, L. Jiang, X. Chen and Y. Z. Chen, "PROFEAT: A Web Server for Computing Structural and Physicochemical Features of Proteins and Peptides from Amino Acid Sequence", *Nucleic Acids Research*, Vol. 34, pp. 32–37, 2006.

71. Estevez, P. A., M. Tesmer, C. A. Perez and J. M. Zurada, "Normalized Mutual Information Feature Selection", *IEEE Transactions on Neural Networks*, Vol. 20, No. 2, pp. 189–201, 2009.

72. Borga, M., *Learning Multidimensional Signal Processing*, Ph.D. Thesis, Department of Electrical Engineering, Linköping University, Linköping, Sweden, 1998.

73. Gurban, M. and J. P. Thiran, "Information Theoretic Feature Extraction for Audio-Visual Speech Recognition", *IEEE Transactions on Signal Processing*, Vol. 57, No. 12, pp. 4765–4776, 2009.

74. Favorov, O. and D. Ryder, "SINBAD: A Neocortical Mechanism for Discovering Environmental Variables and Regularities Hidden in Sensory Input", *Biological Cybernetics*, Vol. 90, pp. 191–202, 2004.

75. Kursun, O. and O. Favorov, "SINBAD Automation of Scientific Discovery: From Factor Analysis to Theory Synthesis", *Natural Computing*, Vol. 3, No. 2, pp. 207–233, 2004.

76. Akaho, S., "A Kernel Method for Canonical Correlation Analysis", *Proceedings of the International Meeting of the Psychometric Society (IMPS2001)*, Springer-Verlag, 2001.

77. Bach, F. R. and M. I. Jordan, "Kernel Independent Component Analysis", *Journal of Machine Learning Research*, Vol. 3, pp. 1–48, 2002.

78. Kursun, O. and O. Favorov, "Feature Selection and Extraction Using an Unsupervised Biologically-Suggested Approximation to Gebelein's Maximal Correlation", *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 23, No. 3, pp. 337–358, 2010.

79. Akyuz, S. O. and G. W. Weber, "On Numerical Optimization theory of Infinite Kernel Learning", *Journal of Global Optimization*, Vol. 48, No. 2, pp. 215–239, 2010.

80. Scholkopf, B. and A. Smola, *Learning with Kernels*, The MIT Press, Cambridge, Massachusetts, 2002.

81. Wold, H., "Nonlinear Estimation by Iterative Least Squares Procedures", F. N. David (Editor), *Research Papers in Statistics*, Festschrift for J. Neyman, pp. 411–444, Wiley, New York, 1966.

82. Croux, C. and P. Filzmoser, "A Robust Biplot Representation of Two-way Tables", A. Rizzi, M. Vichi and H. H. Bock (Editors), *Advances in Data Science and Classification*, Studies in Classification, Data Analysis, and Knowledge Organization, pp. 355–361, Springer Berlin - Heidelberg, 1998.

83. Lai, P. and C. Fyfe, "A Neural Implementation of Canonical Correlation Analysis", *Neural Networks*, Vol. 12, No. 10, pp. 1391–1397, 1999.

84. Kursun, O. and O. Favorov, "What Can SVMS Teach Each Other?", *In Artificial Neural Networks in Engineering (ANNIE 2004)*, ASME Press, New York, 2004.

85. Godambe, V. P., "Estimating Functions", Vol. 7 of *Oxford Statistical Science Series*, Oxford University Press, New York, 1991.

86. Rousseeuw, P. J., "Multivariate Estimation with High Breakdown Point", W. G. *et al.* (Editor), *Mathematical Statistics and Applications*, Vol. B, pp. 283–297, 1985.

87. Rousseuw, P. J. and K. V. Driessen, "A Fast Algorithm for the Minimum Covariance Determinant Estimator", *Technometrics*, Vol. 41, pp. 212–223, 1999.

88. Zhang, J., D. J. Olive and P. Ye, "Robust Covariance Matrix Estimation with Canonical Correlation Analysis", *International Journal of Statistics and Probability*, Vol. 1, No. 2, pp. 119–136, 2012.

89. Sun, S., "A Survey of Multi-view Machine Learning", *Neural Computing and Applications*, Vol. 23, pp. 2031–2038, 2013.

90. Strehl, A. and J. Ghosh, "Cluster Ensembles - A Knowledge Reuse Framework for

Combining Multiple Partitions", *Journal of Machine Learning Research*, Vol. 3, pp. 583–617, 2002.

91. Hansen, L. and P. Salamon, "Neural Network Ensembles", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, pp. 993–1001, 1990.

92. Breiman, L., "Bagging Predictors", *Machine Learning*, Vol. 24, No. 2, pp. 123–140, 1996.

93. Sharkey, A., "On Combining Artificial Neural Nets", *Connection Science*, Vol. B, pp. 299–313, 1999.

94. Ghosh, J., "Multiclassifier Systems: Back to the Future", F. Roli and J. Kittler (Editors), *Multiple Classier Systems*, Vol. 2364 of *Lecture Notes in Computer Science*, pp. 1–15, Springer Berlin - Heidelberg, 2002.

95. Krogh, A. and J. Vedelsby, "Neural Network Ensembles, Cross Validation, and Active Learning", D. S. Touretzky, M. C. Mozer and M. E. Hasselmo (Editors), *Advances in Neural Information Processing Systems*, Vol. 7, The MIT Press, 1965.

96. Hashem, S., "Optimal Linear Combinations of Neural Networks", *Neural Networks*, Vol. 10, No. 4, pp. 599–614, 1997.

97. Opitz, D. and J. Shavlik, "Actively Searching for an Effective Neural-network Ensemble", *Connection Science*, Vol. 8, No. 3, pp. 337–353, 1996.

98. Opitz, D. and J. Shavlik, "Generating Accurate and Diverse Members of a Neural Network Ensemble", D. Touretsky, M. Mozer and M. Hasselmo (Editors), *Advances in Neural Information Processing Systems*, Vol. 8, pp. 535–54, The MIT Press, 1996.

99. Kuncheva, L. I. and C. J. Whitaker, "Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy", *Machine Learning*, Vol. 51,

pp. 181–207, 2003.

100. Windeatt, T., "Accuracy/Diversity and Ensemble MLP Classifier Design", *Neural Networks*, Vol. 17, No. 5, pp. 1194–1211, 2006.

101. Opitz, D. W., "Feature Selection for Ensembles", *Proceedings of 16th National Conference on Artificial Intelligence*, pp. 379–384, The MIT Press, Orlando, Florida, USA, 1999.

102. Bryll, R., R. Gutierrez-Osuna and F. Quek, "Attribute Bagging: Improving Accuracy of Classifier Ensembles by Using Random Feature Subsets", *Pattern Recognition*, Vol. 36, pp. 1291–1302, 2003.

103. Wolpert, D., "Stacked Generalization", *Neural Networks*, Vol. 5, No. 2, pp. 241–259, 1992.

104. Bach, F., G. R. G. Lanckriet and M. I. Jordan, "Multiple Kernel Learning, Conic Duality, and the SMO Algorithm", *Proceedings of the 21st International Conference on Machine Learning (ICML '04)*, pp. 6–13, ACM, Banff, Alberta, Canada, 2004.

105. Ruping, S. and T. Scheffer, "Learning with Multiple Views", *Proposal for a Workshop at the 22nd International Conference on Machine Learning (ICML '05)*, ACM, Bonn, Germany, 2005.

106. Yang, P., Y. H. Yang, B. B. Zhou and A. Y. Zomaya, "A Review of Ensemble Methods in Bioinformatics", *Current Bioinformatics*, Vol. 5, pp. 296–308, 2010.

107. Masoud, H., S. Jalili and S. M. H. Hasheminejad, "Dynamic Clustering Using Combinatorial Particle Swarm Optimization", *Applied Intelligence*, Vol. 38, No. 3, pp. 289–314, 2013.

108. Ayad, H. G. and M. S. Kamel, "On Voting-based Consensus of Cluster Ensem-

bles", *Pattern Recognition*, Vol. 43, No. 5, pp. 1943–1953, 2010.

109. Lu, Z., Y. Peng and H. S. Horace, "Combining Multiple Clusterings Using Fast Simulated Annealing", *Pattern Recognition Letters*, Vol. 32, No. 15, pp. 1956–1961, 2011.

110. Mimaroglu, S. and E. Erdil, "Combining Multiple Clusterings Using Similarity Graph", *Pattern Recognition*, Vol. 44, pp. 694–703, 2011.

111. Lloyd, S. P., "Least Squares Quantization in PCM", *IEEE Transactions on Information Theory*, Vol. 28, No. 2, pp. 129–137, 1982.

112. Hamerly, G. and C. Elkan, "Alternatives to the k-means Algorithm that Find Better Clusterings", *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, pp. 600–607, Washington, DC, USA, 2002.

113. Tan, P. N., M. Steinbach and V. Kumar, *Introduction to Data Mining*, Addison-Wesley, 2005.

114. Fred, A. L. N., "Finding Consistent Clusters in Data Partitions", *Proceedings of 3d International Workshop on Multiple Classifier Systems (MCS 2002)*, pp. 309–318, Cagliari, Italy, 2002.

115. Fred, A. L. N. and A. K. Jain, "Data Clustering Using Evidence Accumulation", *Proceedings of the 16th International Conference on Pattern Recognition (ICPR 2002)*, pp. 276–280, Quebec City, 2002.

116. Mimaroglu, S. and M. Yagci, "CLICOM: Cliques for Combining Multiple Clusterings", *Expert Systems with Applications*, Vol. 39, No. 2, pp. 1889–1901, 2012.

117. Ghaemi, R., N. Sulaiman, H. Ibrahim and N. Mustapha, "A Survey: Clustering Ensembles Techniques", *World Academy of Science, Engineering and Technology*,

Vol. 50, pp. 636–645, 2009.

118. Topchy, A., A. K. Jain and W. Punch, "Combining Multiple Weak Clusterings", *Proceedings of the Third IEEE International Conference on Data Mining (ICDM 2003)*, pp. 331–338, Florida, USA, 2003.

119. Topchy, A., A. K. Jain and W. Punch, "A Mixture Model for Clustering Ensembles", *Proceedings of the SIAM International Conference on Data Mining (ICDM 2004)*, pp. 379–390, Michigan State University, USA, 2004.

120. Lau, K. M., K.-M. Kim and S. S. P. Shen, "Potential Predictability of Seasonal Precipitation Over the United States from Canonical Ensemble Correlation Predictions", *Geophysical Research Letters*, Vol. 29, No. 7, pp. 1(1)–1(4), 2002.

121. Mo, K. C. and W. M. Thiaw, "Ensemble Canonical Correlation Prediction of Precipitation Over the Sahel", *Geophysical Research Letters*, Vol. 29, No. 12, pp. 111–114, 2002.

122. Mo, K. C., "Ensemble Canonical Correlation Prediction of Surface Temperature over the United States", *Journal of Climate*, Vol. 16, pp. 1665–1683, 2003.

123. Shao, J. and D. Tu, *The Jackknife and Bootstrap*, Springer-Verlag, 1995.

124. Rao, C. R., "The Use and Interpretation of Principal Component Analysis in Applied Research", *The Indian Journal of Statistics*, Vol. 26, No. 4, pp. 329–358, 1964.

125. Fukunaga, K., *Introduction to Statistical Pattern Recognition*, Academic Press, Boston, USA, 1990.

126. Breiman, L. and J. H. Friedman, "Estimating Optimal Transformations for Multiple Regression and Correlation", *Journal of the American Statistical Association*, Vol. 80, pp. 580–598, 1985.

127. Rumelhart, D. E., G. E. Hinton and R. J. Williams, "Learning Internal Representations by Error Propagation", D. E. Rumelhart, J. L. McClelland and P. R. Group (Editors), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Vol. 1, pp. 318–362, MIT Press Cambridge, MA, USA, 1986.

128. Foldiak, P., "Forming Sparse Representations by Local Anti-Hebbian Learning", *Biological Cybernetics*, Vol. 64, No. 2, pp. 165–170, 1990.

129. Lucey, P., J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar and I. Matthews, "The Extended Cohn-Kande Dataset (CK+): A Complete Facial Expression Dataset for Action Unit and Emotion-specified Expression", *Prooceedings of the Third IEEE Workshop on CVPR for Human Communicative Behavior Analysis*, pp. 94–101, San Francisco, CA, USA, 2010.

130. Ulukaya, S., *Affect Recognition from Facial Expressions for Human-Computer Interaction*, M.Sc. Thesis, Bahcesehir University, Istanbul, Turkey, 2011.

131. Ulukaya, S. and C. E. Erdem, "Estimation of the Neutral Face Shape Using Gaussian Mixture Models", *ICASSP*, pp. 1385–1388, 2012.

132. Karaali, A., *Face Detection and Facial Expression Recognition Using Moment Invariants*, M.Sc. Thesis, Bahcesehir University, Istanbul, Turkey, 2012.

133. Sakar, C. O., O. Kursun, A. Karaali and C. E. Erdem, "Feature Extraction for Facial Expression Recognition by Canonical Correlation Analysis", *Proceedings of the IEEE 20th Signal Processing and Applications Conference (SIU)*, pp. 1–3, Mugla, Turkey, 2012.

134. Xing, E. P., R. Yan and A. G. Hauptmann, "Mining Associated Text and Images with Dual-wing Harmoniums", *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence (UAI 2005)*, pp. 633–641, AUAI Press, University of Edinburgh, Edinburgh, Scotland, 2005.

135. Nene, S., S. Nayar and H. Murase, *A Comparison of Methods for Multi-Class Support Vector Machines*, Tech. Rep. CUCS-006-96, Columbia University, 1996.

136. Vapnik, V. N., *The Nature of Statistical Learning Theory*, Springer-Verlag, New York, 1995.

137. Hsu, C. W. and C. J. Lin, "A Comparison of Methods for Multi-Class Support Vector Machines", *IEEE Transactions on Neural Networks*, Vol. 13, pp. 415–425, 2002.

138. Abney, S., "Bootstrapping", *Proceedings of the Fortieth Annual Conference of the Association for Computational Linguistics*, pp. 360–367, Association for Computational Linguistics, Philadelphia, Pennsylvania, 2002.

139. Nanuwa, S. and H. Seker, "Investigation into the Role of Sequence-Driven-Features for Prediction of Protein Structural Classes", *Proceedings of the 8th IEEE International Conference on Bioinformatics and Bioengineering*, pp. 1–6, IEEE, Athens, Greece, 2008.

140. Kurgan, L. A. and L. Homaeian, "Prediction of Structural Classes for Protein Sequences and Domains-Impact of Prediction Algorithms, Sequence Representation and Homology, and Test Procedures on Accuracy", *Pattern Recognition*, Vol. 39, No. 12, pp. 2323–2343, 2006.

141. Levitt, M. and C. Chothia, "Structural Patterns in Globular Proteins", *Nature*, Vol. 261, No. 5561, pp. 552–558, 1976.

142. Guvenir, H. A., B. Acar, G. Demiroz and A. Cekin, "A Supervised Machine Learning Algorithm for Arrhythmia Analysis", *Proceedings of the Computers in Cardiology Conference*, pp. 433–436, Lund, Sweden, 1997.

143. Khadra, L., A. S. Al-Fahoum and S. Binajjaj, "A Quantitative Analysis Approach for Cardiac Arrhythmia Classification Using Higher Order Spectral Techniques",

IEEE Transactions on Biomedical Engineering, Vol. 52, pp. 1840–1845, 2005.