UNIVERSITY OF SOUTHAMPTON

FACULTY OF ENGINEERING, SCIENCE AND MATHEMATICS
School of Electronics and Computer Science

Mutual Features for Pattern Classification

by

Heiko Claussen

Thesis for the degree of Doctor of Philosophy

June 2009

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING, SCIENCE AND MATHEMATICS
SCHOOL OF ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

MUTUAL FEATURES FOR PATTERN CLASSIFICATION

by Heiko Claussen

The mean of a data set is one trivial representation of data from one class. This thesis discusses mutual interdependence analysis (MIA) that is successfully used to extract more involved representations, or "mutual features", accounting for samples in the class. MIA aims to extract a common or mutual signature that is invariant to changes in the inputs. For example, a mutual feature is a speaker signature under varying channel conditions or a face signature under varying illumination conditions. By definition, the mutual feature is a linear combination of class examples that is equally correlated with all training samples in the class. An equivalent view is to find a direction to project the dataset such that projection lengths are maximally correlated. The MIA optimization criterion is presented from the perspectives of canonical correlation analysis and Bayesian estimation. This allows to state and solve the criterion for mutual features concisely and to infer other properties of its closed form, unique solution under various statistical assumptions. Moreover, a generalized MIA solution (GMIA) is defined that enables utilization of *a priori* knowledge. MIA and GMIA work well even if the mutual signature accounts only for a small part of the energy in the inputs. Real world problems do not exactly fit the signal model of an equally correlated common signature. Therefore, the behavior of MIA is analyzed in situations where its model does not exactly fit. For these situations it is shown that GMIA continues to extract meaningful information. Furthermore, the GMIA result is compared to ubiquitous signal processing methods. It is shown that GMIA extends these current tools visualizing previously hidden information. The utility of both MIA and GMIA is demonstrated on two standard pattern recognition problems: text–independent speaker verification and illumination–independent face recognition. For example, GMIA achieves an equal error rate (EER) of 4.0% in the text–independent speaker verification application on the full NTIMIT database of 630 speakers. On the other hand, for illumination–independent face recognition, MIA achieves an identification error rate of 7.4% in exhausive leave–one–out tests on the Yale database. Overall, MIA and GMIA are found to achieve competitive pattern classification performance to other modern algorithms.

Contents

List of Figures

List of Tables

Nomenclature

$\text{abs}\,(y_i)$	absolute value of y_i
$\boldsymbol{b}$	basis function
$\boldsymbol{B}$	set of basis functions
$\boldsymbol{c}$	weighting vector
$J(\boldsymbol{y})$	cost fof the variable $\boldsymbol{y}$
$\boldsymbol{C_x}$	covariance matrix of $\boldsymbol{x}$
$\text{cov}\,\{\boldsymbol{x}, \boldsymbol{y}\}$	covariance of $\boldsymbol{x}$ and $\boldsymbol{y}$
D	dimensionality of $\boldsymbol{x}$
$\boldsymbol{D}$	diagonal eigenvalue matrix
$\frac{\partial T(x)}{\partial x} = T'(x)$	derivative of $T(x)$ with respect to x
$\lvert\boldsymbol{W}\rvert$	determinant of the matrix $\boldsymbol{W}$
$\hat{\boldsymbol{w}}$	estimate of $\boldsymbol{w}$
λ_i	eigenvalue number i
$\boldsymbol{E}$	eigenvector matrix
$\boldsymbol{E}^i$	eigenvector number i
$\mathcal{E}(\boldsymbol{x})$	pooled energy of $\boldsymbol{x}$
E	speech excitation in the Fourier domain
$\hat{E}\,\{\boldsymbol{x}\}$	empirical expected value of $\boldsymbol{x}$
$\mathcal{F}$	Fourier transform
$\mathcal{F}^{-1}$	inverse Fourier transform
f	frequency in Hz
G	Glottal shaping filter in the Fourier domain
$H(\boldsymbol{y})$	entropy of $\boldsymbol{y}$
$\boldsymbol{H}$	image instance
$\boldsymbol{I}$	identity matrix
$I(y_1, y_2, ..., y_m)$	mutual information of the random variables y_i
$\boldsymbol{I}(\boldsymbol{a})$	Fisher information matrix for vector $\boldsymbol{a}$
$\mathcal{I}$	light source intensity
K	number of classes
$\boldsymbol{K}$	Kernel matrix
$K(P\,\|\|Q\,)$	Kullback–Leibler divergence between the probability distribution P and Q

kurt($\boldsymbol{x}$)	kurtosis of vector $\boldsymbol{x}$
$\vec{\boldsymbol{l}}$	vector of the light source direction
$L(\boldsymbol{B})$	likelihood of $\boldsymbol{B}$
L	lip radiation filter in the Fourier domain
$\mathcal{L}(\boldsymbol{B})$	logarithmic likelihood of $\boldsymbol{B}$
$\Phi(\boldsymbol{y})$	map of $\boldsymbol{y}$ to a kernel–defined feature space
MFCC	mel–frequency cepstral coefficient
M	mel–cepstral filter
m	mel–frequency
$\boldsymbol{\mu}$	sample mean vector
$\bar{x}$	mean of x
$\mathbb{N}^N$	natural number of dimensionality N
N	number of data samples
$\boldsymbol{n}$	noise vector
$\mathcal{N}(\mu, \sigma^2)$	normal distribution with mean μ and variance σ^2
$\vec{\boldsymbol{n}}$	normal vector of an object surface
$\underset{i}{\text{median}}(x_i)$	median over x_i
$\mathcal{N}(\boldsymbol{a}, \boldsymbol{C})$	multivariate normal distribution with mean curve $\boldsymbol{a}$ and covariance $\boldsymbol{C}$
$\mathcal{NULL}(\boldsymbol{Y})$	nullspace of matrix $\boldsymbol{Y}$
$O(N)$	computational complexity in the order of N (big O notation)
$\|\boldsymbol{w}\|$	norm of $\boldsymbol{w}$
$\underline{\mathbf{1}}$	vector of ones
$\underline{\underline{\mathbf{1}}}$	matrix of ones
$p(\boldsymbol{x}\|\boldsymbol{y})$	conditional probability of $\boldsymbol{x}$ given $\boldsymbol{y}$
$p_x(\cdot)$	probability density function of x
$p(\boldsymbol{X}, \boldsymbol{a})$	probability density function with $\boldsymbol{a}$ as a parameter
$P(k)$	probability of the occurrence of class k
$\boldsymbol{P}$	projection matrix
$\tilde{\boldsymbol{x}}$	projection of $\boldsymbol{x}$
$\mathbb{R}^N$	real number of dimensionality N
$\boldsymbol{s}$	source signal
$\boldsymbol{S}$	scatter matrix
$\tilde{\boldsymbol{S}}(\boldsymbol{X}\|\boldsymbol{w})$	scatter matrix of projected samples
S	speech signal in the Fourier domain
SV	voiced speech signal part in the Fourier domain
sign(x)	sign of x
$\mathcal{S}$	score
$\boldsymbol{\Gamma}$	Tikhonov matrix
tr($\boldsymbol{C}$)	trace of the matrix $\boldsymbol{C}$
$\boldsymbol{x}^T$	transposed of vector $\boldsymbol{x}$

$\mathcal{U}(\alpha, \beta)$	uniform distribution between the values α and β
$\sigma_{\boldsymbol{x}}^2$	variance of $\boldsymbol{x}$
var $\{\boldsymbol{x}\}$	variance of $\boldsymbol{x}$
V	vocal tract filter in the Fourier domain
$\boldsymbol{V}$	whitening matrix
$\boldsymbol{w}$	projecting direction
$\boldsymbol{z}$	whitened version of $\boldsymbol{x}$
$\boldsymbol{x}$	data vector of dimensionality D
$\boldsymbol{X}$	matrix whose columns are $\boldsymbol{x}_i$ with $i = 1, \ldots, N$
$\underline{\mathbf{0}}$	vector of zeros

Abbreviations and Acronyms

2D	Two dimensional
3D	Three dimensional
BGLM	Bayesian general linear model
BRDF	Bidirectional reflectance distribution function
cdf	Cumulative distribution function
CCA	Canonical correlation analysis
CRLB	Cramer-Rao lower bound
EER	Equal error rate
FAR	False acceptance rate
FFT	Fast Fourier transformation
FLDA	Fisher's linear discriminant analysis
FRR	False rejection rate
GMIA	General mutual interdependence analysis
HMM	Hidden Markov model
ICA	Independent component analysis
iid	Independent and identically distributed
IR	Identification rate
ISA	Independent subspace analysis
JND	Just–noticeable difference
KICA	Kernel independent component analysis
KLT	Karhunen–Loève transformation
KPCA	Kernel principal component analysis
KSVM	Kernel support vector machine
LDA	Linear discriminant analysis
LDC	Linguistic data consortium
LPC	Linear predictive coding
MSTAC	Modified short time auto correlation
MCA	Minor component analysis
MCCA	Modified canonical correlation analysis
MFCC	Mel–frequency cepstral coefficient
MIA	Mutual interdependence analysis
ML	Maximum likelihood

NIST	National institute of standards and technology
PCA	Principal component analysis
pdf	Probability density function
RR	Ridge regression
RV	Random vector
RVM	Relevance vector machine
SAD	Speech activity detection
STAC	Short time autocorrelation
SVM	Support vector machine
TICA	Topographical independent component analysis
TLS	Total least squares
UBM	universal background model
VQ	Vector quantization
XCA	Extreme component analysis

Declaration of Authorship

I, **HEIKO CLAUSSEN**, declare that the thesis entitled
MUTUAL FEATURES FOR PATTERN CLASSIFICATION and the work presented in the thesis are both my own, and have been generated by me as the result of my own original research. I confirm that:

- this work was done wholly or mainly while in candidature for a research degree at this University;
- where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
- where I have consulted the published work of others, this is always clearly attributed;
- where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
- I have acknowledged all main sources of help;
- where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
- parts of this work have been published as:
  - Claussen, H., Rosca, J., Damper, R., 2009. Generalized mutual interdependence analysis. *In: International Conference on Acoustics, Speech and Signal Processing.* Taipei, Taiwan, pp. 3317–3320.
  - Claussen, H., Rosca, J., Damper, R., 2008. Mutual features for robust identification and verification. *In: International Conference on Acoustics, Speech and Signal Processing.* Las Vegas, NV, pp. 1849-1852.
  - Claussen, H., Rosca, J., Damper, R., 2007. Mutual interdependence analysis. *In: Independent Component Analysis and Blind Signal Separation.* Springer–Verlag, Heidelberg, Germany, pp. 446-453.

Signed:

Date:

Acknowledgements

During the course of my Ph.D., I had often reason to be thankful for advice, discussions and support from many people. In particular, I would like to thank my supervisors Prof. Robert Damper (University of Southampton) and Dr. Justinian Rosca (Siemens Corporate Research) for their confidence, constructive feedback and enthusiasm in this work. I would like to express considerable gratitude for enabling many hours of technical discussions until late in the night and on weekends. This work would not have been possible without their continuous support.

Additionally, I would like to thank my family, friends and colleagues for their tips, comments and encouragements. Finally, I am grateful to Siemens Corporate Research for the financial support which provided the opportunity for this thesis.

To my parents:
Christine and Reimer Claussen

Chapter 1

Introduction

In modern applications, machines aim to automate or support processes that were solely performed by humans in the past. On one hand, the goal is to improve their speed, quality, reliability and reproducibility. On the other hand, such automations promise a reduction in costs and enable new products and services. Successful examples include optical character recognition for automatic sorting of mail or support of border control using biometric scanners. Reasons for the improvements using machines are their large memory, fast computational speed, low access and response times, exact timing, etc. However, while many tasks seem intuitive for the human, it is challenging to define the necessary signal processing procedures in a machine. For example, how to recognize a person on a gray scale image under different scaling, resolution, camera angles, illumination conditions, presence of occlusions etc.?

A first step to design a system that automates tasks which are successfully performed by a human is to model the human behavior, senses etc. To enable this, interdisciplinary knowledge from, e.g., natural sciences, psychology and the humanities, is utilized. For example, listeners were asked to define tones of equal distance in frequency. These experiments showed that the sensitivity of the human ear is nonlinear over frequency. Features that are based on these results are successfully used for speaker verification and audio compression. However, are behavioral studies sufficient to model complex and abstract tasks optimally? It can be assumed that a more general understanding of human thinking is necessary to describe initial human learning patterns leading ultimately to artificial intelligence. Moreover, this generalization can help to verify if the current process of feature extraction is feasible or in some way limiting when modeling complex tasks. In the following, these points are addressed in context to knowledge of objects, i.e., how can a machine detect, recognize or classify an object. The task of a machine that models the human behavior appears similar to the way a newborn human gains knowledge of the world. For before a newborn starts to speak, it is able to recognize familiar faces or objects. However, it is difficult to analyze this process scientifically because of ambiguity in the interpretation of responses and the missing ability to recall memories of this initial learning phase. One possible theory that describes this initial learning phase is given by:

> There can be no doubt that all our knowledge begins with experience. For how should our faculty of knowledge be awakened into action did not objects affecting our senses partly of themselves produce representations, partly arouse the activity of our understanding to compare these representations, and, by combining or separating them, work up the raw material of the sensible impressions into that knowledge of objects which is entitled experience? (Kant, 2003, p. 41, originally published 1781)

Following this hypothesis, the sensor data can be transformed into knowledge of objects therefore enabling a machine to perform abstract tasks. Note that this view aligns with current signal processing approaches used to model human behavior. Thus, the current way of modeling appears not to limit the automation of complex and abstract tasks. However, as the machine will not produce representations of objects itself, the challenge remains to define these features whose combination and correlations make up the raw material of knowledge on which decisions are based. It can be assumed that it is initially necessary to detect patterns before characteristic features can be defined. A possible definition of a pattern is a recurring structure, style or form of elements or events that contains information which can be used to predict or classify future appearances. For example, to learn the first language, a child needs to find links between words and objects, feelings, behaviors, etc. The goal is to find features that represent characteristic patterns in the data. This thesis focuses on the design and analysis of a novel set of features for pattern classification.

1.1 Problem Statement

There are multiple approaches and philosophies to define characteristic features. For example, one can manually define features based on behavioral experiments as discussed above. The disadvantage of such approaches is that they are not well adapted to new scenarios or data. That is, it is not feasible to define or detect features manually for every new input. Therefore, modern approaches tend toward an automatic extraction of features. Each of the methods for automatic feature extraction uses a different viewpoint and criteria to find its 'optimal' representation. Most pattern recognition problems implicitly assume that the number of observations is much higher than the dimensionality of each observation. This allows one to study characteristics of the distributional observations and design proper discriminant functions for classification. Generally, one can distinguish between supervised and unsupervised approaches. While *a priori* information about the class labels is available in the supervised case, unsupervised methods extract both features and class labels from the data.

Ubiquitous methods for automatic feature extraction include principal component analysis (PCA) (Pearson, 1901), independent component analysis (ICA) (Jutten and Herault, 1991), Fisher's linear discriminant analysis (FLDA) (Fisher, 1936), canonical correlation analysis (CCA) (Hotelling, 1936) etc. In the following, a sketch of these methods is given

to describe how each of them defines and captures the desired characteristics in the data. For instance, PCA defines directions in the input space as features of interest. The first direction is the one that maximizes the variance of the input data projections. The following directions are found in the same way but are also orthogonal to the previous ones. Thus, principal components capture the directions of maximum variance in the input set. Another view on characteristics in the data is taken by ICA. Rather than capturing directions of high variance in the data, ICA aims to find statistically independent representations. That is, the occurrence of one feature does not contain information about the state of another one. Note, that there is no measure used in ICA to evaluate the importance of each feature. Thus, one has additionally to find the feature affiliations between the training and testing set. Other approaches like FLDA use class label information aiming to extract characteristics that are maximally discriminative between classes. The empirical cost function of FLDA finds features that maximize the ratio of the between- and within-class scatter of the training data. Again another view is taken by CCA. Here, it is assumed that two datasets contain information about a common source. CCA finds pairs of projecting directions in both datasets that are maximally correlated. These directions are assumed to represent the characteristics of interest. A more detailed discussion of these and other related methods is given in chapters of the thesis.

Most available methods aim to extract features that capture differences, e.g., between classes. They suffer if the training data include various distortions. For example, in face recognition, most of the image variations are captured in differences of illumination conditions. Thus, by focusing on differences between the training instances, one focuses on the distortion rather than the information represented by the face. This illustrates that, e.g., the use of PCA is problematic in such applications, as discussed in a later chapter. Remaining with this example, one can imagine that pictures from dissimilar faces in the training set are taken in different illumination conditions, e.g., one person is photographed on a cloudy and the next one on a sunny day. The difference in illumination will be represented as most discriminant feature to distinguish these people using the FLDA approach. Clearly this is not desirable. Also, imagine the training set to include only a small number of classes. Is is sensible to detect these classes dependent on their differences that clearly can not capture a sufficient representation of the world? That is, if new unseen instances appear, will this system be able to classify them as unknown?

In these situations it appears meaningful to focus on the extraction of common or 'mutual' features from a given class. The question is if there exists such an invariant class representation or signature in the data and if it can be extracted. For example, assume a number of speech recordings from one speaker. Each of them is recorded over different nonlinearly distorted telephone channels. Is it possible to extract a feature that is equally present in all recordings? Clearly, such an invariant would be independent of the channel conditions in the training set. Assuming that this mutual feature represents the speaker, it can be used to detect him or her in future recordings even if these are distorted by unknown nonlinearities. The goal of this thesis is to find and analyze an invariant representation of high dimensional instances of a single class. These mutual features are evaluated in applications for pattern classification.

1.2 Challenges

Initially, it appears straightforward to extract a common pattern from a set of inputs. One may suggest to simply use the mean or median as this common representation. However, both of these representations have only limited capabilities to cancel distortion effects that are present in the inputs. Other possibilities are to approximate the mutual pattern in high dimensional representations by regression or curve fitting. This would enable the cancellation of noise and possibly some of the distortions. However, these approaches would also simplify the structure of the data, potentially canceling information. Clearly, there are many ways to approach this problem. Thus, the question remains how to define, measure and finally extract this desired commonality?

There are numerous feature extraction methods available that extract various characteristics from the data. As a starting point it is meaningful to evaluate their criteria for usability to extract mutual signatures. For example, one method that extracts invariance from the data is minor component analysis (MCA) (Xu et al., 1992) discussed later in the thesis. Is MCA already a good solution for our problem or are there issues when using it on high dimensional data? Even if no suitable solution for our problem is available, this search may unveil numerous measures of similarity supporting the design of a new approach.

It can be observed that methods are not only used because of their performance. Another important factor is the comprehensibility of the method itself and its underlying theory. For example, PCA is ubiquitously used because of its simplicity. Thus, users have a high confidence in its results and implementation. To enable a wide use, it is the goal to find a basic method for the extraction of mutual signatures that can be solved in closed form. Only after defining a plausible theory and simple solution does it appear meaningful to increase the complexity of the method enabling improvements of its performance.

In real world applications, it is common to cut long or continuous signals into segments using windowing functions. Gabor's uncertainty principle (Gabor, 1950) states that there exists a window size dependent trade–off between the frequency resolution and the time resolution of a signal. Therefore, the window size is usually selected dependent on the application. It appears meaningless to extract commonalities from infinitely small data segments. Naturally, a method to extract the mutual signature will be limited by the window choice. Thus, challenges are to analyze segmentation effects and to find limitations of the method designed for mutual feature extraction.

If it is true that the degree of commonality between inputs is dependent on their representation, e.g., window size, one can assume that the desired mutual signature is not equally present in all inputs. For instance, imagine that one input segment contains speech while another one is partly silence. It can be concluded that a larger amount of speaker dependent information is present in the first segment. This effect is expected for many applications. However, in this thesis it is assumed that a mutual signature is an invariant of a set of inputs. Clearly, one challenge is to

tackle problems where this model of similarity does not hold exactly.

The goal is to find a method to extract a mutual signature from high dimensional data. Note that many signal processing approaches suffer from the curse of dimensionality (Bellman, 1961) as discussed later in this thesis. In a nutshell, the curse of dimensionality points out that the number of samples necessary to estimate a distribution with a fixed error increases exponentially with the number of dimensions. Therefore, it is important to evaluate the properties of a high dimensional space. Is it possible to avoid this problem such that a limited number of instances suffice to extract a mutual signature?

Finally, it is important to test the designed feature extraction method and theory. In a first step, it is advantageous to show certain properties of the mutual signature extraction on synthetic data. However, the usefulness of a theory is finally tested by its application in the real world. The goal is to employ mutual features in multiple applications from different signal processing fields. In this way, the wide applicability of the approach is tested. However, it is expected that some standard, application dependent procedures can not be used because of differences in feature extraction philosophy. Thus it will be challenging to redesign the pre- and post–processing to enable a successful utilization of mutual features.

1.3 Contributions

This section gives a summary of selected contributions that originated from the work discussed in this dissertation. Their sequence is consistent with their appearance in the thesis.

- A novel algorithm called mutual interdependence analysis (MIA) is proposed to extract a common signature from a high dimensional dataset. Problems related to the high dimensional space are avoided by constraining the solution, e.g., to the span of the inputs (see Section 4.1).
- It is proved that there exists, up to its sign, a unique solution to the MIA problem for linearly independent inputs (see Section 4.1). In this case, the scatter onto the MIA result is zero. MIA captures an invariance in the data.
- It is shown that the mean subtracted space of linearly independent inputs does not include the mutual signature (see Sections 4.1 and 4.2). Thus, this common preprocessing step cancels a pattern in the data.
- The similarities of MIA with other related signal processing approaches, e.g., MCA (Introduction of Chapter 4), Bayesian estimation (Section 4.4), linear discriminant analysis (LDA) and CCA (Section 4.2), are evaluated. Each brings insight to the properties of MIA. Selected properties are defined in corollary 4.3–4.6.
- The MIA approach is generalized to account for variability in the MIA assumptions, for example, if the common signature is not equally present in the inputs. The new

method is called generalized MIA (GMIA). A dual form of the GMIA algorithm is presented to enable efficient computation of problems with large number of inputs or high dimensionality. An iterative method is sketched to find a mutual signature for high dimensional problems with large number of inputs (see Section 4.4).

- It is shown that MIA finds patterns that are hidden to ubiquitous signal processing methods. It is verified that MIA extracts a mutual signature from synthetic data. Moreover, it is demonstrated when and how MIA, GMIA or the mean represent a common signature in the inputs (see Section 5.1). It is illustrated on synthetically mixed face images that MIA can be used to extract illumination invariant representations from two–dimensional data (see Section 5.2).

- A successful application of GMIA for text–independent speaker verification is shown in Chapter 6. GMIA achieves an equal error rate (EER) of 4.0% using the full NTIMIT database of 630 speakers. A higher–order statistics based speech activity detector is implemented (see Section 6.4.1). Moreover, a modified short time autocorrelation (MSTAC) approach is defined to increase the quality of voiced speech extraction (see Section 6.4.2).

- MIA is employed for illumination–independent face recognition in Chapter 7. The mutual face method achieves an error rate of 7.4% in exhaustive leave–one–out tests on cropped images of the Yale face database. MIA achieves comparable or better results than standard methods.

The discussed work in this thesis resulted in the following publications:

- Claussen, H., Rosca, J., Damper, R., 2009. Signature extraction using mutual interdependencies. *Pattern Recognition.* In review.

- Claussen, H., Rosca, J., Damper, R., 2009. Generalized mutual interdependence analysis. *In: International Conference on Acoustics, Speech and Signal Processing.* Taipei, Taiwan, pp. 3317–3320.

- Claussen, H., Rosca, J., Damper, R., 2008. Mutual features for robust identification and verification. *In: International Conference on Acoustics, Speech and Signal Processing.* Las Vegas, NV, pp. 1849-1852.

- Claussen, H., Rosca, J., Damper, R., 2007. Mutual interdependence analysis. *In: Independent Component Analysis and Blind Signal Separation.* Springer–Verlag, Heidelberg, Germany, pp. 446-453.

1.4 Thesis Outline

Here, the structure of the thesis is described providing the reader with an overview of each chapter. Goals of this section are also to show the interconnection between chapters and to

provide the reader familiar with the subject the possibility to focus on his/her points of interest.

Chapter 1 motivates the concept of mutual features for pattern classification. Characteristic class information is often not captured by highly variable signal components. For example, in face recognition, class independent information like illumination can contribute more to the signal variation than the edges of local features like mouth, eyes, nose etc. A representation that is invariant to the changes in the training data is assumed to be invariant to unknown changes in the inputs. Therefore, such a mutual signature is expected to be robust to distortions like differences in illumination representing a powerful feature for classification. Moreover, this chapter discusses challenges related to finding this mutual signature and provides a list of selected contributions of work presented in this thesis.

Chapter 2 provides a description of approaches that are either related to the method discussed in this thesis or ubiquitously used in classification and feature extraction tasks. The goals of this chapter are to give an overview of related concepts and to show their interconnection. This illustrates how the proposed mutual signature approach extends the present landscape of feature extraction methods. Additionally, the overview provides a foundation for demonstrating similarities between the mutual signatures based and ubiquitous methods later in this thesis. This analysis aids the understanding of mutual features for pattern classification visualizing its properties and leading to a generalization of the approach. Readers can skip this section if they are familiar with PCA, MCA, CCA, Bayesian estimation, Gaussian mixture models as well as FLDA and its similarity to CCA.

Chapter 3 discusses selected higher–order and nonlinear methods. This gives a new perspective to feature extraction problems. Described concepts are used in later chapters, e.g., a kurtosis–based feature for voice activity detection. Moreover, this chapter motivates pattern analysis in high dimensional spaces. That is, by viewing a problem in a high dimensional space it is possible to use linear methods to solve nonlinear problems in its lower dimensional representation, e.g., the dual space. Readers that are familiar with these concepts can skip this section.

Chapter 4 gives a detailed motivation to the extraction of mutual features for pattern classification. Thereafter, an algorithm called mutual interdependence analysis (MIA) is proposed. It is proved that this problem has, up to its sign, a unique solution. Subsequently, MIA is reinterpreted from the point of view of a modified CCA problem. Properties of MIA are analyzed showing, e.g., when the MIA criterion has a defined solution or is equivalent to the sample mean. A regularization of MIA is proposed to enable matrix inversions if inputs are nearly collinear. In real world applications, it may not be appropriate to assume a common signature to be equally present in the inputs. That is, for speaker verification it can be expected that an input instance containing mostly silence captures less speaker characteristic information than one without pauses. This problem is addressed by the introduction of a generalized MIA (GMIA) criterion motivated from a Bayesian point of view. The generalization enables the use of prior knowledge regarding the model misfit and the mutual signature. For instance, it is possible to provide an expectation of this signature. A dual GMIA solution is given to tackle problems

of either high dimensionality or large number of inputs. MIA can be solved in closed form by constraining the space of solutions. This chapter demonstrates how to change the constraint from the span of the inputs to an alternate function space.

Chapter 5 evaluates MIA and GMIA on synthetically generated data. It is shown that MIA extracts an invariant in the data given the synthetic model. Furthermore, GMIA extracts a common signature that is hidden to ubiquitous methods such as PCA, ICA and the mean. This chapter shows results of a statistical experiment that demonstrates when MIA, GMIA or the mean are preferable. For example, MIA and GMIA are effective if the common signature represents a small part of the signal energy. Furthermore GMIA is preferable over MIA if the assumption of equal presence of the mutual signature in the inputs does not hold. The application of MIA is shown on both one- and two–dimensional data. Finally, differently illuminated face images are generated. It is demonstrated that MIA extracts an illumination invariant 'mutual face'.

Chapter 6 discusses the implementation of a GMIA–based method for text–independent speaker verification on challenging data. Common methods and features for speaker verification are discussed providing a background of the field. Thereafter, the algorithms used for preprocessing, e.g., for voice activity detection and voiced speech detection, are described. It is discussed how GMIA signatures are extracted from each class. In the following, the selected background model is specified. It estimates the quality of the input by computing the highest score with all learned signatures. The decision on acceptance is dependent on the difference between this high score and the score with the signature of the claimed identity. This implementation achieves equal error rates of 4.0% on the full NTIMIT database of 630 speakers.

Chapter 7 describes the implementation of a MIA–based method for illumination–robust face recognition. The Yale database was selected providing challenging data including variation in illumination conditions, facial expressions, occlusions and misalignment. Semi–automatic preprocessing of this data is discussed. Thereafter, the extraction of the mutual faces as well as the computation of similarity scores are described. Mutual faces for illumination–robust face recognition achieve an identification error rate of 7.4% in exhaustive leave–one–out tests on the Yale database.

Chapter 8 concludes the thesis by first reminding the reader of the goals and challenges followed by a discussion on how these points were addressed.

Chapter 9 describes promising areas of further work. For example, links to kernel support vector machines are suggested that could lead to new MIA applications and further understanding. The applications for text–independent speaker verification and illumination–robust face recognition are based on multiple assumptions. This chapter pinpoints simplified models and implementations that may limit the current results.

Chapter 2

Related Work

This chapter provides a basis of concepts used in prominent algorithms for statistical signal processing, feature extraction and classification. Furthermore, the interconnectedness of the discussed algorithms is illustrated enabling comparison of their properties and assumptions. These concepts build the foundation for new and improved algorithmic approaches as well as their analysis and advancement discussed in later chapters of this thesis.

Section 2.1 discusses principal component analysis (PCA) as well as its application in the Karhunen–Loève transformation and whitening. Because of its popularity, PCA is used as an entry point into the discussion of algorithmic concepts. Thereafter, Section 2.2 introduces the alternative minor component analysis (MCA) method. Section 2.3 shows how the PCA and MCA concepts can be combined in extreme component analysis (XCA) resulting in possibly better data representations. In the following, Section 2.4 discusses the Bayesian inspired linear discriminant analysis (LDA) method for classification. Section 2.5 introduces Fisher's linear discriminant analysis (FLDA) and illustrates the conditions under which the results with its empirical cost function are equivalent to LDA. Thereafter, in Section 2.6 canonical correlation analysis (CCA) is discussed. It is shown that CCA results in FLDA if the classification table is used as second input. To demonstrate similarities to the regression domain, Section 2.7 analyses similarities of least squares estimation (LSE), total least squares estimation (TLS) and ridge regression (RR). Section 2.8 uses the Bayesian general linear model (BGLM) to derive RR from a Bayesian perspective and to illustrate the algorithmic incorporation of prior knowledge. Thereafter, Section 2.9 discusses the design, training and use of Gaussian mixture models (GMM). Section 2.10 completes this chapter with its summary.

2.1 Principal Component Analysis

Principal component analysis (PCA) (Pearson, 1901) is a popular technique for feature extraction. Originally, PCA was designed to find a hyperplane that fits a cloud of N points $\boldsymbol{X} \in \mathbb{R}^{D \times N}$ of dimensionality D with a minimum mean square error. Here, the principal

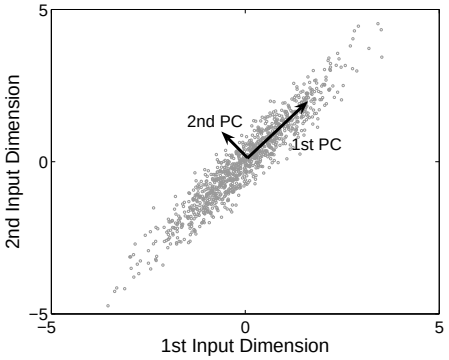


FIGURE 2.1: Geometrical interpretation of PCA: The principal components represent a set of orthogonal directions that point toward the maximum data variance.

components $\boldsymbol{w}_i \in \mathbb{R}^D$ are defined as orthogonal directions that span this hyperplane. That is, the first principal component represents a one–dimensional hyperplane that fits the data with minimum mean square error, the second principal component satisfies this criterion in the orthogonal subspace etc. The name 'principal component' is deduced from the principal axis of an elliptic input data cloud that are found by this method. The cost function of PCA is given by:

$$\hat{\boldsymbol{w}}_{\mathrm{PCA}} = \underset{\boldsymbol{w},\|\boldsymbol{w}\|=1}{\arg\max} \frac{\boldsymbol{w}^T \cdot \boldsymbol{X} \cdot \boldsymbol{X}^T \cdot \boldsymbol{w}}{\boldsymbol{w}^T \cdot \boldsymbol{w}}$$

A geometrical interpretation of PCA is shown in Figure 2.1. The first principal component (PC) $\boldsymbol{w}_1$ points in the direction of the maximal input data variance. The second principal component points in the direction of the maximal variance in the orthogonal subspace $(\boldsymbol{I} - \boldsymbol{w}_1 \cdot \boldsymbol{w}_1^T) \cdot \boldsymbol{X}$ to the first principal component and so on. Hence, a subset of the first N principal components spans the N–dimensional space that retains a maximum of the input data variance. This property of PCA is utilised for data compression in the Karhunen–Loève transformation (Section 2.1.1). Because of the orthogonality of the principal component directions, the input data projections on them are uncorrelated. Therefore, the covariance matrix of the points in the projected input space is diagonal. Consequently, PCA can be described by an eigenvalue decomposition diagonalising the covariance matrix of the input data. Thus, the eigenvector with the largest eigenvalue corresponds to the first principal component, the one with the second largest eigenvalue to the second principal component, etc. An important preprocessing step of PCA is mean input subtraction. Because of their relation to PCA, multiple methods including minor component analysis (MCA) rely on this preprocessing. The problem that this preprocessing step leads to is discussed in Section 4.1.

Because of correlations, the resulting transfer of the inputs to another coordinate system by PCA might already lead to a dimensionality reduction. In this case, where the data can be fully described in a lower dimensional subspace, the dimensionality reduction is equivalent to a lossless data compression.

2.1.1 Karhunen–Loève Transformation

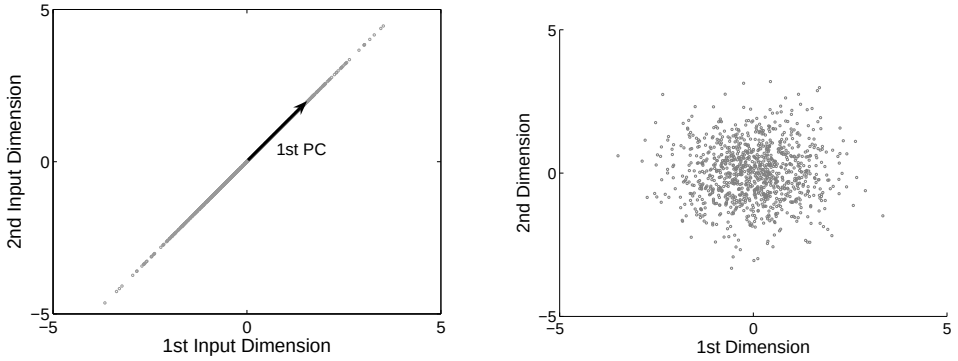
In the literature, the term PCA is interchangeable with Hotelling (Hotelling, 1933) and Karhunen–Loève transformation (KLT) (Karhunen, 1947). In this thesis, the approach of using principal components for dimensionality reduction is referred to as Karhunen–Loève transformation although the idea was initially introduced by Hotelling. The reason for this is the generalisation of the theory by Karhunen and Loève. In contrast to the early work of Pearson and Hotelling, Karhunen and Loève represent a stochastic process as an infinite linear combination of orthogonal random functions. Here, the basis is dependent on the inputs. This is unlike the Fourier transform where the basis is constrained to sinusoidal functions. In the following, the discrete case is also denoted KLT despite its original definition in the infinite dimensional space.

The goal of this transformation is to find the set of N vectors $\hat{\boldsymbol{x}}_j \in \mathbb{R}^D$ that represent the original data $\boldsymbol{x}_j \in \mathbb{R}^D$ while spanning a lower, d–dimensional space. This is achieved by a projection of the original data onto the set of orthonormal directions $\boldsymbol{w}_i \in \mathbb{R}^D$ with $i = 1, \ldots, d$ that are found to minimise the mean square error $\hat{E}\left\{\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|^2\right\}$. The projection can be formalised as $\hat{\boldsymbol{x}} = \sum_{i=1}^{d} \left(\boldsymbol{w}_i^T \cdot \boldsymbol{x}\right) \boldsymbol{w}_i$.

Figure 2.2(a) illustrates an example of the KLT for $d = 1$ given the input data from Figure 2.1. A constraint of the KLT is the orthogonality of the data elements in the resulting space and $\|\boldsymbol{w}\| = 1$. Hence, $\sum_{j=1}^{N} \hat{x}_{ij}\hat{x}_{kj} = 0, \forall i \neq k$. Because of the orthogonality constraint, the KLT results in an uncorrelated representation of the data. It is assumed that uncorrelation minimises 'redundancy' of the data representation and that the 'information content' of each data vector is dependent on its variance. Therefore, by representing the data with an uncorrelated basis while disregarding directions of minimum variance, the data are compressed with a minimum loss of information. This assumption can be equated with the approach where the data are represented by projections on the eigenvectors of its covariance matrix. The eigenvectors with the highest eigenvalues represent the directions of maximum variance. Thus, the lower dimensional result, with the minimum mean square error to the original data, can be found in the case only eigenvectors with the lowest eigenvalues are disregarded. For illustration of this approach, the eigenvectors $\boldsymbol{E}^i$ are assumed to be sorted by their corresponding eigenvalues. The concept of the Karhunen–Loève transformation is shown in:

$$
\begin{aligned}
\boldsymbol{x} &= \sum_{i=1}^{D} \left(\boldsymbol{x}^T \cdot \boldsymbol{E}^i\right) \cdot \boldsymbol{E}^i = \hat{\boldsymbol{x}} + \boldsymbol{e} \\
\Rightarrow \hat{\boldsymbol{x}} &= \sum_{i=1}^{d} \left(\boldsymbol{x}^T \cdot \boldsymbol{E}^i\right) \cdot \boldsymbol{E}^i \quad ; \quad \boldsymbol{e} = \sum_{i=d+1}^{D} \left(\boldsymbol{x}^T \cdot \boldsymbol{E}^i\right) \cdot \boldsymbol{E}^i
\end{aligned}
$$

While $\hat{\boldsymbol{x}}$ represents the compressed data in the direction of the d eigenvectors $\boldsymbol{E}^i$ with the largest eigenvalues, $\boldsymbol{e}$ represents the data parts which are lost in the compression process.



(a) Karhunen–Loève transformation (b) Whitened input space

FIGURE 2.2: Demonstration of the KLT and whitening. (a) The KLT compresses data by projection onto the first d principal components. (b) Whitening cancels all second order information of a dataset by scaling the principal component directions to unit variance. Note that there exist no distinct principal component directions in the whitened space.

2.1.2 Whitening

The whitening procedure (Hyvärinen et al., 2001) is an important preprocessing step for signal processing algorithms that are based on higher–order statistics such as independent component analysis (ICA) (see Section 3.1). Therefore, this section shows the definition and some features of this method. A random vector (RV) $\boldsymbol{z}$ is called whitened if its elements are zero–mean, uncorrelated and of unit variance. Figure 2.2(b) shows the result of a whitening procedure applied to the data in Figure 2.1. The transfer of a vector $\boldsymbol{x}$ to the whitened space is achieved by multiplication with a so–called whitening matrix $\boldsymbol{V}$:

$$\boldsymbol{z} = \boldsymbol{V} \cdot \boldsymbol{x}$$

PCA and whitening aim to find an uncorrelated representation of a dataset. Thus, both methods solve the eigenvalue decomposition of the sample covariance matrix $\boldsymbol{C_{XX}} = \hat{E}\left\{\boldsymbol{x} \cdot \boldsymbol{x}^T\right\}$. In the following let $\boldsymbol{D}$ and $\boldsymbol{E}$ represent the diagonal eigenvalue and eigenvector matrix of $\boldsymbol{C_{XX}}$ respectively. Hence, the eigenvalue decomposition of $\boldsymbol{C_{XX}}$ is given by:

$$\hat{E}\left\{\boldsymbol{x} \cdot \boldsymbol{x}^T\right\} = \boldsymbol{E} \cdot \boldsymbol{D} \cdot \boldsymbol{E}^T$$

Note that the covariance matrix of a whitened vector results in the identity matrix. While the unit variance constrained results in ones at the diagonal elements of its covariance matrix, the uncorrelation of the different vector elements forces all off–diagonal values to zero. Using these constraints, a whitening matrix $\boldsymbol{V}$ can be derived as follows:

$$\begin{aligned}\hat{E}\left\{\boldsymbol{z}\cdot\boldsymbol{z}^T\right\} &= \boldsymbol{I}\\ &= \boldsymbol{V}\cdot\hat{E}\left\{\boldsymbol{x}\cdot\boldsymbol{x}^T\right\}\cdot\boldsymbol{V}^T\\ &= \boldsymbol{D}^{-0.5}\cdot\boldsymbol{E}^T\cdot\boldsymbol{E}\cdot\boldsymbol{D}\cdot\boldsymbol{E}^T\cdot\boldsymbol{E}\cdot\boldsymbol{D}^{-0.5}\\ \Rightarrow\boldsymbol{V} &= \boldsymbol{D}^{-0.5}\cdot\boldsymbol{E}\end{aligned}$$

The resulting matrix $\boldsymbol{V}$ transfers $\boldsymbol{E}\cdot\boldsymbol{D}\cdot\boldsymbol{E}^T$, by multiplication from both sides, into the identity matrix. There are no restrictions concerning the statistical properties of the variables that are whitened or processed by PCA. However, the first- and second–order statistics of the data have to be known or measurable.

2.2 Minor Component Analysis

In contrast to PCA, minor component analysis (MCA) aims to find directions $\boldsymbol{w}_i \in \mathbb{R}^D$ of minimum variance in the data $\boldsymbol{X} \in \mathbb{R}^{D\times N}$. It is hypothesised that important information of a system is present in directions where the output data are constrained. This concept will be of importance in later chapters of this thesis. MCA is used for spectral estimation, curve and hyper–surface fitting, cognitive perception and computer vision. Xu et al. (1992) coined the name minor component analysis to refer to neural network fitting methods that compute minor components. The authors also suggested that their iterative MCA algorithm solves the total least squares (TLS) problem discussed in Section 2.7.1. The MCA criterion is given by:

$$\hat{\boldsymbol{w}}_{\mathrm{MCA}} = \underset{\boldsymbol{w},\|\boldsymbol{w}\|=1}{\arg\min}\,\frac{\boldsymbol{w}^T\cdot\boldsymbol{X}\cdot\boldsymbol{X}^T\cdot\boldsymbol{w}}{\boldsymbol{w}^T\cdot\boldsymbol{w}} \tag{2.1}$$

If the rank of the covariance matrix $\boldsymbol{C}_{\boldsymbol{XX}} = \boldsymbol{X}\cdot\boldsymbol{X}^T$ of the zero mean inputs is smaller than $D-1$, there exists no unique solution to equation (2.1). Note that MCA minimises the Rayleigh quotient $r = \frac{\boldsymbol{w}^T\cdot\boldsymbol{C}_{\boldsymbol{XX}}\cdot\boldsymbol{w}}{\boldsymbol{w}^T\cdot\boldsymbol{w}}$. Thus, MCA finds the smallest eigenvalue/eigenvector pair of $\boldsymbol{C}_{\boldsymbol{XX}}$. An iterative solution to equation (2.1) is suggested in Xu et al. (1992) using the derivative of the Rayleigh quotient $\frac{\partial r}{\partial \boldsymbol{w}} = \frac{2}{\boldsymbol{w}^T\cdot\boldsymbol{w}}\left(\boldsymbol{C}_{\boldsymbol{XX}}\cdot\boldsymbol{w} - \frac{\boldsymbol{w}^T\cdot\boldsymbol{C}_{\boldsymbol{XX}}\cdot\boldsymbol{w}\cdot\boldsymbol{w}}{\boldsymbol{w}^T\cdot\boldsymbol{w}}\right)$ as cost function:

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \alpha\left(\boldsymbol{C}_{\boldsymbol{XX}}\cdot\boldsymbol{w}_t - \frac{\boldsymbol{w}_t^T\cdot\boldsymbol{C}_{\boldsymbol{XX}}\cdot\boldsymbol{w}_t\cdot\boldsymbol{w}_t}{\boldsymbol{w}_t^T\cdot\boldsymbol{w}_t}\right)$$

Note that the learning rate α influences the convergence of the algorithm. After each iteration, the resulting vector is normalised: $\boldsymbol{w}_{t+1} = \frac{\boldsymbol{w}_{t+1}}{\|\boldsymbol{w}_{t+1}\|}$. Additional minor components can be found in orthogonal input spaces using a Gram–Schmidt–like deflation process. For example, the second minor component uses $\boldsymbol{X}_2 = (\boldsymbol{I} - \boldsymbol{w}_1\cdot\boldsymbol{w}_1^T)\cdot\boldsymbol{X}$ as the input space of the algorithm. Figure 2.3 illustrates the MCA method geometrically given the dataset of Figure 2.1. Note

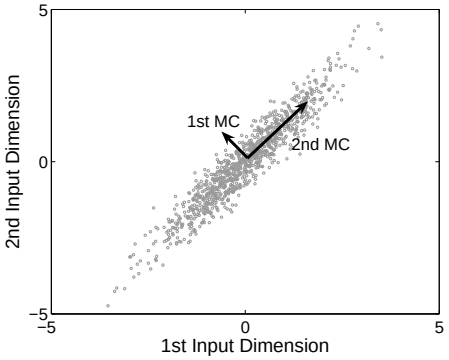


FIGURE 2.3: Geometrical interpretation of MCA: The minor components represent a set of orthogonal directions that point toward the minimum data variance.

that the minor component directions are equivalent to the principal component directions in reverse order (see for illustration Figure 2.1). Further information about MCA, including its probabilistic model, can be found in Thompson (1979), Oja (1992) and Williams and Agakov (2002).

2.3 Extreme Component Analysis

The goal of extreme component analysis (XCA) (Welling et al., 2004) is to combine the concepts of PCA and MCA to find a maximum likelihood representation of the training data $\boldsymbol{X} \in \mathbb{R}^{D \times N}$ given a fixed number of d components. PCA (Section 2.1) models the data as a signal embedded in background noise. It is assumed that the d directions of maximum variance represent most information while the disregarded $D - d$ dimensions contain random Gaussian noise. On the other hand, MCA (Section 2.2) assumes a high variance background model with the d constrained directions of minimum variance representing the important information of the data. XCA assumes that all directions of extreme variance contain information about inherent data structure. Therefore, a combination of principal and minor components is assumed optimal. Welling et al. (2004) shows that the extreme components of a dataset are principal components if the ordered eigenvalues become invariant after some value. This situation is referred to as log–convex spectrum. However, the extreme components are minor components if the large eigenvalues are invariant while the lower eigenvalues show structure (referred to as log–concave spectrum). This suggests that a dataset with a constant plateau between its high and low eigenvalues can be better modelled by XCA than by either PCA or MCA.

The method proposed by Welling et al. (2004) to find d extreme components can be described as follows. First, d principal and d minor components are extracted from the data. In a second step, subsets $\boldsymbol{g}$ of d eigenvalues from mixed PCA/MCA eigenvector combinations are evaluated using the XCA cost function:

$$J(\boldsymbol{g}) = \sum_{i \in \boldsymbol{g}} \log \lambda_i^2 + (D - d)\, \log \left(\operatorname{trace}\left(\boldsymbol{C_{XX}}\right) - \sum_{i \in \boldsymbol{g}} \log \lambda_i^2 \right) \tag{2.2}$$

Here, $\boldsymbol{C_{XX}}$ and λ represent the covariance matrix of the inputs and its selected eigenvalues respectively. Note that because PCA and MCA assume the eigenvalues of the discriminant vectors to be contiguous, only d compositions of $\boldsymbol{g}$ have to be evaluated. The eigenvectors that correspond to the combination of eigenvalues yielding the lowest cost in equation (2.2) are selected as extreme components of the dataset.

2.4 Linear Discriminant Analysis

In contrast to the previously–introduced methods that focus on feature extraction and dimensionality reduction, linear discriminant analysis (LDA) (Duda and Hart, 1973, p. 130) is mainly used for classification. The LDA model assumes the probability distributions $p(\boldsymbol{x}|k)$ with $k = 1, \ldots, K$ of all K classes to be Gaussian with mean $\boldsymbol{\mu}^{(k)}$ and equal covariance $\boldsymbol{C}$. Therefore, it is possible to find linear boundaries between classes that describe equal posterior probabilities. In this thesis, this simple classification approach is used as introduction to the following sections. Let $P(k)$ denote the probability of the occurrence of class k. Utilising Bayes theorem (Bayes, 1763) one can find the posterior probability of class k as:

$$p(k|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|k) \cdot P(k)}{\sum_{l=1}^{K} p(\boldsymbol{x}|l) \cdot P(l)} \tag{2.3}$$

By assuming

$$p(\boldsymbol{x}|k) = \frac{1}{\sqrt{(2 \cdot \pi)^D \cdot |\boldsymbol{C}|}} \cdot \exp\left\{ -\frac{1}{2} \cdot \left(\boldsymbol{x} - \boldsymbol{\mu}^{(k)}\right)^T \cdot \boldsymbol{C}^{-1} \cdot \left(\boldsymbol{x} - \boldsymbol{\mu}^{(k)}\right) \right\} \tag{2.4}$$

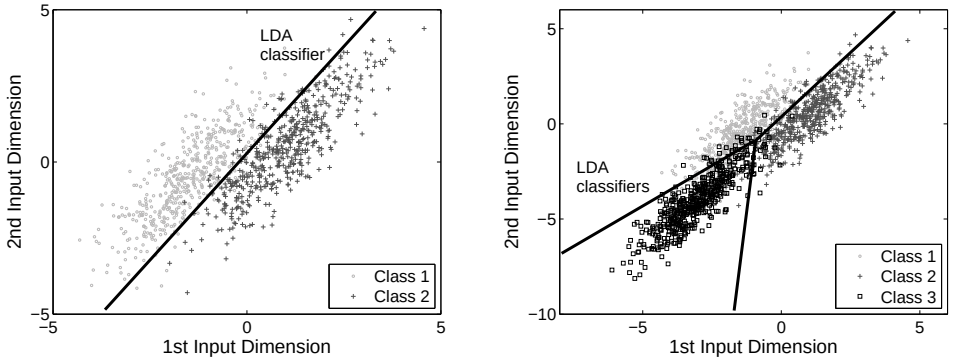
one can locate the separating hyperplane of classes k and l where $\boldsymbol{x}$ satisfies:

$$\log \frac{p(k|\boldsymbol{x})}{p(l|\boldsymbol{x})} = 0 \tag{2.5}$$

Therefore, the sign of $\log \frac{p(k|\boldsymbol{x})}{p(l|\boldsymbol{x})}$ can be used for classification. A positive result represents a higher probability that $\boldsymbol{x}$ is member of class k than class l. An example of a two–class LDA problem is illustrated in Figure 2.4(a). By inserting equations (2.3) and (2.4) in (2.5), one can find the following logarithmic likelihood ratio test for classification:

$$\left(\boldsymbol{\mu}^{(k)} - \boldsymbol{\mu}^{(l)}\right)^T \cdot \boldsymbol{C}^{-1} \cdot \boldsymbol{x} \underset{l}{\overset{k}{\gtrless}} \frac{1}{2} \cdot \left(\left(\boldsymbol{\mu}^{(k)} + \boldsymbol{\mu}^{(l)}\right)^T \cdot \boldsymbol{C}^{-1} \cdot \left(\boldsymbol{\mu}^{(k)} - \boldsymbol{\mu}^{(l)}\right) \right) + \log \frac{P(l)}{P(k)} \tag{2.6}$$

By performing K similar tests, LDA can find the most probable class membership of a new data point $\boldsymbol{x}$. An example of LDA in multi–class problems is shown in Figure 2.4(b).



(a) 2 class LDA problem (b) 3 class LDA problem

FIGURE 2.4: Classification with LDA. (a) LDA finds the linear function of equal posterior class probability between two identically Gaussian distributed classes. (b) For multi–class problems, LDA estimates pairwise linear boundaries between the classes that contribute section wise to the global classifier.

A more general solution to this classification problem is given by quadratic discriminant analysis (QDA) (Hastie et al., 2001). Here, the covariance matrices of the Gaussian priors are not assumed to be equal.

2.5 Fisher's Linear Discriminant Analysis

Fisher's linear discriminant analysis (FLDA) (Fisher, 1936) is a prominent empirical extension to the Bayesian–motivated LDA in Section 2.4. In contrast to LDA, FLDA does not assume prior knowledge of the class probability distributions. However, it effectively uses the intuition that members of one class are well represented in a space where they have small variance. This classification concept will be of importance in Chapter 4. Originally, FLDA was defined for the case of two classes. The goal is to find a linear function that 'optimally' discriminates between them. Figure 2.5 illustrates this concept by comparing class separability using FLDA with a PCA based approach. In the following, we will also refer to its natural multi–class generalisation as FLDA and denote $N^{(k)}$ to be the number of samples in class k. A prominent application of FLDA is the Fisher face approach (Belhumeur et al., 1997) for face recognition. The empirical cost function of FLDA is given by:

$$\hat{\boldsymbol{W}}_{\mathrm{FLDA}} = \arg\max_{\boldsymbol{W}} \frac{|\boldsymbol{W} \cdot \boldsymbol{S}_B \cdot \boldsymbol{W}|}{|\boldsymbol{W} \cdot \boldsymbol{S}_W \cdot \boldsymbol{W}|} \tag{2.7}$$

Its goal is to find a projection $\boldsymbol{W}$ that maximises the quotient of the determinants of the between–class scatter matrix $\boldsymbol{S}_B = \sum_{k=1}^{K} N^{(k)} \cdot \left(\boldsymbol{\mu}^{(k)} - \boldsymbol{\mu}\right) \cdot \left(\boldsymbol{\mu}^{(k)} - \boldsymbol{\mu}\right)^T$ and the within–class scatter matrix $\boldsymbol{S}_W = \sum_{k=1}^{K} \boldsymbol{S}^{(k)} = \sum_{k=1}^{K} \sum_{\boldsymbol{x}_i \in \boldsymbol{X}^{(k)}} \left(\boldsymbol{x}_i - \boldsymbol{\mu}^{(k)}\right) \cdot \left(\boldsymbol{x}_i - \boldsymbol{\mu}^{(k)}\right)^T$. This problem is known as generalised Rayleigh quotient and can be solved by the generalised

eigenvector problem:

$$\boldsymbol{S}_B \cdot \boldsymbol{w}_i = \lambda_i \cdot \boldsymbol{S}_W \cdot \boldsymbol{w}_i \tag{2.8}$$

The resulting generalized eigenvectors $\boldsymbol{w}_i$ form the columns of the projection matrix $\boldsymbol{W}$. Similarly to the Karhunen–Loève transformation (Section 2.1.1, Karhunen 1947), one can further reduce the dimensionality of the projections $\tilde{\boldsymbol{x}} = \boldsymbol{W} \cdot \boldsymbol{x}$ by disregarding the generalized eigenvectors with the lowest eigenvalues. This method is called reduced-rank LDA. Classification using Fisher's LDA is done in this projected space. The Fisher and Bayesian inspired LDA versions are equivalent for two–class problems of Gaussian variables with equal prior probabilities $P(k)$ and equal, invertible covariance matrices $\boldsymbol{C} = \boldsymbol{C}^{(k)} = \boldsymbol{C}^{(l)}$. In this case, Fisher's LDA can be solved by the simple eigenvector problem: $\boldsymbol{S}_W^{-1} \cdot \boldsymbol{S}_B \cdot \boldsymbol{w} = \lambda \cdot \boldsymbol{w}$. By substituting the previously introduced definitions of the scatter matrices $\boldsymbol{S}_W$ and $\boldsymbol{S}_B$ into this eigenvector problem we obtain:

$$\left(\sum_{k=1}^{2} N^{(k)} \cdot \boldsymbol{C}^{(k)} \right)^{-1} \cdot \left(\sum_{k=1}^{2} N^{(k)} \cdot \left(\boldsymbol{\mu}^{(k)} - \boldsymbol{\mu} \right) \cdot \left(\boldsymbol{\mu}^{(k)} - \boldsymbol{\mu} \right)^T \right) \cdot \boldsymbol{w} = \lambda \cdot \boldsymbol{w}$$

This can be further simplified to:

$$\frac{1}{4} \cdot \boldsymbol{C}^{-1} \cdot \left(\boldsymbol{\mu}^{(k)} - \boldsymbol{\mu}^{(l)} \right) \cdot \left(\boldsymbol{\mu}^{(k)} - \boldsymbol{\mu}^{(l)} \right)^T \cdot \boldsymbol{w} = \lambda \cdot \boldsymbol{w} \tag{2.9}$$

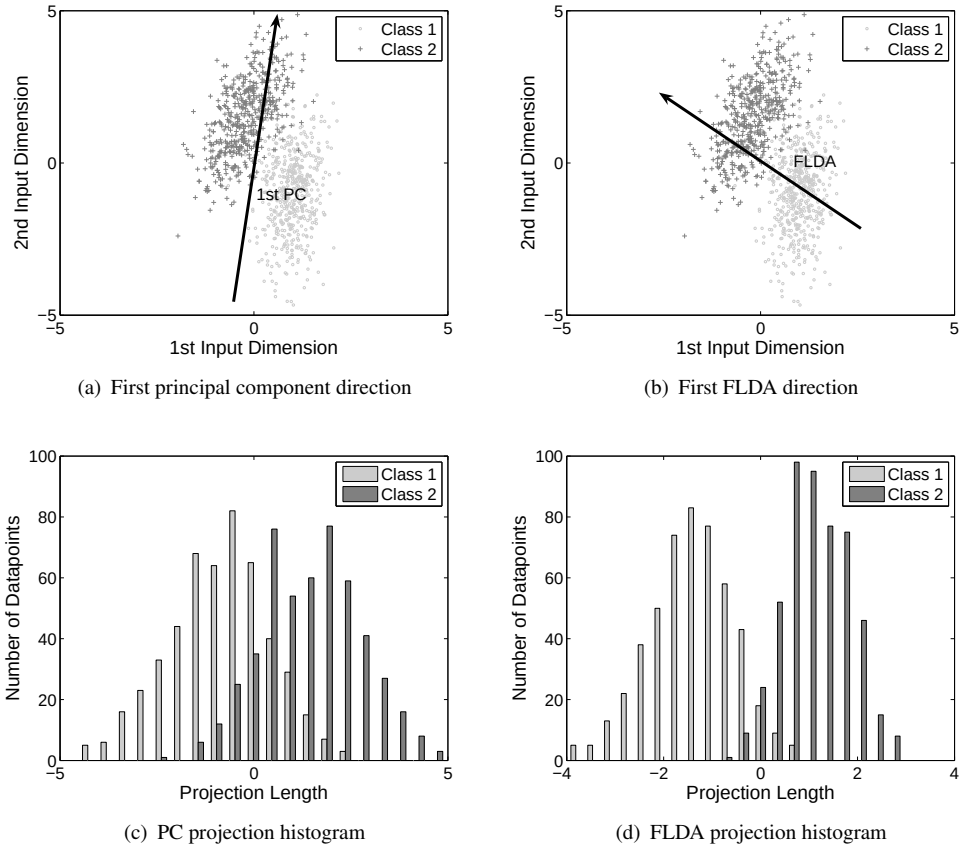
It can be shown that $\boldsymbol{w} = \boldsymbol{C}^{-1} \cdot \left(\boldsymbol{\mu}^{(k)} - \boldsymbol{\mu}^{(l)} \right)$ is a solution to the eigenvector problem in equation (2.9). In the following, the data are represented by their projections on $\boldsymbol{w}$. In this one–dimensional projected space, $\tilde{\boldsymbol{x}} = \frac{1}{2} \cdot \left(\tilde{\boldsymbol{\mu}}^{(k)} + \tilde{\boldsymbol{\mu}}^{(l)} \right)$ represents the point of equal posterior probability. Hence, the hyperplane of FLDA can be found in the original space as:

$$\begin{aligned} 0 &= \tilde{\boldsymbol{x}} - \frac{1}{2} \cdot \left(\tilde{\boldsymbol{\mu}}^{(k)} + \tilde{\boldsymbol{\mu}}^{(l)} \right) \\ &= \boldsymbol{w}^T \cdot \boldsymbol{x} - \frac{1}{2} \cdot \left(\boldsymbol{w}^T \cdot \boldsymbol{\mu}^{(k)} + \boldsymbol{w}^T \cdot \boldsymbol{\mu}^{(l)} \right) \\ &= \left(\boldsymbol{\mu}^{(k)} - \boldsymbol{\mu}^{(l)} \right)^T \cdot \boldsymbol{C}^{-1} \cdot \boldsymbol{x} - \frac{1}{2} \cdot \left(\left(\boldsymbol{\mu}^{(k)} + \boldsymbol{\mu}^{(l)} \right)^T \cdot \boldsymbol{C}^{-1} \cdot \left(\boldsymbol{\mu}^{(k)} - \boldsymbol{\mu}^{(l)} \right) \right) \end{aligned} \tag{2.10}$$

It can be easily seen that equation (2.6) is equivalent to equation (2.10). This proves that FLDA is equivalent to LDA for two–class problems of Gaussian variables with equal prior probabilities and equal, invertible covariance matrices.

2.6 Canonical Correlation Analysis

If a common source $\boldsymbol{s} \in \mathbb{R}^N$ influences two datasets $\boldsymbol{X} \in \mathbb{R}^{D \times N}$ and $\boldsymbol{Z} \in \mathbb{R}^{K \times N}$, of possibly different dimensionality, canonical correlation analysis (CCA) (Hotelling, 1936) can be used to extract this inherent similarity. The goal of CCA is to find two vectors to project the



(a) First principal component direction

(b) First FLDA direction

(c) PC projection histogram

(d) FLDA projection histogram

FIGURE 2.5: Comparison of FLDA and PCA for classification. (a) The first principal component represents the direction of maximum variance in the data. (b) FLDA finds a direction that maximises the overall variance of data projections while minimizing the variance of projections from each class. (c) The histograms of the class projections on the first principal component strongly overlap. Note that this projected space is not optimal for classification. (d) Using the FLDA classifier, the histograms of the class projections separate better than in (c).

datasets onto such that their projection lengths are maximally correlated. Let $\boldsymbol{C_{XZ}}$ denote the cross–covariance matrix between the datasets $\boldsymbol{X}$ and $\boldsymbol{Z}$. Then the CCA problem is given by maximisation of the objective function:

$$J(\boldsymbol{a}, \boldsymbol{b}) = \frac{\boldsymbol{a}^T \cdot \boldsymbol{C_{XZ}} \cdot \boldsymbol{b}}{\sqrt{\boldsymbol{a}^T \cdot \boldsymbol{C_{XX}} \cdot \boldsymbol{a}} \cdot \sqrt{\boldsymbol{b}^T \cdot \boldsymbol{C_{ZZ}} \cdot \boldsymbol{b}}} \tag{2.11}$$

over the vectors $\boldsymbol{a}$ and $\boldsymbol{b}$. The CCA problem can be solved by a singular value decomposition (SVD) of $\boldsymbol{C_{XX}^{-\frac{1}{2}}} \cdot \boldsymbol{C_{XZ}} \cdot \boldsymbol{C_{ZZ}^{-\frac{1}{2}}}$ (Mardia et al., 1979). The solution can be obtained by solving the two eigenvector problems:

$$\left(\boldsymbol{C_{XX}^{-\frac{1}{2}}} \cdot \boldsymbol{C_{XZ}} \cdot \boldsymbol{C_{ZZ}^{-1}} \cdot \boldsymbol{C_{ZX}} \cdot \boldsymbol{C_{XX}^{-\frac{1}{2}}}\right) \cdot \boldsymbol{a} = \lambda \cdot \boldsymbol{a} \tag{2.12}$$

and

$$\left(\boldsymbol{C_{ZZ}^{-\frac{1}{2}}} \cdot \boldsymbol{C_{ZX}} \cdot \boldsymbol{C_{XX}^{-1}} \cdot \boldsymbol{C_{XZ}} \cdot \boldsymbol{C_{ZZ}^{-\frac{1}{2}}}\right) \cdot \boldsymbol{b} = \lambda \cdot \boldsymbol{b} \ . \tag{2.13}$$

One can hypothesise that the maximally correlated projections $\boldsymbol{X}^T \cdot \boldsymbol{a}$ and $\boldsymbol{Z}^T \cdot \boldsymbol{b}$ represent an estimate of the common source.

Canonical correlation analysis can be used to extract classification relevant information from a set of inputs. Indeed, let $\boldsymbol{X}$ be the union of all data points and $\boldsymbol{Z}$ the table of corresponding class memberships, $k = 1, \ldots, K$ and $i = 1, \ldots, N$:

$$\boldsymbol{Z}_{ki} = \begin{cases} 1, & \text{if } \boldsymbol{x}_i \in \boldsymbol{X}^{(k)} \\ 0, & \text{otherwise.} \end{cases} \tag{2.14}$$

All classification relevant information is represented by this classification table. Therefore, this information is retained in those input components of $\boldsymbol{X}$ that originate from a common virtual source with the classification table.

In the following, we align the domain specific notation of CCA and FLDA to simplify their comparison. While CCA is defined using covariance matrices, FLDA uses scatter matrices $\boldsymbol{S}$. The sample covariance matrix $\boldsymbol{C_{XX}} = \frac{1}{N}\boldsymbol{S_{XX}}$, the scatter of the points in $\boldsymbol{X}$. Analogous to the correlation matrices, we denote $\boldsymbol{S_{XZ}} = \boldsymbol{S_{ZX}^T}$ to be the cross scatter matrix. The sample covariance matrices in equation (2.11) can be replaced by their corresponding scatter matrices because of the scaling invariance of the CCA criterion. In the following, $\underline{\underline{\mathbf{1}}}$ is a matrix of ones and $\boldsymbol{P} = \boldsymbol{I} - \frac{1}{N} \cdot \underline{\underline{\mathbf{1}}}$ represents a mean subtracting projection matrix. If $\boldsymbol{Z} \cdot \boldsymbol{Z}^T$ is of full rank, the scatter matrices can be computed as follows:

$$\begin{aligned}
\boldsymbol{S_{XX}} &= (\boldsymbol{X} \cdot \boldsymbol{P}) \cdot (\boldsymbol{X} \cdot \boldsymbol{P})^T = \boldsymbol{X} \cdot \boldsymbol{X}^T - \frac{1}{N} \cdot \boldsymbol{X} \cdot \underline{\underline{\mathbf{1}}} \cdot \boldsymbol{X}^T \\
\boldsymbol{S_{XZ}} &= (\boldsymbol{X} \cdot \boldsymbol{P}) \cdot (\boldsymbol{Z} \cdot \boldsymbol{P})^T = \boldsymbol{X} \cdot \boldsymbol{Z}^T \cdot \left(\boldsymbol{Z} \cdot \boldsymbol{Z}^T\right)^{-1} \cdot \left(\boldsymbol{Z} \cdot \boldsymbol{Z}^T - \frac{1}{N} \cdot \boldsymbol{Z} \cdot \underline{\underline{\mathbf{1}}} \cdot \boldsymbol{Z}^T\right) \\
\boldsymbol{S_{ZZ}} &= (\boldsymbol{Z} \cdot \boldsymbol{P}) \cdot (\boldsymbol{Z} \cdot \boldsymbol{P})^T = \boldsymbol{Z} \cdot \boldsymbol{Z}^T - \frac{1}{N} \cdot \boldsymbol{Z} \cdot \underline{\underline{\mathbf{1}}} \cdot \boldsymbol{Z}^T
\end{aligned}$$

Note also that the scatter matrix $\boldsymbol{S_{XX}}$ can be decomposed to a within- and between–class scatter matrix (Duda and Hart, 1973, p. 119) $\boldsymbol{S_{XX}} = \boldsymbol{S}_W + \boldsymbol{S}_B$ with

$$\boldsymbol{S}_W = \boldsymbol{X} \cdot \left(\boldsymbol{I} - \boldsymbol{Z}^T \cdot \left(\boldsymbol{Z} \cdot \boldsymbol{Z}^T\right)^{-1} \cdot \boldsymbol{Z}\right) \cdot \boldsymbol{X}^T$$

and

$$\boldsymbol{S}_B = \boldsymbol{X} \cdot \boldsymbol{Z}^T \cdot \left(\boldsymbol{Z} \cdot \boldsymbol{Z}^T\right)^{-1} \cdot \boldsymbol{Z} \cdot \boldsymbol{X}^T - \frac{1}{N} \cdot \boldsymbol{X} \cdot \underline{\underline{\mathbf{1}}} \cdot \boldsymbol{X}^T \ .$$

The CCA problem can be solved by the two eigenvector problems (2.12) and (2.13). By replacing the sample covariance matrices with their corresponding scatter matrices, the eigenvector problems can be rewritten as equations (2.15) and (2.16) below:

$$\begin{aligned}
\left(\boldsymbol{S_{XX}^{-\frac{1}{2}}} \cdot \boldsymbol{S_{XZ}} \cdot \boldsymbol{S_{ZZ}^{-1}} \cdot \boldsymbol{S_{ZX}} \cdot \boldsymbol{S_{XX}^{-\frac{1}{2}}}\right) \cdot \boldsymbol{a} &= \lambda \cdot \boldsymbol{a} \quad \text{, or} \\
\left((\boldsymbol{S}_W + \boldsymbol{S}_B)^{-\frac{1}{2}} \cdot \boldsymbol{S}_B \cdot (\boldsymbol{S}_W + \boldsymbol{S}_B)^{-\frac{1}{2}}\right) \cdot \boldsymbol{a} &= \lambda \cdot \boldsymbol{a} \ .
\end{aligned}$$

This further implies:

$$\begin{aligned} \boldsymbol{S}_B \cdot \boldsymbol{a} &= \lambda \cdot (\boldsymbol{S}_W + \boldsymbol{S}_B) \cdot \boldsymbol{a} \quad \text{, or} \\ \boldsymbol{S}_B \cdot \boldsymbol{a} &= \frac{\lambda}{1-\lambda} \cdot \boldsymbol{S}_W \cdot \boldsymbol{a} \ . \end{aligned} \tag{2.15}$$

Similarly:

$$\left(\boldsymbol{S}_{\boldsymbol{ZZ}}^{-\frac{1}{2}} \cdot \boldsymbol{S}_{\boldsymbol{ZX}} \cdot \boldsymbol{S}_{\boldsymbol{XX}}^{-1} \cdot \boldsymbol{S}_{\boldsymbol{XZ}} \cdot \boldsymbol{S}_{\boldsymbol{ZZ}}^{-\frac{1}{2}} \right) \cdot \boldsymbol{b} = \lambda \cdot \boldsymbol{b} \tag{2.16}$$

Note that the eigenvector equation (2.15) is equivalent to the FLDA formulation in equation (2.8). The two equations have the same eigenvector as solutions. In other words, the FLDA discriminant features ($\boldsymbol{w}$'s) are the same as the directions $\boldsymbol{a}$, which are solutions of the modified CCA criterion. The equivalence between this special CCA approach and FLDA has been also shown in slightly different ways (Bartlett, 1938; Mardia et al., 1979; Hastie et al., 1994; Bach and Jordan, 2005). This validates the use of a classification table as second CCA input set. Furthermore, it shows that CCA is a generalization of FLDA contributing a statistical foundation to this empirical approach.

2.7 Least Squares Estimation

The goal of least square estimation (LSE) (Kay, 1993) is to find a linear model that optimally describes the relation between the inputs $\boldsymbol{X} \in \mathbb{R}^{N \times D}$ and the observed outputs $\boldsymbol{y} \in \mathbb{R}^N$. A common application of LSE is the prediction of future outputs given new inputs. In the following, we discuss LSE and its properties due to similarities in its mathematical formulation with work in Chapter 4. The ordinary LSE signal model is given by:

$$\boldsymbol{y} = \boldsymbol{X} \cdot \boldsymbol{\beta} + \boldsymbol{n} \tag{2.17}$$

with $\hat{E}\{\boldsymbol{n}\} = \boldsymbol{0}$ and the covariance matrix of $\boldsymbol{n}$ being $\boldsymbol{C}_{\boldsymbol{n}} = \sigma^2 \cdot \boldsymbol{I}$. As illustrated in Figure 2.6(a), the optimality criterion of LSE is given by the minimum sum of squared residuals. Thus, its cost function can be found to: $J(\boldsymbol{\beta}) = \|\boldsymbol{y} - \boldsymbol{X} \cdot \boldsymbol{\beta}\|^2$. By setting the gradient of the cost function to zero, the estimator results in:

$$\hat{\boldsymbol{\beta}}_{\mathrm{LS}} = \left(\boldsymbol{X}^T \cdot \boldsymbol{X}\right)^{-1} \cdot \boldsymbol{X}^T \cdot \boldsymbol{y} \tag{2.18}$$

By inserting the signal model into the LSE function, it can easily be seen that LSE is unbiased: $\hat{\boldsymbol{\beta}}_{\mathrm{LS}} = \hat{E}\left\{\left(\boldsymbol{X}^T \cdot \boldsymbol{X}\right)^{-1} \cdot \boldsymbol{X}^T \cdot (\boldsymbol{X} \cdot \boldsymbol{\beta} + \boldsymbol{n})\right\} = \boldsymbol{\beta}$. Kay (1993, Theorem 3.2) shows that if a pdf $p(\boldsymbol{x}, \boldsymbol{\beta})$ satisfies the regularity condition: $\hat{E}\left\{\frac{\delta \log p(\boldsymbol{x},\boldsymbol{\beta})}{\delta \boldsymbol{\beta}}\right\} = \boldsymbol{0} \quad \forall \boldsymbol{\beta}$, an unbiased estimator $\boldsymbol{\beta}$ may attain the Cramer-Rao lower bound (CRLB) if and only if $\frac{\delta \log p(\boldsymbol{x},\boldsymbol{\beta})}{\delta \boldsymbol{\beta}} = \boldsymbol{I}(\boldsymbol{\beta}) \cdot (g(\boldsymbol{x}) - \boldsymbol{\beta})$. Here, $\boldsymbol{I}(\boldsymbol{\beta})$ and g($\boldsymbol{x}$) represent the Fisher information matrix and the minimum variance unbiased estimator (MVU) $\hat{\boldsymbol{\beta}}_{\mathrm{MVU}} = g(\boldsymbol{x})$ respectively. Note that $\boldsymbol{I}^{-1}(\boldsymbol{\beta})$ equals the covariance matrix of $\hat{\boldsymbol{\beta}}_{\mathrm{MVU}}$. In the following it is shown that for the special

case $\boldsymbol{n} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \cdot \boldsymbol{I})$, $\hat{\boldsymbol{\beta}}_{\mathrm{LS}}$ is the best linear unbiased estimator (BLUE):

$$\begin{aligned}\frac{\delta \log p(\boldsymbol{x}, \boldsymbol{\beta})}{\delta \boldsymbol{\beta}} &= \frac{\delta}{\delta \boldsymbol{\beta}} \left(-\log\left(2\pi\sigma^2\right)^{\frac{D}{2}} - \frac{1}{2\sigma^2} \left(\boldsymbol{y} - \boldsymbol{X} \cdot \boldsymbol{\beta}\right)^T \cdot \left(\boldsymbol{y} - \boldsymbol{X} \cdot \boldsymbol{\beta}\right) \right) \\ &= \frac{1}{\sigma^2} \left(\boldsymbol{X}^T \cdot \boldsymbol{y} - \boldsymbol{X}^T \cdot \boldsymbol{X} \cdot \boldsymbol{\beta}\right) \\ &= \frac{\boldsymbol{X}^T \cdot \boldsymbol{X}}{\sigma^2} \cdot \left(\left(\boldsymbol{X}^T \cdot \boldsymbol{X}\right)^{-1} \cdot \boldsymbol{X}^T \cdot \boldsymbol{y} - \boldsymbol{\beta} \right)\end{aligned}$$

Note that in this case, $\boldsymbol{I}(\boldsymbol{\beta}) = \frac{\boldsymbol{X}^T \cdot \boldsymbol{X}}{\sigma^2}$ and $\hat{\boldsymbol{\beta}}_{\mathrm{MVU}} = \left(\boldsymbol{X}^T \cdot \boldsymbol{X}\right)^{-1} \cdot \boldsymbol{X}^T \cdot \boldsymbol{y} = \hat{\boldsymbol{\beta}}_{\mathrm{LS}}$. This result can be generalised to arbitrary pdf's of $\boldsymbol{n}$ with $\hat{E}\{\boldsymbol{n}\} = \boldsymbol{0}$ and covariance matrix $\boldsymbol{C_n}$. In this case the Gauss-Markov theorem states that the BLUE is found to be $\hat{\boldsymbol{\beta}}_{\mathrm{BLUE}} = \left(\boldsymbol{X}^T \cdot \boldsymbol{C_n}^{-1} \cdot \boldsymbol{X}\right)^{-1} \cdot \boldsymbol{X}^T \cdot \boldsymbol{C_n}^{-1} \cdot \boldsymbol{y}$ (Kay, 1993).

2.7.1 Total Least Squares

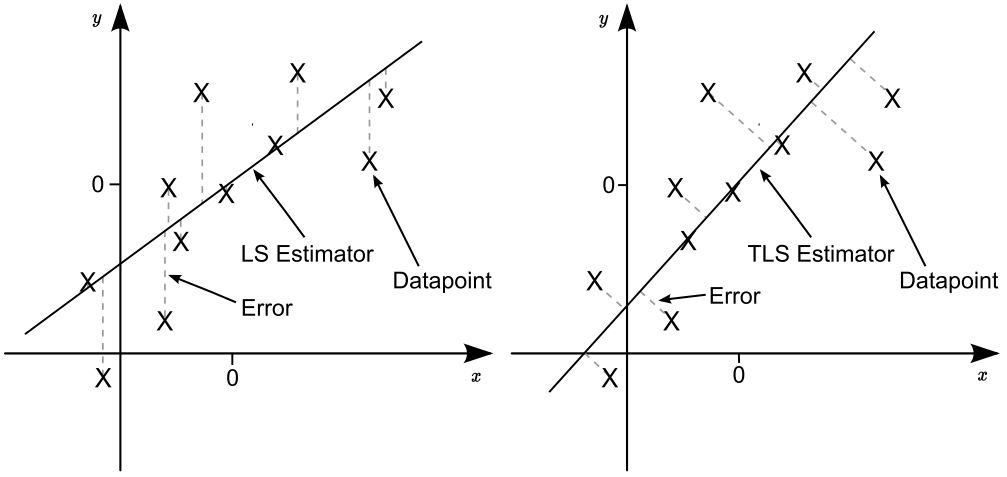
Total least squares estimation (TLS) is a generalised version of the ordinary LSE. It is also known as "orthogonal regression", "errors in variables" and "rigorous least squares". While the term TLS is relatively new, its general idea goes back to Adcock (1877). An overview of TLS, including recent advances and more detailed historical information, can be found in Markovsky and Huffel (2007). The goal of TLS is to model not only noise in the input data $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]^T \in \mathbb{R}^{N \times D}$ but also measurement uncertainties in $\boldsymbol{y} \in \mathbb{R}^N$. This can be realised by modification of the ordinary LSE model in equation (2.17) to:

$$\boldsymbol{y} + \Delta\boldsymbol{y} = (\boldsymbol{X} + \Delta\boldsymbol{X}) \cdot \boldsymbol{\beta}$$

TLS finds the linear transformation $\boldsymbol{\beta}$ that minimises the residuals in both inputs and outputs. Similarly to ordinary LSE, $\boldsymbol{\Delta} = [\Delta\boldsymbol{X}, \Delta\boldsymbol{y}]$ is assumed to be zero mean and Gaussian distributed with covariance matrix $\boldsymbol{C_\Delta} = \sigma^2 \cdot \boldsymbol{I}$. Geometrically, the TLS approach can be viewed as the search for a linear function that minimises the sum of squared orthogonal distances to the input points. A visualisation of the approach as well as a comparison to ordinary LSE is shown in Figure 2.6. The cost function $J(\boldsymbol{\beta})$ of the TLS method can be found using the normal equation of the linear model:

$$\begin{bmatrix} \boldsymbol{X} & \boldsymbol{y} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{\beta} \\ -1 \end{bmatrix} = \underline{\boldsymbol{0}}$$

Projection of the data points onto the normalised normal vector of the linear model results in:



(a) Ordinary least squares estimation (b) Total least squares estimation

FIGURE 2.6: Comparison of ordinary and total least squares estimation. (a) Ordinary least squares estimation finds a linear approximation of the data that minimises the squared distances to the outputs y. (b) Total least squares estimation minimises the orthogonal distances to the desired linear data representation.

$$J(\boldsymbol{\beta}) = \left\| \frac{\begin{bmatrix} \boldsymbol{X} & \boldsymbol{y} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{\beta} \\ -1 \end{bmatrix}}{\sqrt{\begin{bmatrix} \boldsymbol{\beta} & -1 \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{\beta} \\ -1 \end{bmatrix}}} \right\|^2 = \frac{\|\boldsymbol{y} - \boldsymbol{X} \cdot \boldsymbol{\beta}\|^2}{\|\boldsymbol{\beta}\| + 1} \tag{2.19}$$

As discussed in Golub and Loan (1980) and Markovsky and Huffel (2007), there exists a closed form solution to

$$\hat{\boldsymbol{\beta}}_{\mathrm{TLS}} = \arg\min_{\boldsymbol{\beta}} J(\boldsymbol{\beta})$$

if $\boldsymbol{X}$ is of full rank and has unique eigenvalues $\sigma_1^2 \leq \ldots \leq \sigma_i^2$. In this case, the value of the cost function equals the smallest eigenvalue of the inputs $J\left(\hat{\boldsymbol{\beta}}\right) = \sigma_i^2$. The closed form solution to this special TLS case is given by:

$$\hat{\boldsymbol{\beta}}_{\mathrm{TLS}} = \left(\boldsymbol{X}^T \cdot \boldsymbol{X} - \sigma_i^2 \boldsymbol{I}\right)^{-1} \cdot \boldsymbol{X}^T \cdot \boldsymbol{y} \tag{2.20}$$

Note that equation (2.20) represents a negatively regularised version of the ordinary LSE in equation (2.18). This results in a condition number increase of the inverse which leads to a reduced robustness of TLS to errors.

2.7.2 Ridge Regression

In many real world applications, inputs are highly correlated. The resulting near collinearity leads to ill–conditioning of the inverse $(\boldsymbol{X}^T \cdot \boldsymbol{X})^{-1}$. The goal of ridge regression (RR) (Hoerl and Kennard, 1970) is to overcome this problem by penalising large norms of the estimated parameters $\boldsymbol{\beta}$. A generalisation of RR is known as Tikhonov regularization (TR). In the following, let $\boldsymbol{\Gamma}$ represent an arbitrary matrix called Tikhonov matrix. The cost function of TR is given by:

$$J(\boldsymbol{\beta}) = \|\boldsymbol{y} - \boldsymbol{X} \cdot \boldsymbol{\beta}\|^2 + \|\boldsymbol{\Gamma} \cdot \boldsymbol{\beta}\|^2 \tag{2.21}$$

Setting the derivative of equation (2.21) to zero, the closed form solution of TR can be found as:

$$\hat{\boldsymbol{\beta}}_{\mathrm{TR}} = \left(\boldsymbol{X}^T \cdot \boldsymbol{X} - \boldsymbol{\Gamma}^T \cdot \boldsymbol{\Gamma}\right)^{-1} \cdot \boldsymbol{X}^T \cdot \boldsymbol{y}$$

In RR the Tikhonov matrix is assumed to be a diagonal matrix of non–negative values (Hoerl and Kennard, 1970). Usually, a scaled version of the identity matrix $\boldsymbol{\Gamma} = \sqrt{\alpha}\,\boldsymbol{I}$ is used. Thus, in the literature the RR solution is commonly given by:

$$\hat{\boldsymbol{\beta}}_{\mathrm{RR}} = \left(\boldsymbol{X}^T \cdot \boldsymbol{X} - \alpha\,\boldsymbol{I}\right)^{-1} \cdot \boldsymbol{X}^T \cdot \boldsymbol{y} \tag{2.22}$$

In the following, the often empirically–found α is referred to as the Tikhonov factor. Using this assumption, the regularisation can be interpreted as follows. On the one hand, it can be shown that α introduces a bias that can lead to an estimate $\hat{\boldsymbol{\beta}}_{\mathrm{RR}}$ with smaller mean square error than the corresponding least square estimate $\hat{\boldsymbol{\beta}}_{\mathrm{LSE}}$. On the other hand, the regularisation can be simply viewed as a penalty on the sparseness of $\hat{\boldsymbol{\beta}}_{\mathrm{RR}}$. It is the intuition that in the case of strongly collinear inputs a minimum mean square estimator is close to their mean. If the Tikhonov matrix is not simplified to a scaled identity matrix, it can be used to include prior knowledge, i.e., smoothness of consecutive variables in $\hat{\boldsymbol{\beta}}_{\mathrm{TR}}$.

2.8 The Bayesian General Linear Model

If there is prior knowledge of the model parameters available, it can be advantageous to use a Bayesian approach. The expected outcome can be predicted based on the posterior probability using the additionally available information of the input distributions. In this way, it is possible to find biased estimators that can achieve a lower variance than the corresponding BLUE. In the following, we introduce the Bayesian general linear model (BGLM) to show a formal derivation of ridge regression. Furthermore, the BGLM builds on the foundation in Section 4.1

to motivate the integration of prior knowledge. Similar to regression, the BGLM is used to model statistically a system and predict future outputs. The general linear model is defined as:

$$\boldsymbol{y} = \boldsymbol{X} \cdot \boldsymbol{\beta} + \boldsymbol{n} \tag{2.23}$$

As discussed in Kay (1993), the expected value $\hat{E}\{\boldsymbol{\beta}|\boldsymbol{y}\}$ from the conditional probability $p(\boldsymbol{\beta}|\boldsymbol{y})$ can be introduced as a biased estimator of $\boldsymbol{\beta}$. If $\boldsymbol{n} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{C_n})$ and $\boldsymbol{\beta} \sim \mathcal{N}(\boldsymbol{\mu_\beta}, \boldsymbol{C_\beta})$ are independent Gaussian variables, the joint probability density function (pdf) $p(\boldsymbol{y}, \boldsymbol{\beta})$ as well as the conditional pdf $p(\boldsymbol{\beta}|\boldsymbol{y})$ are Gaussian. Considering our prior assumptions, $p(\boldsymbol{y}) = \mathcal{N}(\boldsymbol{\mu_y}, \boldsymbol{C_y})$ and $p(\boldsymbol{y}, \boldsymbol{\beta}) = N\left(\begin{bmatrix}\boldsymbol{\mu_y}\\ \boldsymbol{\mu_\beta}\end{bmatrix}, \begin{bmatrix}\boldsymbol{C_y} & \boldsymbol{C_{y\beta}}\\ \boldsymbol{C_{\beta y}} & \boldsymbol{C_\beta}\end{bmatrix}\right)$. Using these assumptions, the conditional probability can be computed as follows:

$$\begin{aligned} p(\boldsymbol{\beta}|\boldsymbol{y}) &= \frac{p(\boldsymbol{y}, \boldsymbol{\beta})}{p(\boldsymbol{y})} \\ &= \frac{\frac{1}{\sqrt{(2\pi)^{D+N}\left|\begin{bmatrix}\boldsymbol{C_y} & \boldsymbol{C_{y\beta}}\\ \boldsymbol{C_{\beta y}} & \boldsymbol{C_\beta}\end{bmatrix}\right|}} \exp\left[-\frac{1}{2}\begin{bmatrix}\boldsymbol{y}-\boldsymbol{\mu_y}\\ \boldsymbol{\beta}-\boldsymbol{\mu_\beta}\end{bmatrix}^T \cdot \begin{bmatrix}\boldsymbol{C_y} & \boldsymbol{C_{y\beta}}\\ \boldsymbol{C_{\beta y}} & \boldsymbol{C_\beta}\end{bmatrix}^{-1} \cdot \begin{bmatrix}\boldsymbol{y}-\boldsymbol{\mu_y}\\ \boldsymbol{\beta}-\boldsymbol{\mu_\beta}\end{bmatrix}\right]}{\frac{1}{\sqrt{(2\pi)^D|\boldsymbol{C_y}|}} \exp\left[-\frac{1}{2}\left(\boldsymbol{y}-\boldsymbol{\mu_y}\right)^T \cdot \boldsymbol{C_y}^{-1} \cdot \left(\boldsymbol{y}-\boldsymbol{\mu_y}\right)\right]} \end{aligned}$$

As discussed in Horn and Johnson (1999), the partitioned determinant can be transformed to $\left|\begin{bmatrix}\boldsymbol{C_y} & \boldsymbol{C_{y\beta}}\\ \boldsymbol{C_{\beta y}} & \boldsymbol{C_\beta}\end{bmatrix}\right| = |\boldsymbol{C_y}|\left|\boldsymbol{C_\beta} - \boldsymbol{C_{\beta y}} \cdot \boldsymbol{C_y}^{-1} \cdot \boldsymbol{C_{y\beta}}\right|$. Therefore, the conditional probability can be simplified to:

$$p(\boldsymbol{\beta}|\boldsymbol{y}) = \frac{\exp\left[-\frac{1}{2}\left[\begin{bmatrix}\boldsymbol{y}-\boldsymbol{\mu_y}\\ \boldsymbol{\beta}-\boldsymbol{\mu_\beta}\end{bmatrix}^T \cdot \begin{bmatrix}\boldsymbol{C_y} & \boldsymbol{C_{y\beta}}\\ \boldsymbol{C_{\beta y}} & \boldsymbol{C_\beta}\end{bmatrix}^{-1} \cdot \begin{bmatrix}\boldsymbol{y}-\boldsymbol{\mu_y}\\ \boldsymbol{\beta}-\boldsymbol{\mu_\beta}\end{bmatrix} - \left(\boldsymbol{y}-\boldsymbol{\mu_y}\right)^T \cdot \boldsymbol{C_y}^{-1} \cdot \left(\boldsymbol{y}-\boldsymbol{\mu_y}\right)\right]\right]}{\sqrt{(2\pi)^N\left|\boldsymbol{C_\beta} - \boldsymbol{C_{\beta y}} \cdot \boldsymbol{C_y}^{-1} \cdot \boldsymbol{C_{y\beta}}\right|}}$$

This representation of the posterior probability $p(\boldsymbol{\beta}|\boldsymbol{y})$ unveils the covariance matrix $\boldsymbol{C_{\beta|y}} = \boldsymbol{C_\beta} - \boldsymbol{C_{\beta y}} \cdot \boldsymbol{C_y}^{-1} \cdot \boldsymbol{C_{y\beta}}$. Using the matrix inversion lemma for partitioned matrices, the joint covariance matrix of $p(\boldsymbol{y}, \boldsymbol{\beta})$ can be found as:

$$\begin{bmatrix}\boldsymbol{C_y} & \boldsymbol{C_{y\beta}}\\ \boldsymbol{C_{\beta y}} & \boldsymbol{C_\beta}\end{bmatrix}^{-1} = \begin{bmatrix}\boldsymbol{C_y}^{-1}+\boldsymbol{C_y}^{-1}\cdot\boldsymbol{C_{y\beta}}\cdot\boldsymbol{C_{\beta|y}}^{-1}\cdot\boldsymbol{C_{\beta y}}\cdot\boldsymbol{C_y}^{-1} & -\boldsymbol{C_y}^{-1}\cdot\boldsymbol{C_{y\beta}}\cdot\boldsymbol{C_{\beta|y}}^{-1}\\ -\boldsymbol{C_{\beta|y}}^{-1}\cdot\boldsymbol{C_{\beta y}}\cdot\boldsymbol{C_y}^{-1} & \boldsymbol{C_{\beta|y}}^{-1}\end{bmatrix}$$

After some computation, the posterior probability $p(\boldsymbol{\beta}|\boldsymbol{y})$ can be simplified to:

$$p(\boldsymbol{\beta}|\boldsymbol{y}) = \frac{1}{\sqrt{(2\pi)^N\left|\boldsymbol{C_{\beta|y}}\right|}} \exp\left[-\frac{1}{2}\left[\boldsymbol{\beta} - \hat{E}\{\boldsymbol{\beta}|\boldsymbol{y}\}\right]^T \cdot \boldsymbol{C_{\beta|y}}^{-1} \cdot \left[\boldsymbol{\beta} - \hat{E}\{\boldsymbol{\beta}|\boldsymbol{y}\}\right]\right]$$

where

$$\hat{E}\{\boldsymbol{\beta}|\boldsymbol{y}\} = \boldsymbol{\mu_\beta} + \boldsymbol{C_{\beta y}} \cdot \boldsymbol{C_y}^{-1} \cdot \left(\boldsymbol{y} - \boldsymbol{\mu_y}\right)$$

Using the linear signal model $\boldsymbol{y} = \boldsymbol{X} \cdot \boldsymbol{\beta} + \boldsymbol{n}$ as well as the independence assumption of $\boldsymbol{\beta}$

and $\boldsymbol{n}$, the $\boldsymbol{y}$ dependent components can be expanded to:

$$\begin{aligned}
\boldsymbol{\mu_y} &= \hat{E}\left\{\boldsymbol{X}\cdot\boldsymbol{\beta}+\boldsymbol{n}\right\} = \boldsymbol{X}\cdot\boldsymbol{\mu_\beta} \\
\boldsymbol{C_y} &= \hat{E}\left\{\left(\boldsymbol{X}\cdot\boldsymbol{\beta}+\boldsymbol{n}-\boldsymbol{X}\cdot\boldsymbol{\mu_\beta}\right)\cdot\left(\boldsymbol{X}\cdot\boldsymbol{\beta}+\boldsymbol{n}-\boldsymbol{X}\cdot\boldsymbol{\mu_\beta}\right)^T\right\} = \boldsymbol{X}\cdot\boldsymbol{C_\beta}\cdot\boldsymbol{X}^T+\boldsymbol{C_n} \\
\boldsymbol{C_{\beta y}} &= \hat{E}\left\{\left(\boldsymbol{\beta}-\boldsymbol{\mu_\beta}\right)\cdot\left(\boldsymbol{X}\cdot\boldsymbol{\beta}+\boldsymbol{n}-\boldsymbol{X}\cdot\boldsymbol{\mu_\beta}\right)^T\right\} = \boldsymbol{C_\beta}\cdot\boldsymbol{X}^T
\end{aligned}$$

Thus, the posterior expectation of $\boldsymbol{\beta}$ given $\boldsymbol{y}$ is found as:

$$\begin{aligned}
\hat{E}\left\{\boldsymbol{\beta}|\boldsymbol{y}\right\} &= \boldsymbol{\mu_\beta}+\boldsymbol{C_\beta}\cdot\boldsymbol{X}^T\cdot\left(\boldsymbol{X}\cdot\boldsymbol{C_\beta}\cdot\boldsymbol{X}^T+\boldsymbol{C_n}\right)^{-1}\cdot\left(\boldsymbol{y}-\boldsymbol{X}\cdot\boldsymbol{\mu_\beta}\right) \\
&= \boldsymbol{\mu_\beta}+\left(\boldsymbol{X}^T\cdot\boldsymbol{C_n^{-1}}\cdot\boldsymbol{X}+\boldsymbol{C_\beta^{-1}}\right)^{-1}\cdot\boldsymbol{X}^T\cdot\boldsymbol{C_n^{-1}}\cdot\left(\boldsymbol{y}-\boldsymbol{X}\cdot\boldsymbol{\mu_\beta}\right) \quad (2.24)
\end{aligned}$$

Ridge regression (Section 2.7.2) follows from the result in equation (2.24) by further assuming $\boldsymbol{\mu_\beta}=\underline{\boldsymbol{0}}$, $\boldsymbol{C_\beta}=\sigma_{\boldsymbol{\beta}}^2\,\boldsymbol{I}$ and $\boldsymbol{C_n}=\sigma_{\boldsymbol{n}}^2\,\boldsymbol{I}$:

$$\boldsymbol{\beta}_{\mathrm{RIDGE}}=\left(\boldsymbol{X}^T\cdot\boldsymbol{X}+\frac{\sigma_{\boldsymbol{n}}^2}{\sigma_{\boldsymbol{\beta}}^2}\boldsymbol{I}\right)^{-1}\cdot\boldsymbol{X}^T\cdot\boldsymbol{y}$$

For this special case, when the covariance matrices are diagonal and isotropic, the 'optimal' Tikhonov regularization factor $\alpha=\frac{\sigma_{\boldsymbol{n}}^2}{\sigma_{\boldsymbol{\beta}}^2}$ is unveiled.

2.9 Gaussian Mixture Models

Gaussian mixture models (GMM) (Duda and Hart, 1973, p. 170) are a standard and popular approach for classification and probability density estimation. This section discusses the design, training and use of GMM's providing a background for further chapters. GMM's have been successfully used in a wide range of statistical signal processing applications including speaker identification and verification (Reynolds, 1995). The goal of GMM's is to model datasets that originate from a mixture of independent Gaussian variables. The general method assumes the number of mixtures/classes to be known and their parametric distribution families to be Gaussian. Figure 2.7 shows an example of a GMM. In the following, let $\boldsymbol{x}_i \in \mathbb{R}^D$ with $i=1\ldots N$, $\boldsymbol{\mu}^{(k)} \in \mathbb{R}^D$ and $\boldsymbol{C}^{(k)} \in \mathbb{R}^{D\times D}$ with $k=1\ldots K$ representing the N unlabelled input data points, the mean vector and the covariance matrix of the K classes respectively. Furthermore, let $P(k)$ be the probability of class occurrence and $p(\boldsymbol{x}|k)\sim\mathcal{N}(\boldsymbol{\mu}^{(k)},\boldsymbol{C}^{(k)})$. The GMM probability density function $p(\boldsymbol{x}_i|\boldsymbol{\theta})$ with $\boldsymbol{\theta}=\{P(k),\boldsymbol{\mu}^{(k)},\boldsymbol{C}^{(k)}\}_{k=1}^K$ is given by:

$$p(\boldsymbol{x}_i|\boldsymbol{\theta})=\sum_{k=1}^{K}\frac{P(k)}{\sqrt{(2\cdot\pi)^D\cdot\left|\boldsymbol{C}^{(k)}\right|}}\cdot\exp\left\{-\frac{1}{2}\cdot\left(\boldsymbol{x}_i-\boldsymbol{\mu}^{(k)}\right)^T\cdot\boldsymbol{C}^{(k)^{-1}}\cdot\left(\boldsymbol{x}_i-\boldsymbol{\mu}^{(k)}\right)\right\} \quad (2.25)$$

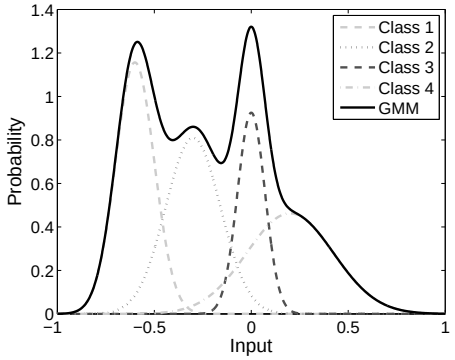


FIGURE 2.7: The Gaussian mixture model represents the data by an additive combination of multiple, differently parametrised and weighted Gaussian probability density functions.

Moreover, the likelihood of $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]$ being distributed by the probability density function in equation (2.25) is given by:

$$L(\boldsymbol{X}|\boldsymbol{\theta}) = \prod_{i=1}^{N} p(\boldsymbol{x}_i|\boldsymbol{\theta}) \tag{2.26}$$

The problem of finding the missing parameters $\boldsymbol{\mu}^{(k)}$, $\boldsymbol{C}^{(k)}$ and $\hat{P}(k)$ is commonly solved with an expectation–maximization (EM) approach (Dempster et al., 1977). The original idea of the EM algorithm is based on previous work of, e.g., Hartley (1958) and Baum (1972). To provide the EM algorithm with an initial starting point, the unknown parameters are guessed. Here it has to be noted that the initial starting conditions influence the final convergence of the EM algorithm. Only convergence to a local optimum is achieved. The EM algorithm is a procedure of two subsequent steps that are iterated until convergence. In the first step, the expectation of the desired probability density function is computed given a set of inputs and parameters. In the second step, the maximum likelihood solution to a refined set of parameters is found utilising the result of the first step. The expectation step for the GMM is given by:

$$m_{ik} = \frac{\hat{P}(k)\, p(\boldsymbol{x}_i|\boldsymbol{\mu}^{(k)}, \boldsymbol{C}^{(k)})}{\sum_{l=1}^{K} \hat{P}(l)\, p(\boldsymbol{x}_i|\boldsymbol{\mu}^{(l)}, \boldsymbol{C}^{(l)})} \tag{2.27}$$

In equation (2.27), the partial membership of each data point $\boldsymbol{x}_i$ in each class k is estimated. The maximisation step of the GMM approach finds the maximally–likely parameters $\boldsymbol{\mu}^{(k)}$, $\boldsymbol{C}^{(k)}$ and $\hat{P}(k)$ given m_{ik}:

$$
\begin{aligned}
\hat{P}(k) &= \frac{1}{N}\sum_{i=1}^{N} m_{ik} \\
\boldsymbol{\mu}^{(k)} &= \frac{\sum_{i=1}^{N} m_{ik}\,\boldsymbol{x}_i}{\sum_{i=1}^{N} m_{ik}} \\
\boldsymbol{C}^{(k)} &= \frac{\sum_{i=1}^{N} m_{ik}\,\left(\boldsymbol{x}_i - \boldsymbol{\mu}^{(k)}\right)\cdot\left(\boldsymbol{x}_i - \boldsymbol{\mu}^{(k)}\right)^T}{\sum_{i=1}^{N} m_{ik}}
\end{aligned}
$$

The probability density function of the GMM that models $\boldsymbol{X}$ in a maximum likelihood sense is found by insertion of the converged EM solution into equation (2.26). This equation is also utilised to evaluate the fit of new datasets to this GMM representation. Further information on the EM algorithm including an introduction and extensions can be found in Bishop (2006, p. 435) and McLachlan and Krishnan (1997) respectively.

2.10 Summary

This chapter described ubiquitous signal processing methods providing a background to later chapters of this thesis. Different philosophies for the extraction of characteristic information from an input were discussed. For example, PCA considers the information content of input projections that represent the highest amount of variance in the data to be maximally important. In contrast, MCA assumes invariant data projections to characterize the inputs. Again XCA uses a mixture of both minor and principal components to represent a dataset best. Other methods, e.g., LDA and FLDA, use class labels to extract features that capture classification relevant structure. Moreover, it was shown that CCA is a generalization of the empirical FLDA criterion. CCA aims to extract a common signal from two datasets finding pairs of directions on which the data projections are maximally correlated. Thereafter, least squares estimation was recapitulated providing another view on input model generation. Finally, the Bayesian general linear and Gaussian mixture models were discussed. Both approaches are used in later chapters of this thesis.

Chapter 3

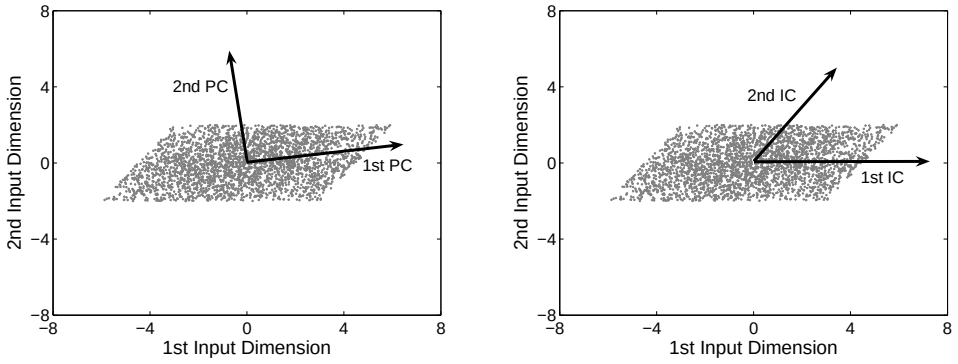
Higher–Order and Nonlinear Methods

Second–order statistics can be used to find simple and robust solutions to real world problems. However, in some applications, the inferred Gaussian distribution is insufficient to model the real distribution of the data. A possible answer to this problem is to use Gaussian mixture models to approximate these non–Gaussian distributions (Todros and Tabrikian, 2004). However, multiple Gaussian distributions have to be used to represent a single non–Gaussian distribution. Therefore, such an approach artificially increases the complexity of the solution. This chapter gives an introduction to higher–order and nonlinear methods that are designed to tackle the problem of non-Gaussian distributed data. Although the discussed methods are only partially linked to later chapters, they provide an important extension to the signal processing concepts discussed in Chapter 2.

Section 3.1 discusses different independent component analysis (ICA) based algorithms which use higher order statistics to model the data. Thereafter, Section 3.2 gives an introduction to a selection of kernel methods that enable simplified nonlinear signal processing. Finally, Section 3.3 summarizes this chapter.

3.1 Independent Component Analysis

The task of independent component analysis (ICA) (Jutten and Herault, 1991) is to find and separate a number of linearly mixed statistically independent source signals. Statistical independence of two random variables s_1 and s_2 is given if their joint pdf factorises to the product of their marginal pdf's: $p(s_1, s_2) = p(s_1)\, p(s_2)$. It follows from this property that independent random variables are uncorrelated: $E\{s_1\, s_2\} = E\{s_1\}\, E\{s_2\}$. This attribute is commonly used to simplify the procedure of finding independent components. The most prominent ICA application is the "cocktail–party problem" (Cherry, 1953). Here, a microphone array is used to record a number of people speaking simultaneously in the same room. Thereafter, ICA is utilised to separate the different speech signals by means of the recorded data and the assumption of statistically independent sources. Figure 3.1 shows the difference



(a) PCA of subgaussian data

(b) ICA of subgaussian data

FIGURE 3.1: Comparison of PCA and ICA using linear transformed subgaussian data. (a) The principal component directions do not represent the non–Gaussian data structure. (b) ICA finds non–orthogonal directions that are aligned with the data.

of PCA and ICA on a non–Gaussian dataset. Note that ICA finds a closer representation of the inherent data structure. In the following, let $\boldsymbol{x}$, $\boldsymbol{s}$ and $\boldsymbol{A}$ represent the measured data, the unknown source signals and the mixing matrix respectively. The basic linear and noiseless ICA model is given by:

$$\boldsymbol{x} = \boldsymbol{A} \cdot \boldsymbol{s} \tag{3.1}$$

The mixing matrix and source signal are both estimated purely based on the measurements $\boldsymbol{x}$ and the assumption of statistical independence of $\boldsymbol{s}$. In order to reduce the influence of noise it is common to compress the input using the Karhunen–Loève transformation (Section 2.1.1). Moreover, because of the uncorrelation of the independent components $\boldsymbol{s}$ it is possible to simplify the search by whitening (Section 2.1.2) of $\boldsymbol{x}$. In the following, let $\boldsymbol{z}$ denote the whitened and preprocessed measurements $\boldsymbol{x}$. The unmixing model of ICA is given by:

$$\boldsymbol{s} = \boldsymbol{B} \cdot \boldsymbol{z}$$

Whitening and preprocessing of the inputs leads to a transfer of $\boldsymbol{A}$ to a new mixing matrix $\boldsymbol{B}^{-1} = \boldsymbol{V} \cdot \boldsymbol{A}$. The new mixing matrix is orthogonal constraining the search for an unmixing matrix $\boldsymbol{B}$ to the space of orthogonal matrices. By assuming $\hat{E}\left\{\boldsymbol{z} \cdot \boldsymbol{z}^T\right\} = \hat{E}\left\{\boldsymbol{s} \cdot \boldsymbol{s}^T\right\} = \boldsymbol{I}$, the orthogonality of $\boldsymbol{B}^{-1}$ can be seen from:

$$\hat{E}\left\{\boldsymbol{z} \cdot \boldsymbol{z}^T\right\} = \boldsymbol{B}^{-1} \cdot \hat{E}\left\{\boldsymbol{s} \cdot \boldsymbol{s}^T\right\} \cdot (\boldsymbol{B}^{-1})^T = \boldsymbol{B}^{-1} \cdot (\boldsymbol{B}^{-1})^T = \boldsymbol{I}$$

A way to visualise ICA is to regard the recovered data $\boldsymbol{s}$ as an orthogonal projection of the whitened data $\boldsymbol{z}$. Hence, the task of ICA can be pictured as a rotation of the data in the whitened

space such that the independent components point along the axes. In the following, let $\boldsymbol{w}_i$ denote the orthogonal directions on which the whitened data is projected. Note that because of ambiguities of ICA (Section 3.1.4), $\boldsymbol{w}_i$ can only represent the independent component directions up to their orientation and sequence ($\boldsymbol{w}_i = \pm\boldsymbol{b}_j$). Here, $\boldsymbol{b}_j$ represents column j of the unmixing matrix $\boldsymbol{B}$. The solution of ICA is commonly derived using a maximum likelihood approach, by utilising higher order statistics (i.e., kurtosis) or minimisation of mutual information in the data representations. These principles are described in the following.

3.1.1 Kurtosis

The fourth–order statistic of the data, called kurtosis, is a simple discriminant for the search of independent components. Kurtosis is positive for super–Gaussian, negative for sub–Gaussian and zero for Gaussian distributed random variables. Therefore it is a measure of nongaussianity. In the case of zero mean variables, the kurtosis value of the vector $\boldsymbol{y}$ is calculated as:

$$\text{kurt}(\boldsymbol{y}) = \hat{E}\left\{\boldsymbol{y}^4\right\} - 3(\hat{E}\left\{\boldsymbol{y}^2\right\})^2 \tag{3.2}$$

The link between nongaussianity and the independent component directions can be visualised using the central limit theorem (Laplace, 1810). According to this theorem, the pdf of a mixture of non–Gaussian, iid variables is closer distributed to a Gaussian than the pdf's of the original variables. Therefore, the variables maximise statistical independence if the data representation minimises the similarity of its variables to Gaussian distributions. Thus, kurtosis–based ICA algorithms aim to find directions that maximize the absolute kurtosis value. In the following, let $\boldsymbol{z}$ represents the whitened version of $\boldsymbol{x}$ and $y = \boldsymbol{w}^T \cdot \boldsymbol{z}$. The derivative of the absolute kurtosis value from equation (3.2) with respect to the projecting direction $\boldsymbol{w}$ is given by:

$$\frac{\partial \text{abs}\left(\text{kurt}(\boldsymbol{w}^T \cdot \boldsymbol{z})\right)}{\partial \boldsymbol{w}} = 4\,\text{sign}\big(\text{kurt}(\boldsymbol{w}^T \cdot \boldsymbol{z})\big)\left[\hat{E}\left\{\boldsymbol{z}\,(\boldsymbol{w}^T \cdot \boldsymbol{z})^3\right\} - 3\,\boldsymbol{w}\,\|\boldsymbol{w}\|^2\right]$$

This result can be used to derive a simple version of the FastICA algorithm (Hyvärinen and Oja, 1997; Hyvärinen, 1999). The iterative kurtosis–based FastICA learning rule is given by:

$$\boldsymbol{w} \leftarrow \hat{E}\left\{\boldsymbol{z}\,(\boldsymbol{w}^T \cdot \boldsymbol{z})^3\right\} - 3\,\boldsymbol{w} \tag{3.3}$$

The advantages of this algorithm are its cubic convergence and the missing learning rate parameter. This makes it a simple but powerful tool for the search for independent components. The procedure to find independent components using equation (3.3) can be described as follows. First, a random start vector $\boldsymbol{w_1}$ is chosen for $\boldsymbol{w}$. Second, using equation (3.3), the direction $\boldsymbol{w_1}$ is iteratively found that maximises the value of kurtosis. After each iteration, the norm of $\boldsymbol{w_1}$ is reset to unity to retain the unit variance of $\boldsymbol{z}$. As discussed in Section 3.1, the task of ICA

is to rotate the whitened independent components in the directions of the axes. Thus, after convergence, the independent component is aligned with the coordinate axis by multiplication of the data $\boldsymbol{z}$ with the 'optimized' vector $\boldsymbol{w_1}$. To align other components, the process is repeated with new vectors $\boldsymbol{w_i}$ that are constrained to the orthogonal space of previously found directions. Therefore, the alignment from previous transformations is maintained.

This method can be interpreted visually by rotating the whitened input data cloud to receive maximal non–Gaussian projections on the axes. If this point of maximisation is reached, the axes point in the directions of maximally independent components. Although it is possible to create simple and computational effective algorithms based on kurtosis, this measure is rarely used in processing of real–life data. The reason for this is its strong dependence on outliers. Hence, the kurtosis measure is usually replaced by a function that reduces this high emphasis on distanced values. One example for such a weighting function is $\frac{1}{a}\log\left(\cosh\left(a\,\boldsymbol{y}\right)\right)$ with an arbitrary factor a.

3.1.2 Maximum Likelihood

A common method for finding independent components is maximum likelihood (ML) estimation. This approach is very important since many other ICA methods can be described as a special case or derived from it. The aim of this estimation is to find directions in a multidimensional data set that have the highest probability to represent the original data. Hence, the probability density functions p_k of all k independent components have to be known or need to be estimated. Thereafter, an unmixing matrix $\boldsymbol{B}$ is found that transfers the pdf's of the mixed data $\boldsymbol{x}$ to match with the estimated pdf's maximally. In case the input data is whitened, the ML estimation finds the independent components unbiased by the uncertainty of the initial pdf estimation. It was shown in Hyvärinen et al. (2001) that the initial pdf only has to be chosen from the correct family, i.e., sub- or super–Gaussian. Here it should be noted that the ML estimation does not need whitened input data to find the independent components. However, pre–whitening of the input is advantageous. This preprocessing step reduces the space of the sought–after unmixing matrix $\boldsymbol{B}$ to the space of orthogonal matrices. Consequently, the convergence of the ML based gradient algorithms is significantly improved (Hyvärinen et al., 2001). This optimisation is also used to derive another version of the FastICA algorithm. In the following, let $\boldsymbol{s}$ represent the unmixed independent components, $\boldsymbol{A}$ the unknown mixing matrix, $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N] \in \mathbb{R}^{D\times N}$ the input data and $\boldsymbol{b}_k$ the column k of the unmixing matrix $\boldsymbol{B}$. The log likelihood algorithm, including a short derivation, is given by:

$$
\begin{aligned}
\boldsymbol{x} &= \boldsymbol{A} \cdot \boldsymbol{s} \\
\Rightarrow \boldsymbol{s} &= \boldsymbol{A}^{-1} \cdot \boldsymbol{x} = \boldsymbol{B} \cdot \boldsymbol{x} \\
\Rightarrow \boldsymbol{s}_k &= \boldsymbol{b}_k^T \cdot \boldsymbol{x} \\
p_x(\boldsymbol{x}) &= \text{abs}(|\boldsymbol{B}|)\, p_s(\boldsymbol{s}) = \text{abs}(|\boldsymbol{B}|) \prod_k p_k(\boldsymbol{s}_k) \\
L(\boldsymbol{B}) &= \prod_{i=1}^{N} \prod_{k=1}^{K} p_k\left(\boldsymbol{b}_k^T \cdot \boldsymbol{x}_i\right) \text{abs}(|\boldsymbol{B}|) \\
\arg\max_{\boldsymbol{B}}\Big(\mathcal{L}(\boldsymbol{B})\Big) &= \arg\max_{\boldsymbol{B}} \left(\sum_{i=1}^{N} \sum_{k=1}^{K} \log\Big(p_k\big(\boldsymbol{b}_k^T \cdot \boldsymbol{x}_i\big)\Big) + N \log\Big(\text{abs}(|\boldsymbol{B}|)\Big) \right)
\end{aligned}
$$

The log likelihood measure has the same extreme value positions as the likelihood $L(\boldsymbol{B})$. However, the logarithm simplifies the algorithm. Therefore, the logarithmic likelihood formulation is usually preferred. With slight changes to the algorithm, the sum over N can be replaced by the empirical expectation operator. Hence, ML estimation maximises the expectation that resulting variables are generated by processes with pdf's that are similar to the estimates.

3.1.3 Mutual Information

Minimisation of mutual information aims to find a data representation that minimises the dependencies between the input variables. This alternative concept for ICA is included because of its importance in the literature and for the completeness of this introduction. Additionally to a discussion of the method, the similarities to different, parallel ICA approaches are analysed. These similarities suggest the meaningfulness of the methods. The mutual information $I(\cdot)$ between variables $\{y_i\}_{i=1\ldots N}$ is given by:

$$
I(y_1, \ldots, y_N) = \sum_{i=1}^{N} H(y_i) - H(\boldsymbol{y})
$$

Note that $\sum_{i=1}^{N} H(y_i) \geq H(\boldsymbol{y})$ and $I(\boldsymbol{y}) \geq 0$. Suppose that the ICA model of equation (3.1) holds and $\boldsymbol{y} = \boldsymbol{B} \cdot \boldsymbol{z}$ with $\boldsymbol{B}$ being an orthogonal rotation matrix. In this case it can be shown that the entropy $H(\boldsymbol{y})$ is independent of $\boldsymbol{B}$. Thus, minimisation of mutual information equals the minimisation of $\sum_{i=1}^{N} H(y_i)$. The alternative definition of mutual information $I(\boldsymbol{y}) = K\big(p(\boldsymbol{y}) || \prod_{i=1}^{N} p(y_i)\big)$ illustrates its minimum for independent variables. Here,

$$
K(P \| Q) = \sum_i P(y_i) \cdot \log\left(\frac{P(y_i)}{Q(y_i)}\right)
$$

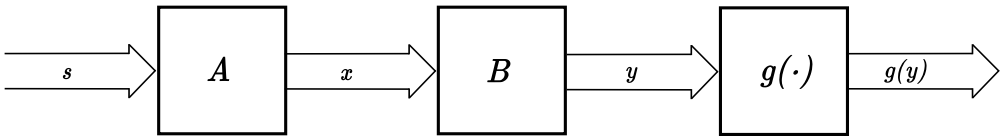


FIGURE 3.2: Schematic structure of the Infomax algorithm. The nonlinearity $g(\cdot)$ represents the node of a neural network. Mixing matrix $\boldsymbol{B}$ is found to transfer input $\boldsymbol{x}$ to a uniform distributed variable $g(\boldsymbol{y})$.

represents the Kullback–Leibler divergence (KLD) of the pdf's P and Q. $K\big(p(\boldsymbol{y})||\prod_{i=1}^{N} p(y_i)\big) = 0$ if the joint pdf $p(\boldsymbol{y})$ equals the product of the marginal pdf's $p(y_i)$, which is the definition of independent variables y_i. A similar approach is taken in neural network solutions, the Infomax principle of Bell and Sejnowski (1995) and the ML estimation. Here, the independent components are found by maximisation of the output entropy $H\left(g(\boldsymbol{y})\right)$.

Figure 3.2 illustrates the structure of the Infomax based ICA algorithm. In the following, $\boldsymbol{s}$ and $\boldsymbol{A}$ represent the vector of independent components and the mixing matrix respectively. Both are unknown and have to be estimated based on the output vector $\boldsymbol{x}$. The output $\boldsymbol{y}$ of the unmixing matrix $\boldsymbol{B}$ is transferred by a nonlinear function $g(\cdot)$. The optimum of the mutual information minimising algorithm is found if the transformed output $g(\boldsymbol{y})$ is uniformly distributed. This ICA approach is equivalent to a neural network solution where the node utilises the nonlinear function $g(\cdot)$ and the training adapts the unmixing matrix $\boldsymbol{B}$ to maximise the similarity of the output with a uniform distribution. Cardoso (1997) and Hyvärinen et al. (2001) show the equality between output entropy maximization and ML estimation for the case where the nonlinearity $g(\cdot)$ is chosen to be the cumulative density function (cdf) of the original data $\boldsymbol{x}$. This results from the uniform distribution of variables that are transformed by their own cdf. Theorem 2.1 by Devroye (1986) illustrates this transformation to a uniform distributed variable.

Alternatively, the equivalence of output entropy maximisation and the previously–discussed KLD–based method can be explained as follows. On the one hand, the KLD–based method aims to find a $\boldsymbol{B}$ that minimises the difference of $p(\boldsymbol{b}_i^T \cdot \boldsymbol{x})$ to prior known marginal pdf's $p(y_i)$. On the other hand, the Infomax method transfers the marginal pdf's to uniform distributions using the nonlinear function $g(\cdot)$. Thereafter, Infomax aims to find a $\boldsymbol{B}$ that minimises the difference to the now–uniform distributed marginal pdf's. This is equivalent to maximisation of the output entropy. If $P = p\big(g(\boldsymbol{B} \cdot \boldsymbol{x})\big)$, the output entropy is given by:

$$H(P) \stackrel{\text{def}}{=} \log(N) - K(P||\mathcal{U})$$

Therefore, the unmixing matrix $\boldsymbol{B}$ is found by minimisation of the Kullback–Leibler divergence $K(P||\mathcal{U})$. A more detailed discussion of this method can be found in Cardoso (1997).

3.1.4 Limits of ICA

Using only higher–order statistics leads to one of the major issues of ICA. Because the fourth–order statistics of Gaussian variables are zero, Gaussian signals cannot be separated by simple ICA. The border case, which can still be tackled, is a mixture with a single Gaussian signal. This can be achieved by exclusion of the detectable sub- and super–Gaussian variables. Another way of looking at the separation problem of Gaussian distributed variables is based on the central limit theorem. If the pdf of a variable is strongly dissimilar to a Gaussian distribution, the variable is less likely to be a combination of multiple iid variables. Hence, as discussed earlier, the difference to a Gaussian distribution can be used as measure of independence (Hyvärinen et al., 2001). On the other hand, a Gaussian distributed variable could be interpreted as a mixture of an infinite number of independent components. Therefore, normal distributed variables cannot be separated using simple ICA. Another feature that cannot be extracted by ICA is the orientation of the original components. Moreover, the independence constraint does not give information about the sequence of the found components. Hence, all permutations and orientations of the independent components are equivalent ICA results.

3.2 Kernel Methods

The kernel methods (KMs) (Shawe-Taylor and Cristianini, 2004) were introduced to tackle signal processing problems that are poorly modelled by linear systems. These problems include e.g., analysis of nonlinear relationships between variables, text, images and sequenced data. Solving such nonlinear problems directly is often infeasible because of their high complexity and the problematic interpretation of stability and statistical significance of the result. Hence, KMs use the so called "kernel trick", i.e., they analyse the inputs in a linearised high–dimensional space. Therefore, well–researched linear methods can be used in this kernel domain enabling statistical interpretability of the result. Let $\boldsymbol{x}_i \in \mathbb{R}^D$, $\boldsymbol{y}_j \in \mathbb{R}^D$ with $i, j = 1, \ldots, N$ and $\Phi(\boldsymbol{x}) = [\Phi_1(\boldsymbol{x}), \ldots, \Phi_M(\boldsymbol{x})]$ with $M > D$ represent two variables from the input space and the nonlinear map of $\boldsymbol{x}$ to a higher dimensional kernel–defined feature space respectively. An example of a kernel matrix is the polynomial kernel of second degree for $D = 2$:

$$\begin{aligned} \boldsymbol{K}(\boldsymbol{x}_i, \boldsymbol{y}_j) &= \Phi(\boldsymbol{x}_i)^T \cdot \Phi(\boldsymbol{y}_j) \\ &= \left[x_{1i}^2, x_{2i}^2, \sqrt{2}\, x_{1i}\, x_{2i}\right] \cdot \left[y_{1j}^2, y_{2j}^2, \sqrt{2}\, y_{1j}\, y_{2j}\right]^T = \left(\boldsymbol{x}_i^T \cdot \boldsymbol{y}_j\right)^2 \end{aligned} \tag{3.4}$$

Second degree polynomial relationships between the variables in the original two–dimensional space are simplified to linear relationships in the mapped three–dimensional space. Moreover, it can be seen in equation (3.4) that the kernel matrix simplifies to the square of a scalar product. Therefore, if a signal processing algorithm utilises only the kernel matrix, there is no need to explicitly compute points in the high–dimensional space. Here it should be noted that the kernel matrix may not contain all information from the original datapoints. One example of

this is the Gram matrix based kernel: $\boldsymbol{K}(\boldsymbol{x},\boldsymbol{x})_{ij} = \Phi(\boldsymbol{x}_i)^T \cdot \Phi(\boldsymbol{x}_j)$. Because all rotation matrices $\boldsymbol{R}$ satisfy the orthogonality condition $\boldsymbol{R} \cdot \boldsymbol{R}^T = \boldsymbol{I}$, the Gram matrix based kernel of inner products is invariant to rotation around the origin. This illustrates the importance of the problem–dependent kernel choice. In the following, a selection of prominent kernel–based statistical signal processing methods are discussed. Further information on kernel methods and their application in support vector machines can be found in Shawe-Taylor and Cristianini (2004).

3.2.1 Kernel PCA

Kernel PCA (KPCA) utilises nonlinear dependencies for feature extraction and data compression. The approach is motivated by the successful implementation of the kernel trick in the field of support vector machines. The experience in this field proved that many natural dependencies are not purely linear. Therefore, the transfer of a signal processing problem to a nonlinear space has the potential of finding better solutions. Figure 3.3 illustrates the basic principle of KPCA. The nonlinearity Φ is used to linearise the original input space. Thereafter, the transfered data are analysed using PCA. Thus, KPCA can be seen as the application of PCA in a kernel–defined input space. In the following, $\boldsymbol{E}^k$ denotes the eigenvector k of the data in this kernel space. The desired projections onto the eigenvectors in the nonlinear space are given by:

$$\boldsymbol{E}^k \cdot \Phi(\boldsymbol{x}) = \sum_{i=1}^{N} a_i^k \, \Phi(\boldsymbol{x}_i)^T \cdot \Phi(\boldsymbol{x})$$

Here, $\boldsymbol{a}^k$ represents the map from the span $\Phi(\boldsymbol{x})$ to the eigenvector $\boldsymbol{E}^k$. This representation was chosen for variable reduction in the formula. In the following, let $\boldsymbol{D}$ and $\boldsymbol{K}_{ij} = \Phi(\boldsymbol{x}_i)^T \cdot \Phi(\boldsymbol{x}_j)$ represent the diagonal matrix of sorted eigenvalues and the kernel matrix respectively. The map to the KPCA space $\boldsymbol{a}^k = \frac{1}{\sqrt{\boldsymbol{D}_{kk}}} \boldsymbol{E}^k$ can be found by eigenvalue decomposition of the kernel matrix $[\boldsymbol{E}, \boldsymbol{D}] = \text{eig}\{\boldsymbol{K}\}$. The KPCA projection on component k is given by: $\hat{\boldsymbol{x}}_i^k = \sum_{i=1}^{N} a_i^k \boldsymbol{K}(\boldsymbol{x}_i, \boldsymbol{x})$. More details including a stability analysis of KPCA can be found in Schölkopf and Smola (2002) and Shawe-Taylor and Cristianini (2004).

3.2.2 Kernel CCA

As discussed in Section 2.6, CCA is a powerful linear method generalising e.g., LDA, Fisher's LDA and PCA. In parallel to the linear case, the goal of kernel CCA (KCCA) (Bach and Jordan, 2002; Shawe-Taylor and Cristianini, 2004) is to find the common source of a set of variables. However, rather than linear projections, KCCA finds a set of nonlinear functions such that the correlations in the transformed kernel space are maximised. For clarity, we assume a set of two variables $\boldsymbol{x}$ and $\boldsymbol{y} \in \mathbb{R}^D$. In the following, let $f(\cdot)$ and $g(\cdot)$ represent two nonlinear functions from reproducing kernel Hilbert spaces (RKHS) $\mathcal{F}_f$ and $\mathcal{F}_g \subset \mathbb{R}^D$ respectively. Furthermore,

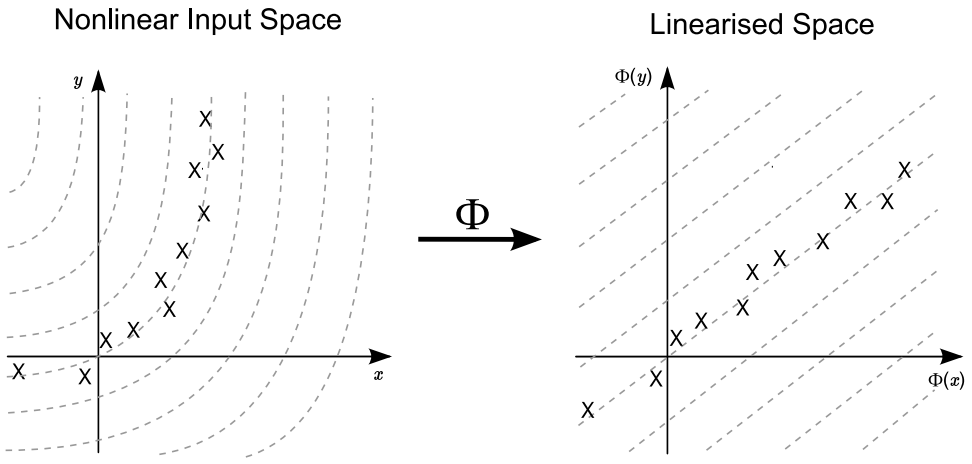


FIGURE 3.3: Linearisation of a feature extraction problem utilizing information of nonlinear dependencies between variables. This enables the use of linear methods for nonlinear feature extraction.

let $\boldsymbol{K}_f(\boldsymbol{x},\boldsymbol{y}) = \Phi_f(\boldsymbol{x})^T \cdot \Phi_f(\boldsymbol{y})$, $\mathcal{F}_f = \sum_{i=1}^N a_i \boldsymbol{K}_f(\boldsymbol{x}_i,\cdot)$ and $\mathcal{F}_g = \sum_{i=1}^N b_i \boldsymbol{K}_g(\boldsymbol{y}_i,\cdot)$ with $a_i, b_i \in \mathbb{R}$. Note that $\boldsymbol{K}_f(\cdot,\cdot)$ and $\boldsymbol{K}_g(\cdot,\cdot)$ represent two kernel matrices and $\mathcal{F}$ a vector space with '$\cdot$' denoting the location of the arguments. Because of the reproducing property of the RKHS:

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} a_i \boldsymbol{K}_f(\boldsymbol{x}_i,\boldsymbol{x}) = \boldsymbol{a}^T \cdot \boldsymbol{K}_f(\boldsymbol{x},\boldsymbol{x}) = \boldsymbol{a}^T \cdot \boldsymbol{K}_f \tag{3.5}$$

$$g(\boldsymbol{y}) = \sum_{i=1}^{N} b_i \boldsymbol{K}_g(\boldsymbol{y}_i,\boldsymbol{y}) = \boldsymbol{b}^T \cdot \boldsymbol{K}_g(\boldsymbol{y},\boldsymbol{y}) = \boldsymbol{b}^T \cdot \boldsymbol{K}_g \tag{3.6}$$

The objective function that is maximised by KCCA is given by:

$$J(f(\boldsymbol{x}), g(\boldsymbol{y})) = \frac{\text{cov}\left\{f(\boldsymbol{x}), g(\boldsymbol{y})\right\}}{\sqrt{\text{var}\left\{f(\boldsymbol{x})\right\}}\sqrt{\text{var}\left\{g(\boldsymbol{y})\right\}}} \tag{3.7}$$

By substituting equations (3.5) and (3.6) in (3.7), the KCCA objective function can be simplified:

$$J(\boldsymbol{a},\boldsymbol{b}) = \frac{\boldsymbol{a}^T \cdot \boldsymbol{K}_f \cdot \boldsymbol{K}_g^T \cdot \boldsymbol{b}}{\sqrt{\boldsymbol{a}^T \cdot \boldsymbol{K}_f \cdot \boldsymbol{K}_f^T \cdot \boldsymbol{a}}\sqrt{\boldsymbol{b}^T \cdot \boldsymbol{K}_g \cdot \boldsymbol{K}_g^T \cdot \boldsymbol{b}}} \tag{3.8}$$

It should be noted that the points are assumed to be centred in the nonlinear kernel space: $\sum_{i=1}^N \Phi_f(\boldsymbol{x_i}) = 0$. If $\boldsymbol{K}$ is a $N \times N$ Gram matrix, this centring can be achieved indirectly by modification of the kernel matrix $\boldsymbol{K} = \boldsymbol{P} \cdot \bar{\boldsymbol{K}} \cdot \boldsymbol{P}$. Here, $\bar{\boldsymbol{K}}$ represents the kernel representation of the not–centred datapoints and $\boldsymbol{P} = \boldsymbol{I} - \frac{1}{N}\underline{\underline{\mathbf{1}}}$. As discussed in Bach and Jordan (2002),

equation (3.8) does not provide a useful estimate of the canonical correlations in the kernel space. The reason for this is the flexibility of the high–dimensional kernel space representation. The highest correlation is achieved if $\boldsymbol{a}$ and $\boldsymbol{b}$ are in the column span of $\boldsymbol{K}_f$ and $\boldsymbol{K}_g$ respectively. In this case equation (3.8) simplifies to a minimisation of the angle between $\boldsymbol{a}$ and $\boldsymbol{b}$. However, if one of the kernel matrices has full rank, its columns represent the full space. For example, if $\boldsymbol{K}_f$ has full rank, the minimum angle between $\boldsymbol{a}$ and $\boldsymbol{b}$ is zero for all possible vectors $\boldsymbol{b}$. This exemplifies the necessity to regularise and constrain the space of solutions. Shawe-Taylor and Cristianini (2004) suggest a regularisation of the norms of $\boldsymbol{a}$ and $\boldsymbol{b}$ leading to the new objective function:

$$J(\boldsymbol{a}, \boldsymbol{b}) = \frac{\boldsymbol{a}^T \cdot \boldsymbol{K}_f \cdot \boldsymbol{K}_g^T \cdot \boldsymbol{b}}{\sqrt{\left((1-\alpha)\ \boldsymbol{a}^T \cdot \boldsymbol{K}_f \cdot \boldsymbol{K}_f^T \cdot \boldsymbol{a} + \alpha \left\|\boldsymbol{a}\right\|^2\right)\ \left((1-\beta)\ \boldsymbol{b}^T \cdot \boldsymbol{K}_g \cdot \boldsymbol{K}_g^T \cdot \boldsymbol{b} + \beta \left\|\boldsymbol{b}\right\|^2\right)}}$$

Here, the parameters $\alpha, \beta \in \mathbb{R}$ are empirically found. As discussed in Bach and Jordan (2002) and Gretton et al. (2005), KCCA can be used as a contrast function to find independent components. If the space of functions $\mathcal{F}$ is large enough, the maximum of equation (3.7) equals zero if and only if the inputs are statistically independent. The intuition is that for uncorrelated not independent variables there exists a nonlinear projection such that their maximum correlation is non–zero.

3.3 Summary

This chapter provided a background on modern approaches addressing non–Gaussian and nonlinear problems. A discussion of ICA and its various cost functions provided a different viewpoint on characteristic information in the inputs. In contrast to previously discussed methods, ICA finds directions whose data projections are statistically independent. That is, the projection length of data onto one independent component provides no statistical information on the projections onto other components. The higher–order statistic kurtosis is used for voice activity detection in Chapter 6. Additionally, kernel methods were described. That is, nonlinear problems are linearized by projection through a kernel function to a specific high dimensional space. Here, the projected problem is addressed linearly. The actual projection to a high dimensional space cancels mathematically and is thus only performed virtually. This illustrated how problems can be simplified by viewing them in a high dimensional space.

Chapter 4

Mutual Interdependence Analysis

In complex applications there is a clear trend toward high–dimensional data acquisition. Examples are image, speech, text processing, bioinformatics and spectroscopy. However, prominent linear signal processing algorithms for feature extraction, regression and classification (see Chapter 2) were developed for problems with a much lower dimensionality D than the number of available datapoints N. One reason for this assumption is that it can prevent issues of high–dimensional data representations demonstrated by the curse of dimensionality (Bellman, 1961). The curse of dimensionality points out that a cartesian grid with 0.1 spacing represents 100 points on a unit square but 10^{10} points on a 10 dimensional unit cube. In the following, a number of examples are given to illustrate the implications of the curse of dimensionality in real problems. A first example is the approximation of a Lipschitz function with D variables. This makes it necessary to perform in the order of $N \propto (\frac{1}{\epsilon})^D$ evaluations on a grid to achieve a uniform approximation error of ϵ. Also for statistical estimation it can be shown that the number of instances that are necessary to model a variable with a given estimation error in a D–dimensional space could grow non-asymptotically with D. Both examples are discussed in more detail in Donoho (2000). Another way to visualize the effect of the curse of dimensionality is the trade off between the number of features e.g., in machine learning. While a large number of features may contain additional information, the estimation of the dependencies between them may require impractically large sets of data. Given these problems, it appears ill–advised to use high–dimensional data representations. One possible solution to the resulting problem of high–dimensional data processing is dimensionality reduction by feature selection. It is necessary to automate such preprocessing because of the amount and complexity of data from modern applications. However, it is unclear if the automatically found compressed space is robust to unknown errors.

As discussed in Section 3.2, processing in high–dimensional spaces also has multiple advantages. One example is that complex nonlinear relationships can be simplified to linear problems in a higher dimensional space. In this way, ICA can be solved in the kernel space using the second order CCA cost function (Section 3.2.2). This suggests that, under some conditions, the correlations between high–dimensional vectors can be used as measure of their dependence.

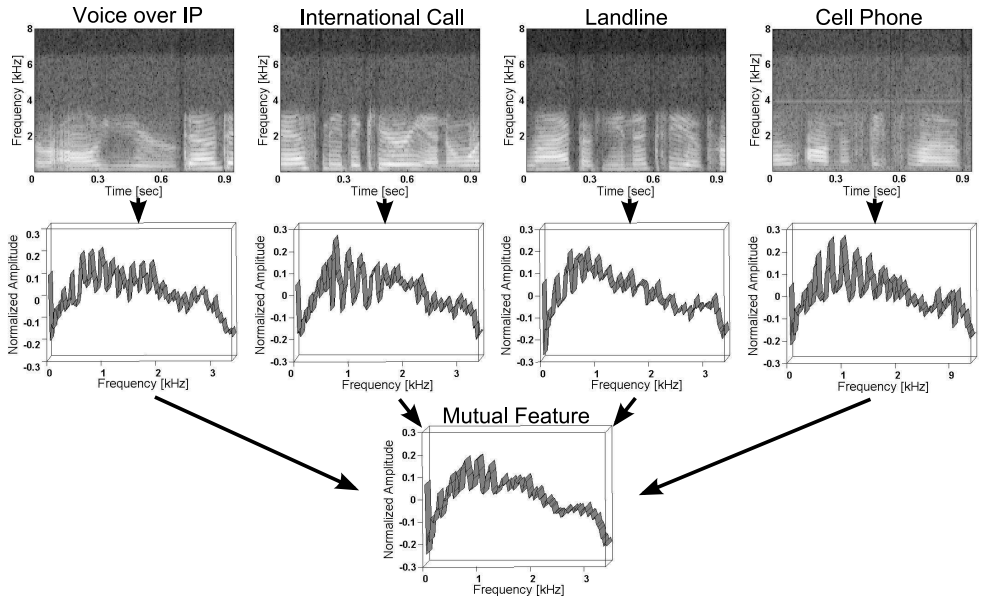


FIGURE 4.1: Device and text independent speaker verification using training data from different, unknown, noisy and nonlinear distorted channels. Does there exists an invariant representation of the data?

Furthermore, high–dimensional data representations are commonly smooth enabling noise cancellation. Additional advantages/properties of high–dimensional data representations as for instance their "concentration of measure" are discussed in Donoho (2000). In summary, taking both advantages and disadvantages into account, it is possible to design stable algorithms that work in a high–dimensional space solving the challenges of modern/complex applications.

This chapter focuses on feature extraction in such a high–dimensional input space. The goal is to extract a single vector that represents the commonalities of a set of inputs. For example, assume the problem of text independent speaker verification. Figure 4.1 illustrates an application where the training data are recordings from different, unknown, noisy and nonlinear distorted channels. Does there exist a single representation of the common invariance in the data? The intuition is that such a representation could be used to verify the origin of unseen recordings. Chapter 6 discusses this application in detail.

A further application where the extraction of a common representation is of interest is illumination invariant face recognition. Figure 4.2 illustrates this image recognition problem. The challenge is that most of the signal energy represents illumination differences rather than structural characteristics of the face. However, is there an inherent structure common to all images of the same face? If this is the case and it can be extracted, this representation can be used as a robust feature for face recognition. More information about this problem and its solution can be found in Chapter 7.

Note that in this chapter the inputs are assumed to be from a single class. Therefore, in contrast to FLDA (Section 2.5), no between–class information can be used to find maximally discriminative features. A reasonable representation $\boldsymbol{w}$ of the high–dimensional inputs $\boldsymbol{x}_i$, which represents

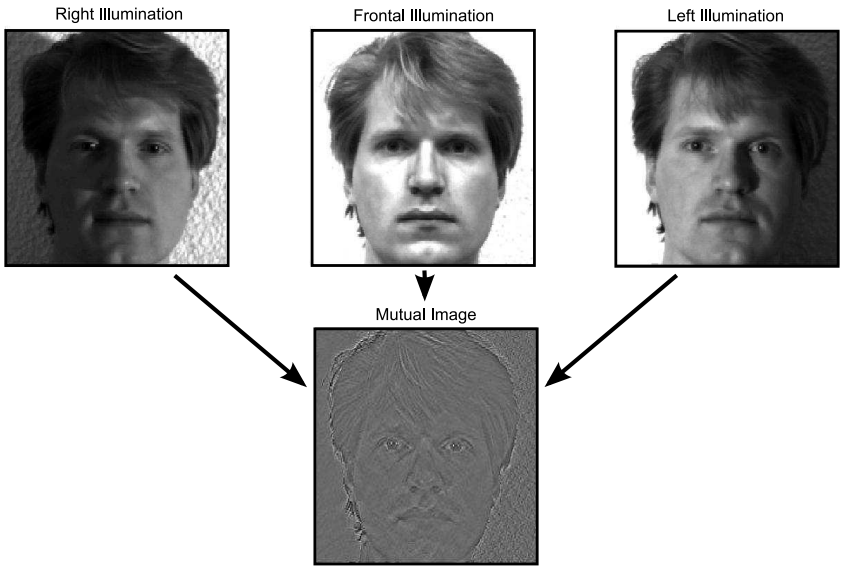


FIGURE 4.2: Illumination invariant face recognition. The main challenge is the high contribution of the illumination differences to the variance between images. Does there exist a mutual, illumination invariant image representation?

them equally, is their mean. However, can a data representation be found that includes additional prior knowledge, e.g., equal dependence of $\boldsymbol{w}$ on the inputs or smoothness of the result? The goal of mutual interdependence analysis (MIA) (Claussen et al., 2007) is to find such a signature. As discussed earlier, the correlation between high–dimensional data vectors can indicate their dependence. Thus MIA finds a data representation that is maximally but equally correlated with all inputs. As discussed in Corollary 4.4 and Section 4.3, such representation is in the span of the inputs. Therefore, MIA finds a vector $\boldsymbol{w}$ that is maximal uniformly correlated with the inputs $\boldsymbol{X}$ while being in their span: $\boldsymbol{w} = \boldsymbol{X} \cdot \boldsymbol{c}$, where $\boldsymbol{c} \in \mathbb{R}^N$.

In the following, assume that $\boldsymbol{w} \in \mathbb{R}^D$ is an equally present, invariant component in a set of N linear independent data vectors $\boldsymbol{x}_i \in \mathbb{R}^D$ with $i = 1, \ldots, N$. Furthermore, let $\boldsymbol{f}_i$ represent the 'variability' of each input vector and assume $\boldsymbol{w}^T \cdot \boldsymbol{f}_i = 0 \quad \forall i$. The mixing model of mutual interdependence analysis is given by:

$$\boldsymbol{x}_i = \boldsymbol{w} + \boldsymbol{f}_i \tag{4.1}$$

MIA aims to extract the invariant component $\boldsymbol{w}$. The previously discussed correlation based criterion of MIA can be interpreted geometrically as the minimisation of the scatter of projection lengths from the inputs onto a desired direction or result $\boldsymbol{w}$. The scatter, a scaled version of the empirical covariance, is given by:

$$\tilde{\boldsymbol{S}}(\boldsymbol{X}|\boldsymbol{w}) = \sum_{i=1}^{N} \left(\boldsymbol{w}^T \cdot \boldsymbol{x}_i - \boldsymbol{w}^T \cdot \boldsymbol{\mu}\right)^2 = \boldsymbol{w}^T \cdot \boldsymbol{S} \cdot \boldsymbol{w} \tag{4.2}$$

where $\boldsymbol{\mu} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i$ is the D–dimensional sample mean and $\boldsymbol{S} = \sum_{i=1}^{N} (\boldsymbol{x}_i - \boldsymbol{\mu}) \cdot (\boldsymbol{x}_i - \boldsymbol{\mu})^T$ is the scatter matrix of the data. Therefore, the MIA problem is to determine $\hat{\boldsymbol{w}}$ as:

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}, \|\boldsymbol{w}\|=1, \boldsymbol{w}=\boldsymbol{X}\cdot\boldsymbol{c}}{\arg\min} \tilde{\boldsymbol{S}}(\boldsymbol{X}|\boldsymbol{w}) \tag{4.3}$$

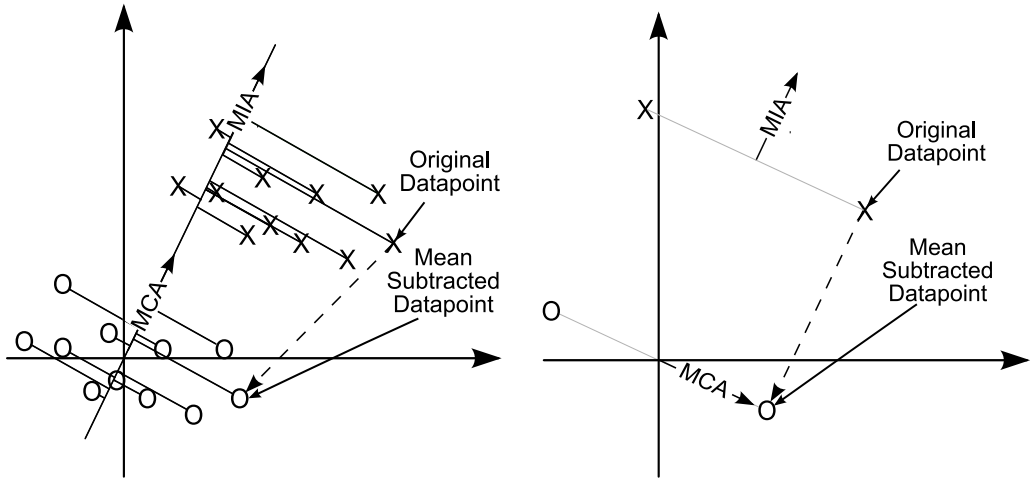
Note that the definitions of FLDA from Section 2.5 and MCA from Section 2.2 also use the within–class scatter matrix $\boldsymbol{S}$. However, in contrast to these, $\boldsymbol{X}$ is not assumed to be preprocessed by mean subtraction. The MIA optimisation problem from equation (4.3) is, up to its sign, unique because of its constrained formulation. Also $\hat{\boldsymbol{w}}$ 'optimally' represents the data samples as one aggregate sample.

As discussed in Section 2.2, MCA has no unique solution if the rank of the scatter matrix is below $D - 1$. Thus, to use MCA in a high–dimensional space with $D > N$ it is reasonable to define the first minor component as the first unique direction of minimum variance. In the following, this constrained MCA formulation is assumed. It can be shown that if rank $(\boldsymbol{S}) < D - 1$, such a direction represents a non–zero variance and is constrained to the space of the mean subtracted inputs. In contrast to this, MIA constrains the space of solutions to the span of the original (not mean subtracted) inputs. The reason for this is that mean subtraction reduces the rank of $\boldsymbol{S}$ for linear independent inputs. However, in a high–dimensional space it can be assumed that the inputs are linearly independent. The mean subtracted versions can no longer fully represent the original inputs because rank $(\boldsymbol{S})$ is equivalent to the dimensionality of the spanned space. This phenomenon and the resulting difference between MIA and MCA are illustrated in Figure 4.3.

In Section 4.1, a solution to the MIA problem is derived. Furthermore, it is proven that MIA has a unique solution, up to its sign, for linearly independent inputs. Section 4.2 demonstrates that an alternative unconstrained MIA criterion can be found. This criterion is used to illustrate MIA properties. In Section 4.3, a regularised MIA version is derived to ensure the stability of the MIA result. Thereafter, Section 4.4 discusses MIA from a Bayesian point of view which is used to generalise the MIA solution. Thus additional prior knowledge and constraints can be used. Section 4.5 shows how MIA can be constrained to different spaces than the span of the input. Subsequently, Section 4.6 interprets MIA from the point of linear and kernel ridge regression. Finally, Section 4.7 summarizes the findings of this chapter.

4.1 Solution to MIA

This section discusses the solution to the MIA problem defined in equation (4.3). Furthermore, it is shown that MIA has, up to its sign, a unique solution for linearly independent inputs. In the following, let $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N] \in \mathbb{R}^{D\times N}$ represent a set of N inputs of dimensionality D. Moreover, let $\boldsymbol{\mu} = \frac{1}{N}\boldsymbol{X} \cdot \underline{\mathbf{1}}$ and $\boldsymbol{Y} = [\boldsymbol{x}_1 - \boldsymbol{\mu}|...|\boldsymbol{x}_N - \boldsymbol{\mu}] = \boldsymbol{X} - \boldsymbol{\mu} \cdot \underline{\mathbf{1}}^T$ denote the D-



(a) MIA and MCA ($D = 2; N = 11$) (b) MIA and MCA (projected space with $D > 2; N = 2$)

FIGURE 4.3: Comparison of MCA and MIA. (a) Results of MIA and MCA in a space of linearly dependent inputs. Note that the span of this input example is not reduced by mean subtraction and that both methods find the same result. (b) Simulation of the results in a high–dimensional input space where the inputs are linearly independent. The unconstrained MCA approach does not have a unique solution if rank $(\boldsymbol{S}) < D - 1$. It is assumed that MCA points in the first unique direction of minimum variance. Note that because of the mean subtraction inflicted rank reduction, MIA has an additional dimension for the optimisation procedure. Hence, MIA finds a different result than MCA.

dimensional mean and the set of mean subtracted inputs respectively. Therefore, the MIA problem from equation (4.3) can also be written as:

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}, \|\boldsymbol{w}\|=1, \boldsymbol{w}=\boldsymbol{X}\cdot\boldsymbol{c}}{\arg\min} \left\| \boldsymbol{w}^T \cdot \boldsymbol{Y} \right\|^2 \tag{4.4}$$

Note that the original space of the inputs spans the mean subtracted space plus possibly one additional dimension. Indeed, the mean subtracted inputs, which are linear combinations of the original inputs, sum up to zero. Mean subtraction cancels linear independence resulting in a one dimensional span reduction.

Theorem 4.1. *The minimum of the criterion in equation (4.4) is zero if the inputs $\boldsymbol{x}_i$ are linearly independent.*

Proof sketch: If inputs are linearly independent and span a space of dimensionality $N \leq D$, then the subspace of the mean subtracted inputs in equation (4.4) has dimensionality $N - 1$. There exists an additional dimension in $\mathbb{R}^N$, orthogonal to this subspace. Thus, the scatter of the mean subtracted inputs can be made zero. The existence of a solution where the criterion in equation (4.4) becomes zero is indicative of an invariance property of the data. ■

Using the projection matrix $\boldsymbol{P}$, the mean subtraction procedure can be simplified to:

$$\boldsymbol{Y} = \boldsymbol{X} \cdot \boldsymbol{P} \quad \text{with} \quad \boldsymbol{P} = \boldsymbol{I} - \frac{1}{N} \underline{\mathbf{1}} \ . \tag{4.5}$$

Obviously, $\sum_{i=1}^{N} \boldsymbol{y}_i = \underline{\mathbf{0}}$. Hence, the nullspace $\mathcal{NULL}(\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N)$ is non–trivial. All vectors $\boldsymbol{w} \in \mathcal{NULL}(\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N)$ will minimise $\tilde{\boldsymbol{S}}(\boldsymbol{X}|\boldsymbol{w})$. The next theorem shows that the problem given by equation (4.4) has, up to its sign, exactly one solution.

Theorem 4.2. *Assume $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N$ are linearly independent. Then, there exists an up to its sign unique $\boldsymbol{w} \neq \underline{\mathbf{0}}$ in $\mathcal{NULL}(\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N)$ such that $\boldsymbol{w}$ is in the span of the inputs $\boldsymbol{x}_i$, $i = 1, \ldots, N$.*

Proof: A solution $\boldsymbol{w} \neq \underline{\mathbf{0}}$ and $\boldsymbol{c} \in \mathbb{R}^N$ of the system of equations:

$$\boldsymbol{w}^T \cdot \boldsymbol{Y} = \underline{\mathbf{0}} \tag{4.6}$$

$$\boldsymbol{w} = \boldsymbol{X} \cdot \boldsymbol{c} \tag{4.7}$$

will also satisfy the Theorem 4.2 and solve the optimisation criterion of the problem in equation (4.4) given $\|\boldsymbol{w}\| = 1$. Indeed, equation (4.6) is equivalent to the existence of $\boldsymbol{w}$ such that $\boldsymbol{w} \in \mathcal{NULL}(\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N)$ and equation (4.7) specifies that $\boldsymbol{w}$ is in the span of the inputs $\boldsymbol{x}_i$ and $\boldsymbol{w} \neq \underline{\mathbf{0}}$. By substitution of $\boldsymbol{w}$ from equation (4.7) and $\boldsymbol{Y}$ from equation (4.5) in (4.6), the problem can be condensed to:

$$\boldsymbol{c}^T \cdot \boldsymbol{X}^T \cdot \boldsymbol{X} \cdot \boldsymbol{P} = \underline{\mathbf{0}} \tag{4.8}$$

Given that $\boldsymbol{G} = (\boldsymbol{X}^T \cdot \boldsymbol{X})$ is a Gram matrix formed by linearly independent vectors $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N$, $\boldsymbol{G}$ is invertible (see Theorem 7.2.10 in Horn and Johnson 1999). Let:

$$\boldsymbol{c}^T = \boldsymbol{d}^T \cdot (\boldsymbol{X}^T \cdot \boldsymbol{X})^{-1} \tag{4.9}$$

Therefore, equation (4.8) becomes: $\boldsymbol{d}^T \cdot \boldsymbol{P} = \underline{\mathbf{0}}$ with $\boldsymbol{d} = \zeta \, \underline{\mathbf{1}}$ and $\zeta \in \mathbb{R}$. When substituting this into equation (4.9) we obtain: $\boldsymbol{c} = \zeta \, (\boldsymbol{X}^T \cdot \boldsymbol{X})^{-1} \cdot \underline{\mathbf{1}}$. Hence:

$$\hat{\boldsymbol{w}} = \zeta \, \boldsymbol{X} \cdot (\boldsymbol{X}^T \cdot \boldsymbol{X})^{-1} \cdot \underline{\mathbf{1}} \tag{4.10}$$

It follows that $\frac{\hat{\boldsymbol{w}}}{\|\hat{\boldsymbol{w}}\|}$ is, up to its sign, a unique solution to equation (4.4). ■

It can be easily seen that additive components in all data samples will not affect the scatter matrix $\boldsymbol{S}$ and therefore will not affect the composition of the MIA solution. MIA captures invariant information present in all samples.

4.2 Alternative MIA Solution

The assumption that the MIA result is in the span of the inputs gives a possibility to find a, up to its sign, unique and closed form solution that minimises the scatter of the inputs. Although the resulting constrained optimization problem in equation (4.3) is well interpretable geometrically, it is difficult to analyse its properties. In the following, an alternative, unconstrained MIA solution is derived. This result aids the analysis of MIA properties and reiterates the choice of the span constraint. Moreover, the alternative approach relates MIA to FLDA and CCA that were discussed in Section 2. This connection to statistically well understood methods helps the interpretability of MIA and increases the confidence in its result.

As discussed earlier, it is assumed that the input samples are from a single class. However, the cost function of FLDA in equation (2.7) is defined using the between class scatter. For the single class case, the between class scatter is zero. Therefore, FLDA can not be used and a direct link from MIA to FLDA can not be established. However, the FLDA–like formulation of the CCA problem in Section 2.6 can be modified to extract an invariant signal from inputs of a single class. Here, $\boldsymbol{Z}$ is defined as classification table from equation (2.14). One interpretation of CCA is from the point of view of the cosine angle between the (non mean subtracted) vectors $\boldsymbol{a}^T \cdot \boldsymbol{X}$ and $\boldsymbol{Z}^T \cdot \boldsymbol{b}$. The aim is to find a vector pair that results in a minimum angle. A modified CCA (MCCA) criterion will be used as follows: First, consider the original inputs rather than the mean subtracted covariance matrices; Second, the class membership table $\boldsymbol{Z}$ for data from a single class collapses to a vector and $\boldsymbol{b}$ to a scalar, therefore $\boldsymbol{Z} \cdot \boldsymbol{b} = \underline{1} \cdot \boldsymbol{b}$. Thus, criterion (2.11) becomes independent of $\boldsymbol{b}$ resulting in:

$$\hat{\boldsymbol{a}}_{\mathrm{MCCA}} = \arg\max_{\boldsymbol{a}} J(\boldsymbol{a}) = \arg\max_{\boldsymbol{a}} \frac{\boldsymbol{a}^T \cdot \boldsymbol{X} \cdot \underline{1}}{\sqrt{\boldsymbol{a}^T \cdot \boldsymbol{X} \cdot \boldsymbol{X}^T \cdot \boldsymbol{a}}} \tag{4.11}$$

This criterion is maximised when the correlation of $\boldsymbol{a}$ with all inputs $\boldsymbol{x}_i$ is as uniform as possible. The solution to this problem can be found by:

$$\frac{\partial J(\boldsymbol{a})}{\partial \boldsymbol{a}} = \boldsymbol{X} \cdot \underline{1} - \boldsymbol{a}^T \cdot \boldsymbol{X} \cdot \underline{1} \cdot \left(\boldsymbol{a}^T \cdot \boldsymbol{X} \cdot \boldsymbol{X}^T \cdot \boldsymbol{a}\right)^{-1} \cdot \boldsymbol{X} \cdot \boldsymbol{X}^T \cdot \boldsymbol{a} = \underline{0} \tag{4.12}$$

Therefore, $\alpha\, \boldsymbol{X} \cdot \underline{1} = \boldsymbol{X} \cdot \boldsymbol{X}^T \cdot \boldsymbol{a}$ with $\alpha = \frac{\boldsymbol{a}^T \cdot \boldsymbol{X} \cdot \boldsymbol{X}^T \cdot \boldsymbol{a}}{\boldsymbol{a}^T \cdot \boldsymbol{X} \cdot \underline{1}}$. Furthermore

$$\begin{aligned}
\hat{\boldsymbol{a}} &= \lim_{\varepsilon \to 0} \left(\alpha \left(\boldsymbol{X} \cdot \boldsymbol{X}^T + \varepsilon\, \boldsymbol{I}\right)^{-1} \cdot \boldsymbol{X} \cdot \underline{1}\right) \\
\hat{\boldsymbol{a}} &= \lim_{\varepsilon \to 0} \left(\alpha \left(\boldsymbol{X} \cdot \boldsymbol{X}^T + \varepsilon\, \boldsymbol{I}\right)^{-1} \cdot \boldsymbol{X} \cdot \boldsymbol{X}^T \cdot \boldsymbol{X} \cdot \left(\boldsymbol{X}^T \cdot \boldsymbol{X}\right)^{-1} \cdot \underline{1}\right) \\
\hat{\boldsymbol{a}} &= \alpha\, \boldsymbol{X} \cdot \left(\boldsymbol{X}^T \cdot \boldsymbol{X}\right)^{-1} \cdot \underline{1}
\end{aligned} \tag{4.13}$$

Note that α is a scalar that results in scale independent solutions and $\varepsilon\, \boldsymbol{I}$ enables the inversion of the rank deficient $\boldsymbol{X} \cdot \boldsymbol{X}^T$. As can be seen, the solution equation (4.13) of the modified CCA problem in equation (4.11) is identical to the MIA solution in equation (4.10). Thus, one can argue that the MCCA and MIA criteria are equal. In the following, we rename $\hat{\boldsymbol{a}}_{\mathrm{MCCA}}$ to $\hat{\boldsymbol{a}}_{\mathrm{MIA}}$.

The new formulation of MIA in equation (4.11) highlights its properties:

Corollary 4.3. *The MIA problem has no defined solution if the inputs are zero mean i.e., if* $\boldsymbol{X} \cdot \underline{\mathbf{1}} = \underline{\mathbf{0}}$,
This is obvious from equation (4.11).

Corollary 4.4. *Any combination* $\hat{\boldsymbol{a}}_{\mathrm{MIA}} + \boldsymbol{b}$ *with* $\boldsymbol{b}$ *in the nullspace of* $\boldsymbol{X}$ *is also a solution to equation (4.11).*
This means that only the component of $\boldsymbol{a}$ that is in the span of $\boldsymbol{X}$ contributes to the criterion in equation (4.11).

Corollary 4.5. *The solution of equation (4.11) is not unique if the* N *inputs* $\boldsymbol{X}$ *do not span the* D*-dimensional space* $\mathbb{R}^D$.
This follows from corollary 4.4. A unique solution can be found by further constraining equation (4.11). One such constraint is that $\boldsymbol{a}$ be a linear combination of the inputs $\boldsymbol{X}$:

$$\hat{\boldsymbol{a}}_{\mathrm{MIA}} = \underset{\boldsymbol{a}, \boldsymbol{a}=\boldsymbol{X}\cdot\boldsymbol{c}}{\arg\max} \frac{\boldsymbol{a}^T \cdot \boldsymbol{X} \cdot \underline{\mathbf{1}}}{\sqrt{\boldsymbol{a}^T \cdot \boldsymbol{X} \cdot \boldsymbol{X}^T \cdot \boldsymbol{a}}} \tag{4.14}$$

Corollary 4.6. *The MIA solution reduces to the mean of the inputs in the special case when the covariance matrix* $\boldsymbol{C_{XX}}$ *has one eigenvalue* λ *of multiplicity D, i.e.,* $\boldsymbol{C_{XX}} = \lambda \boldsymbol{I}$.
Indeed, equation (4.14) can be rewritten as:

$$\hat{\boldsymbol{a}}_{\mathrm{MIA}} = \underset{\boldsymbol{a}, \boldsymbol{a}=\boldsymbol{X}\cdot\boldsymbol{c}}{\arg\max} \frac{\boldsymbol{a}^T \cdot \boldsymbol{\mu}}{\sqrt{\boldsymbol{a}^T \cdot \boldsymbol{C_{XX}} \cdot \boldsymbol{a} + \left(\boldsymbol{a}^T \cdot \boldsymbol{\mu}\right)^2}} \tag{4.15}$$

After normalising with $\boldsymbol{a} = \frac{\boldsymbol{X}\cdot\boldsymbol{c}}{\|\boldsymbol{X}\cdot\boldsymbol{c}\|}$ and using the spectral decomposition theorem (Moon and Stirling, 2000), it can be shown that $\boldsymbol{a}^T \cdot \boldsymbol{C_{XX}} \cdot \boldsymbol{a}$ is invariant to $\boldsymbol{a}$ given equal eigenvalues of $\boldsymbol{C_{XX}}$. The function under equation (4.15) is monotonically increasing in $\boldsymbol{a}^T \cdot \boldsymbol{\mu}$. Therefore, the optimum is obtained when $\frac{\boldsymbol{a}^T \cdot \boldsymbol{\mu}}{\|\boldsymbol{a}\|}$ is maximum resulting in $\hat{\boldsymbol{a}}_{\mathrm{MIA}} = \boldsymbol{\mu}$.

The invariance in data from one class results in a unique direction that is a characteristic feature of the data. This may capture information that is powerful enough to distinguish instances from different classes.

4.3 Regularisation of MIA

If the inputs $\boldsymbol{X} \in \mathbb{R}^{D\times N}$ are nearly collinear, the inverse $\left(\boldsymbol{X}^T \cdot \boldsymbol{X}\right)^{-1}$ in equation (4.13) becomes close to singular. In this case, small variations in the inputs, e.g., because of noise, can have a large effect on the result. This problem can be prevented by regularisation of the norm $\|\boldsymbol{a}\|$ enabling the computation of the inverse and increasing the stability of the solution. This

case parallels with ridge regression (discussed in Section 2.7.2). The regularised MIA criterion is given by:

$$\hat{\boldsymbol{a}}_{\mathrm{MIA}} = \arg\max_{\boldsymbol{a}} J(\boldsymbol{a}) = \arg\max_{\boldsymbol{a}} \left(\frac{\boldsymbol{a}^T \cdot \boldsymbol{X} \cdot \underline{\mathbf{1}}}{\sqrt{\boldsymbol{a}^T \cdot \boldsymbol{X} \cdot \boldsymbol{X}^T \cdot \boldsymbol{a}}} - \lambda \left\| \boldsymbol{a} \right\|^2 \right) \tag{4.16}$$

Note that components of $\boldsymbol{a}$ that are in the nullspace of $\boldsymbol{X}$ do not affect the result of equation (4.11) but are penalised in equation (4.16). The intuition is that the regularised MIA constrains its result to the span of the inputs. Similarly to equation (4.12), the solution to (4.16) is given as the root of the derivative:

$$\frac{\partial J(\boldsymbol{a})}{\partial \boldsymbol{a}} = \boldsymbol{X} \cdot \underline{\mathbf{1}} - \frac{1}{\alpha} \boldsymbol{X} \cdot \boldsymbol{X}^T \cdot \boldsymbol{a} - \frac{2\,\lambda}{\sqrt{\boldsymbol{a}^T \cdot \boldsymbol{X} \cdot \boldsymbol{X}^T \cdot \boldsymbol{a}}} \boldsymbol{a} = \underline{\mathbf{0}}$$

Here, α is an arbitrary scaling factor representing the scale invariance of equation (4.11). Let $\nu = \frac{2\,\lambda \sqrt{\boldsymbol{a}^T \cdot \boldsymbol{X} \cdot \boldsymbol{X}^T \cdot \boldsymbol{a}}}{\boldsymbol{a}^T \cdot \boldsymbol{X} \cdot \underline{\mathbf{1}}}$. Hence:

$$\begin{aligned}
\hat{\boldsymbol{a}} &= \alpha \left(\boldsymbol{X} \cdot \boldsymbol{X}^T + \nu\, \boldsymbol{I} \right)^{-1} \cdot \boldsymbol{X} \cdot \underline{\mathbf{1}} \\
\hat{\boldsymbol{a}} &= \alpha \left(\boldsymbol{X} \cdot \boldsymbol{X}^T + \nu\, \boldsymbol{I} \right)^{-1} \cdot \boldsymbol{X} \cdot \left(\boldsymbol{X}^T \cdot \boldsymbol{X} + \nu\, \boldsymbol{I} \right) \cdot \left(\boldsymbol{X}^T \cdot \boldsymbol{X} + \nu\, \boldsymbol{I} \right)^{-1} \cdot \underline{\mathbf{1}} \\
\hat{\boldsymbol{a}} &= \alpha\, \boldsymbol{X} \cdot \left(\boldsymbol{X}^T \cdot \boldsymbol{X} + \nu\, \boldsymbol{I} \right)^{-1} \cdot \underline{\mathbf{1}}
\end{aligned} \tag{4.17}$$

Similar to RR, ν is chosen empirically. Note that the result of equation (4.17) is close to the mean of the inputs if a high value is chosen for ν. The result of the regularised MIA is given by $\frac{\hat{\boldsymbol{a}}}{\|\hat{\boldsymbol{a}}\|}$. In Section 4.4, the empirical regularisation is replaced by a statistically motivated integration of prior knowledge.

4.4 Bayesian MIA Framework

In this section, a Bayesian viewpoint is taken to include prior knowledge into the MIA criteria from equations (4.3) and (4.14). This is desirable because real data samples do not contain an exactly equal amount of inherent structure. Thus, this uncertainty is modeled to obtain an improved result. Moreover, this different view on MIA enables the demonstration of further MIA properties i.e., the stability of the MIA result for minor variability in the inputs. The derivation of the Bayesian MIA formulation parallels the one discussed in Section 2.8. However, the data model and assumptions are changed to fit the MIA problem. In the following let $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N] \in \mathbb{R}^{D \times N}$, $\boldsymbol{w} \in \mathbb{R}^D$, $\boldsymbol{n} \in \mathbb{R}^N$ and $\boldsymbol{r} \in \mathbb{R}^N$. Therefore, the MIA equivalent to equation (2.23) can be found as:

$$\boldsymbol{r} = \boldsymbol{X}^T \cdot \boldsymbol{w} + \boldsymbol{n} \tag{4.18}$$

The intended meaning of $\boldsymbol{r}$ is the vector of observed projections of inputs $\boldsymbol{x}$ on $\boldsymbol{w}$, while $\boldsymbol{n}$ is measurement noise, e.g., $\boldsymbol{n} \sim \mathcal{N}(\underline{\mathbf{0}}, \boldsymbol{C_n})$. We assume $\boldsymbol{w}$ to be a random variable. Our goal is to estimate $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{\mu_w}, \boldsymbol{C_w})$ assuming that $\boldsymbol{w}$ and $\boldsymbol{r}$ are statistically independent. Ideally, the data $\boldsymbol{r} = \zeta\,\underline{\mathbf{1}}$ follows from the variance minimization objective if no noise is present and the variance of projections is zero, which is exactly the MIA criterion (as expressed in Theorem 4.1). We define a generalized MIA criterion (GMIA) applying an equivalent derivation to Section 2.8:

$$\begin{aligned} \boldsymbol{w}_{\mathrm{GMIA}} &= \boldsymbol{\mu_w} + \boldsymbol{C_w} \cdot \boldsymbol{X} \cdot \left(\boldsymbol{X}^T \cdot \boldsymbol{C_w} \cdot \boldsymbol{X} + \boldsymbol{C_f}\right)^{-1} \cdot \left(\boldsymbol{r} - \boldsymbol{X}^T \cdot \boldsymbol{\mu_w}\right) && (4.19) \\ &= \boldsymbol{\mu_w} + \left(\boldsymbol{X} \cdot \boldsymbol{C_f^{-1}} \cdot \boldsymbol{X}^T + \boldsymbol{C_w^{-1}}\right)^{-1} \cdot \boldsymbol{X} \cdot \boldsymbol{C_f^{-1}} \cdot \left(\boldsymbol{r} - \boldsymbol{X}^T \cdot \boldsymbol{\mu_w}\right) && (4.20) \end{aligned}$$

The GMIA solution, interpreted as a direction in a high dimensional space $\mathbb{R}^D$, aims to minimize the difference between the observed projections $\boldsymbol{r}$ considering prior information on the noise distribution. It is an update of the prior mean $\boldsymbol{\mu_w}$ by the current misfit $\boldsymbol{r} - \boldsymbol{X}^T \cdot \boldsymbol{\mu_w}$ times an input data $\boldsymbol{X}$ and prior covariance dependent weighting matrix. Equations (4.19) and (4.20) suggest various properties of MIA. First note that if $\boldsymbol{C_w} = \boldsymbol{I}$, $\boldsymbol{\mu_w} = \underline{\mathbf{0}}$ and $\boldsymbol{C_f} = \underline{\underline{\mathbf{0}}}$, equation (4.19) becomes identical to (4.10). In general it is desirable that the MIA representation be robust to small variations in $\boldsymbol{X}$ (e.g., due to noise). Equation (4.19) indicates that small variations in $\boldsymbol{X}$ do not have a large effect on the GMIA result. Indeed $\boldsymbol{w}_{\mathrm{GMIA}}$ is an invariant property of the class of inputs. Furthermore, equations (4.19) and (4.20) allow the integration of additional prior knowledge such as smoothness of $\boldsymbol{w}_{\mathrm{GMIA}}$ through the prior $\boldsymbol{C_w}$, correlation of consecutive instances $\boldsymbol{x}_i$ through the prior $\boldsymbol{C_f}$, etc. Moreover, the GMIA formulation can be used to define an iterative procedure tackling datasets with large N and D. In this case it might be unfeasible to compute the matrix inverse. This problem could be approached using the following procedure: First the data is split into subsets that can be handled computationally. Thereafter, $\boldsymbol{w}_{\mathrm{GMIA}}$ is extracted from the first subset. Following subsets are processed using the previous $\boldsymbol{w}_{\mathrm{GMIA}}$ as estimate for $\boldsymbol{\mu_w}$. Introducing a decreasing weight, the result converts to a MIA representation of the whole dataset. Note that the quality of such an iterative approach remains to be tested.

4.5 MIA in Constrained Function Space

In some applications, the components of the result are known *a priori* e.g., in spectroscopy the spectral distributions of the elements of interest may be known. However, the mixture may contain further structure due to noise or additional unconsidered components. Therefore, it may be advantageous to constrain the search for the mutual component to a specific space. Another reason for such a constraint is a high number of available inputs. Thus, it may be required to reduce the size of the inverse in equation (4.10) for computational reasons. In this case, the span of the inputs may be well represented by a subset of them. In the following it is shown how a change in the span constraint affects the MIA solution. Let $\boldsymbol{F} \in \mathbb{R}^{D \times K}$ denote the matrix of

K components that represent the new span constraint. Furthermore, let $\boldsymbol{X} \in \mathbb{R}^{D \times N}$, $\boldsymbol{w} \in \mathbb{R}^{D}$, $\boldsymbol{c} \in \mathbb{R}^{K}$ and $\boldsymbol{P} = \boldsymbol{I} - \frac{1}{N}\underline{\underline{\mathbf{1}}}$ represent the input data, the mutual component, the weighting vector and the mean subtracting projection matrix respectively. In parallel to Section 4.1:

$$\boldsymbol{w}^T \cdot \boldsymbol{X} \cdot \boldsymbol{P} = \underline{\mathbf{0}} \tag{4.21}$$

$$\boldsymbol{w} = \boldsymbol{F} \cdot \boldsymbol{c} \tag{4.22}$$

By substituting equation (4.22) in (4.21):

$$\boldsymbol{c}^T \cdot \boldsymbol{F}^T \cdot \boldsymbol{X} \cdot \boldsymbol{P} = \underline{\mathbf{0}} \tag{4.23}$$

If $\boldsymbol{X}^T \cdot \boldsymbol{F} \cdot \boldsymbol{F}^T \cdot \boldsymbol{X}$ is invertible, the weights can be found to $\boldsymbol{c}^T = \boldsymbol{d} \cdot (\boldsymbol{X}^T \cdot \boldsymbol{F} \cdot \boldsymbol{F}^T \cdot \boldsymbol{X})^{-1} \cdot \boldsymbol{X}^T \cdot \boldsymbol{F}$. Therefore equation (4.23) simplifies to $\boldsymbol{d} \cdot \boldsymbol{P} = \underline{\mathbf{0}}$ resulting in $\boldsymbol{d} = \zeta\, \underline{\mathbf{1}}^T$. Hence:

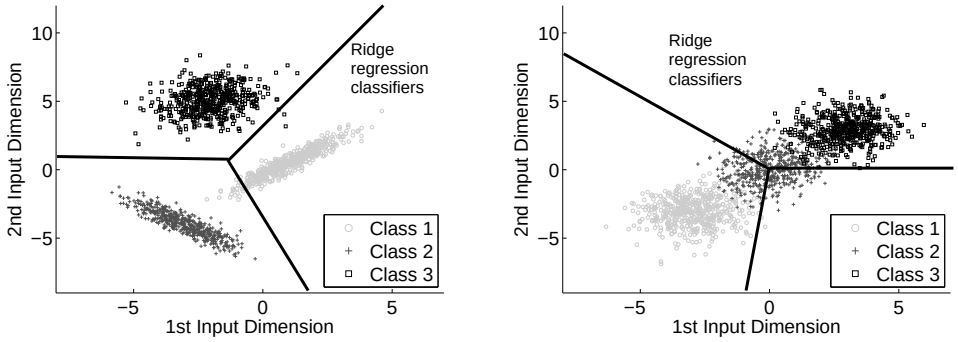
$$\hat{\boldsymbol{w}} = \zeta\, \boldsymbol{F} \cdot \boldsymbol{F}^T \cdot \boldsymbol{X} \cdot (\boldsymbol{X}^T \cdot \boldsymbol{F} \cdot \boldsymbol{F}^T \cdot \boldsymbol{X})^{-1} \cdot \underline{\mathbf{1}} \tag{4.24}$$

For the constraint basis $\boldsymbol{F}$, the MIA result is given by the normalized solution $\frac{\hat{\boldsymbol{w}}}{\|\hat{\boldsymbol{w}}\|}$. Note that this solution is not equivalent to MIA in a projected space if the basis $\boldsymbol{F}$ is not orthonormal. Furthermore, note that $\boldsymbol{F} \cdot \boldsymbol{F}^T$ has the same effect as $\boldsymbol{C}_{\boldsymbol{w}}$ in equation (4.19).

4.6 MIA as Regression

The MIA solution in equation (4.17) has striking similarities to ridge regression (RR) in equation (2.22). In this section, MIA is interpreted as regression to illustrate the similarities of both approaches. In a first step, linear and kernel ridge regression are discussed to familiarize important concepts. Commonly, regression is used to model an observed real valued output $\boldsymbol{y} \in \mathbb{R}^N$ as linear combination of N input vectors $\boldsymbol{x}_i \in \mathbb{R}^D$. The learned model $\hat{\boldsymbol{\beta}}_{\mathrm{RR}}$ can then be used to estimate future outputs $\hat{y}(\boldsymbol{x})$ given the data $\boldsymbol{x}$ as $\hat{y}(\boldsymbol{x}) = \boldsymbol{x}^T \cdot \hat{\boldsymbol{\beta}}_{\mathrm{RR}}$.

However, regression is also utilized for classification (Hastie et al., 2001, p. 81). Here, one aims to predict the posterior probability $p(y = k|\boldsymbol{x})$ that an input $\boldsymbol{x}$ is of class k. This probability is usually unknown during training. Thus, it is common to approximate the posterior probability for all class elements to 1 and for instances of other classes to 0. In this way, each ridge regression classifier aims to separate one class from all others. For more than two classes, this leads to ambiguous class membership regions. To prevent this, the linear classifiers are combined as described in Bishop (2006, p. 183). The advantage of the one–versus–the–rest approach is that, e.g., by excluding out of class instances, one can find a single class representation that is independent of other classes. Thus, not all classifiers have to be retrained



(a) Ridge regression classifiers

(b) Masking effect

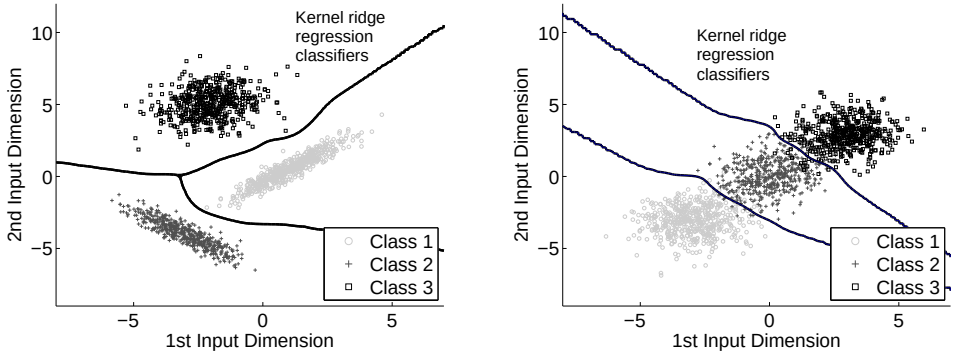
FIGURE 4.4: Linear classification using ridge regression. (a) Three classes of different covariance are separated in a two dimensional space. Note the missclassification on the boundary between classes 1 and 2. This is caused by the sensitivity of linear regression to class instances that are distant to the classification boundary. (b) Masking effect of linear regression. Each classification boundary aims to separate one class from all others (this is called one–versus–the–rest classifier). Therefore linear regression fails to separate class 2 from classes 1 and 3 using a single linear boundary. Note that the final boundaries are combinations of the one–versus–the–rest classifiers preventing ambiguous class membership regions.

if a new class is added. However, one–versus–the–rest based linear classification is not able to reliably separate multi–class problems in low dimensional spaces. This is exemplified in the right plot of Figure 4.4 and discussed below.

Figure 4.4 illustrates classification results using linear regression on three classes in a two dimensional space. In particular, a number of disadvantages of this linear approach are shown. For example, linear regression is sensitive to outliers. The points of class 1 that are most distant to the class 2 boundary have a large impact on its direction and position. This results in partial misclassification of class 1 in the left plot of Figure 4.4. Additionally, the estimate of the posterior probability $p(y = k|\boldsymbol{x})$ is not constrained to the interval $[0, 1]$ resulting in problems with its interpretability. Moreover, linear regression has problems with masking, e.g., if the centers of more than two Gaussian distributed classes are on one line. This results from the rigid nature of the linear boundary in combination with the one–versus–the–rest classification approach. Clearly, it is not possible to find a single linear boundary that separates class 2 from classes 1 and 3 in the right plot of Figure 4.4. As a result, most of the class 2 instances are missclassified.

In the following, it is shown how the lack of robustness to outliers as well as the masking problem can be prevented using a nonlinear extension of ridge regression. As discussed in Section 3.2, linear methods can be transformed to nonlinear ones by application of the kernel trick. The resulting kernel ridge regression algorithm is given in Shawe-Taylor and Cristianini (2004, p. 233) as follows:

$$\hat{y}(\boldsymbol{x}) = \sum_{i=1}^{N} a_i \, \boldsymbol{K}(\boldsymbol{x}_i, \boldsymbol{x}) \quad \text{with} \quad \boldsymbol{a} = (\boldsymbol{K} + \lambda \, \boldsymbol{I})^{-1} \cdot \boldsymbol{y} \ . \tag{4.25}$$



(a) Kernel ridge regression classifiers

(b) Robustness to masking

FIGURE 4.5: Nonlinear classification using kernel ridge regression with a Gaussian kernel of variance 0.2. (a) Classification of three classes with different covariances in a two dimensional space. Note that no training instance is missclassified. (b) Kernel ridge regression does not suffer from the masking effects of linear regression when using a Gaussian kernel. However, both kernel and linear ridge regression are one–versus–the–rest classifier.

As discussed in Hastie et al. (2001, p. 84), it is a loose rule for K–class problems that if $K \geq 3$ classes are lined up, it may be necessary to use polynomial terms of orders up to $K-1$ including their cross products to prevent masking in worst–case scenarios. That is, to resolve masking problems for $K = 10$ it could be necessary to use a polynomial kernel of degree 9. In the following, the Gaussian kernel

$$\boldsymbol{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}{2\,\sigma^2}\right)$$

is selected to represent a solution that is independent of the number of classes. The Gaussian kernel assumes that each class is well modeled by a combination of radial basis functions. The feature space of this kernel is of infinite dimensionality (Shawe-Taylor and Cristianini, 2004, p. 297) due to the infinite Taylor expansion of the Gaussian function. Kernel ridge regression classification results of the problems in Figure 4.4 are illustrated in Figure 4.5. Note that all training data is correctly classified in the left plot of Figure 4.5 demonstrating an increased robustness to outliers. Furthermore, the right plot of Figure 4.5 shows that kernel ridge regression does not suffer from masking problems. This demonstrates that although ridge regression has problems as a linear classifier, its nonlinear extension to kernel ridge regression is a powerful tool for classification. A drawback of kernel ridge regression to e.g., kernel support vector machines (KSVMs) (Shawe-Taylor and Cristianini, 2004) is that all training instances are used to determine the class of new inputs. By selecting a discriminant subset of the training data as support vectors, new instances can be classified faster using lower computational effort.

The relationship of MIA to the previously discussed regression methods is as follows. While linear and kernel ridge regression based classification commonly assume a one–versus–the–rest approach, MIA only considers instances of a single class. Furthermore, MIA assumes no *a priori* knowledge of the posterior probability $p(y = k|\boldsymbol{x})$ defining $\boldsymbol{y} = \underline{\mathbf{1}}$. MIA uses high

dimensional, linearly independent inputs and employs their Gram matrix $\boldsymbol{K}_{ij} = \boldsymbol{x}_i^T \cdot \boldsymbol{x}_j$ as kernel. That is, the output of MIA is constrained to the span of its inputs. Thus, the MIA result of the class k inputs represents a linear regression in the $N^{(k)}$–dimensional subspace of $\mathbb{R}^D$. The assumption of MIA that the input dimensionality D is much larger then their number N makes it difficult to illustrate MIA based classification similarly to the previously discussed regression approaches. That is, while kernel ridge regression only projects the data virtually to a high dimensional space, MIA starts off and remains in a high dimensional space. However, it is the intuition that both, disjointness of the subspaces spanned by instances of the classes k (with $\boldsymbol{X}^{(k)} \in \mathbb{R}^{D \times N^{(k)}}$) and l (with $\boldsymbol{X}^{(l)} \in \mathbb{R}^{D \times N^{(l)}}$) $\forall k \neq l$ as well as the high dimensionality of the spaces $N^{(k)} > K$ with $k, l = 1, \ldots, K$, ensure robustness and prevent masking. This can be visualized as follows. If two input samples are represented in a two dimensional space, the MIA result represents the direction where the projections of both points coincide. It is hypothesized that a projection onto the same point, e.g., from two inputs of a different class, is unlikely. Such masking is even less likely if $D >> N$ and one can assume that the subspace spanned by samples of one class is different to the one spanned by inputs from another class. Furthermore, as the inputs of a single class are linearly independent in their high dimensional input representation, each of them could be separated using a linear classifier. That is, not even samples of the same class mask each other in this high dimensional input space. Therefore, it is assumed, that inputs from different classes will not result in masking problems.

4.7 Summary

Mutual features were found that represent characteristic, high–dimensional patterns from each class. For instance, a mutual feature is a speaker signature under varying channel conditions or a face signature under varying illumination conditions. An algorithm called mutual interdependence analysis (MIA) was proposed to extract these signatures. By definition, the MIA signature is a linear combination of class examples that is equally correlated with all training samples in the class. It was proved that the MIA criterion has, up to its sign, a unique, closed form solution for linearly independent inputs. Thereafter, MIA was viewed from the perspective of a modified CCA problem. This unveiled certain properties of this approach, e.g., when the MIA criterion has a defined solution or is equivalent to the sample mean. Subsequently, a regularized MIA algorithm was proposed that enables the matrix inversion if inputs are nearly collinear. The MIA assumption of an equally present signature in the inputs of one class is not always suitable. For example, for speaker verification it can be assumed that a speech instance with silence contains a reduced degree of speaker characteristic information. This uncertainty in the actual degree of similarity was modeled using a Bayesian point of view. The analysis resulted in the generalized MIA criterion (GMIA). Additionally to modeling a misfit in the MIA assumption, GMIA can utilize *a priori* information on the mutual signature. This enables iterative applications of GMIA, e.g., to solve large size problems. The MIA criterion has, up to its sign, a unique result because of its constrained space of solutions. However, in certain

applications, it may be desirable to find solutions in a different, predefined space. Thus, it was shown how the MIA result can be constrained to a combination of alternate basis functions. Finally, MIA was interpreted from the point of linear and kernel ridge regression illustrating the similarities of the methods. This analysis suggested that MIA can be successfully used for classification without suffering from outliers or masking of classes.

Chapter 5

Synthetic MIA Examples

After the introduction and theoretical analysis of MIA and GMIA in Chapter 4, this chapter visualizes their behavior on synthetic data. The advantage of artificial data is its full controllability. This enables the simulation of states of interest and clear interpretation of the results. Furthermore, with exact *a priori* knowledge of the signal structure, the feature extraction results can be statistically analyzed. In the following chapters, GMIA is used for text–independent speaker verification and illumination–independent face recognition. Both approaches are different in their input data structure. That is, while speaker verification uses one–dimensional audio inputs, face recognition is done on two–dimensional images. One goal of this chapter is to analyze the properties of synthetic examples with both one- and two–dimensional data inputs. This aids the understanding and confidence in the approaches of the following chapters.

First, Section 5.1 compares the feature extraction results of GMIA with the mean, PCA and ICA using one–dimensional synthetic inputs. Thereafter, the extraction performance of GMIA is statistically analyzed showing when MIA, GMIA or the mean best represent a common component in the data. Section 5.2 demonstrates how MIA can be used to extract an illumination independent 'mutual face' from a synthetic mixture of differently illuminated face images. Section 5.3 summarizes the findings of this chapter.

5.1 One–Dimensional Input Examples

In this section, feature extraction is performed on synthetic data in order to visualize differences between MIA, GMIA and commonly used methods. A random signal model is defined to create synthetic problems. This way, the feature extraction results can be compared to the true feature desired. Assume the following generative model for input data $\boldsymbol{x}$:

$$\begin{aligned} \boldsymbol{x}_1 &= \alpha_1 \boldsymbol{s} + \boldsymbol{f}_1 + \boldsymbol{n}_1 \\ \boldsymbol{x}_2 &= \alpha_2 \boldsymbol{s} + \boldsymbol{f}_2 + \boldsymbol{n}_2 \\ &\vdots \\ \boldsymbol{x}_N &= \alpha_N \boldsymbol{s} + \boldsymbol{f}_N + \boldsymbol{n}_N \end{aligned} \tag{5.1}$$

where $\boldsymbol{s}$ is a common, invariant component or feature aimed to be extracted from the inputs, α_i, $i = 1, \ldots, N$ are scalars (typically all close to 1), $\boldsymbol{f}_i$, $i = 1, \ldots, N$ are combinations of basis functions from a given orthogonal dictionary such that any two $\boldsymbol{s}$ and $\boldsymbol{f}_i$ are orthogonal and $\boldsymbol{n}_i$, $i = 1, \ldots, N$ are Gaussian noises. In the following, it is shown that MIA estimates the invariant component $\boldsymbol{s}$.

Let us make this model precise. As before, D and N denote the dimensionality and the number of observations. Additionally, K is the size of a dictionary $\boldsymbol{B}$ of orthogonal basis functions. Let $\boldsymbol{B} = [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_K]$ with $\boldsymbol{b}_k \in \mathbb{R}^D$. Each basis vector $\boldsymbol{b}_k$ is generated as a weighted mixture of maximally J elements of the Fourier basis which are not reused ensuring orthogonality of $\boldsymbol{B}$. The actual number of mixed elements is chosen uniformly at random, $J_k \in \mathbb{N}$ and $J_k \sim \mathcal{U}(1, J)$. For $\boldsymbol{b}_k$, the weights of each Fourier basis element i are given by $w_{jk} \sim \mathcal{N}(0, 1)$, $j = 1, \ldots, J_k$. For $i = 1, \ldots, D$ (analogous to a time dimension) the basis functions are generated as:

$$b_k(i) = \frac{\sum_{j=1}^{J_k} w_{jk} \sin\left(\frac{2\pi i\, \alpha_{jk}}{D} + \beta_{jk} \frac{\pi}{2}\right)}{\sqrt{\frac{D}{2} \sum_{j=1}^{J_k} w_{jk}^2}}$$

with

$$\alpha_{jk} \in \left\{1, \ldots, \frac{D}{2}\right\}; \beta_{jk} \in \{0, 1\}; \{\alpha_{jk}, \beta_{jk}\} \neq \{\alpha_{lp}, \beta_{lp}\}\, \forall j \neq l \text{ or } k \neq p \ .$$

In the following, one of the basis functions $\boldsymbol{b}_k$ is randomly selected to be the common component $\boldsymbol{s} \in \{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_K\}$. The common component is excluded from the basis used to generate uncorrelated additive functions $\boldsymbol{f}_n$, $n = 1, \ldots, N$. Thus only $K - 1$ basis functions can be combined to generate the additive functions $\boldsymbol{f}_n \in \mathbb{R}^D$. The actual number of basis functions J_n is randomly chosen, i.e., similarly to J_k, with $J = K - 1$. The randomly correlated additive components are given by:

$$f_n(i) = \frac{\sum_{j=1}^{J_n} w_{jn} c_{jn}(i)}{\sqrt{\sum_{j=1}^{J_n} w_{jn}^2}}$$

with

$$\boldsymbol{c}_{jn} \in \{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_K\}; \boldsymbol{c}_{jn} \neq \boldsymbol{s}, \forall j, n; \boldsymbol{c}_{jn} \neq \boldsymbol{c}_{lp}, \forall j \neq l \text{ and } n = p \ .$$

Note that $\|\boldsymbol{s}\| = \|\boldsymbol{f}_n\| = \|\boldsymbol{n}_n\| = 1, \forall n = 1, \ldots, N$. To control the mean and variance of the

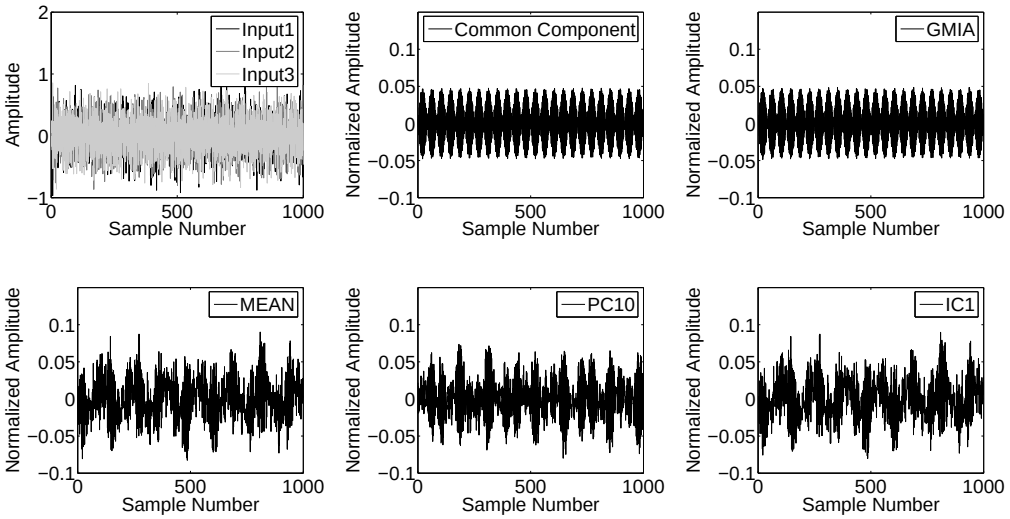


FIGURE 5.1: Comparison of results using various ubiquitous signal processing methods. Top left plot shows, for simplicity, only the first three inputs. The plots of principal and independent component analysis show particular components that maximally correlate with the common component $\boldsymbol{s}$. The GMIA solution turns out to represent the common component best, as it is maximally correlated to it.

norms of common, additive and noise component in the inputs, each component is multiplied by the random variable $a_1 \sim \mathcal{N}(m_1, \sigma_1^2)$, $a_2 \sim \mathcal{N}(m_2, \sigma_2^2)$ and $a_3 \sim \mathcal{N}(m_3, \sigma_3^2)$, respectively. Finally, the synthetic inputs are generated as:

$$\boldsymbol{x}_n = a_1 \boldsymbol{s} + a_2 \boldsymbol{f}_n + a_3 \boldsymbol{n}_n$$

with $\sum_{i=1}^{D} x_n(i) \approx 0$. The parameters of the artificial data generation model are chosen as $D = 1000$, $K = 10$, $J = 10$ and $N = 20$. The parameters of the distributions for a_1, a_2 and a_3 are dependent on the particular experiment and are defined correspondingly.

The GMIA solution is compared in Figure 5.1 (rightmost plot in top row) to the mean of the inputs as well as PCA and ICA results. The mixing model parameters are chosen as $m_1 = 1$, $m_2 = 10$, $m_3 = 0$, $\sigma_1 = 0.05$, $\sigma_2 = 0.05$ and $\sigma_3 = 0.05$. For simplicity, the GMIA parameters are $\boldsymbol{C_w} = \boldsymbol{I}$, $\boldsymbol{C_n} = \lambda \boldsymbol{I}$ and $\boldsymbol{\mu_w} = \underline{\mathbf{0}}$. This parameterization of GMIA by λ (the variance of the noise in model (5.1)), is denoted by GMIA(λ). Its solution represents the non regularized MIA when $\lambda = 0$ and the mean of the inputs when $\lambda \to \infty$. That is, for $\lambda \to \infty$, the inverse $\left(\boldsymbol{X}^T \cdot \boldsymbol{X} + \lambda \boldsymbol{I}\right)^{-1} \to \frac{1}{\lambda} \boldsymbol{I}$ simplifying the solution to $\boldsymbol{w}_{\mathrm{GMIA}} \to \frac{\zeta}{\lambda} \boldsymbol{X} \cdot \underline{\mathbf{1}}$, a scaled mean of the inputs.

PC10, the tenth principal component and IC1, the first independent component were hand selected due to their maximal correlation with the common component. Over all compared methods, GMIA extracts a signature that is maximally correlated to $\boldsymbol{s}$. All other methods fail to extract a signature similar to the common component.

In the following, MIA, GMIA and the sample mean are analyzed and compared in more detail. This is achieved by graphical representation of their results from a large number of randomly created synthetic problems, matching model (5.1), for various values of the variance of $\boldsymbol{n}_i$ (λ). Each column in Figure 5.2 represents a histogram of experimental results for a given value of λ (x-axis). The y-axis indicates the correlation of the GMIA solution with $\boldsymbol{s}$, the true common component. The colour of the point represents the number of experiments, in a series of random experiments, where this specific correlation value is obtained for the given λ. Overall 1000 random experiments are performed, with randomly generated inputs, using various values of λ.

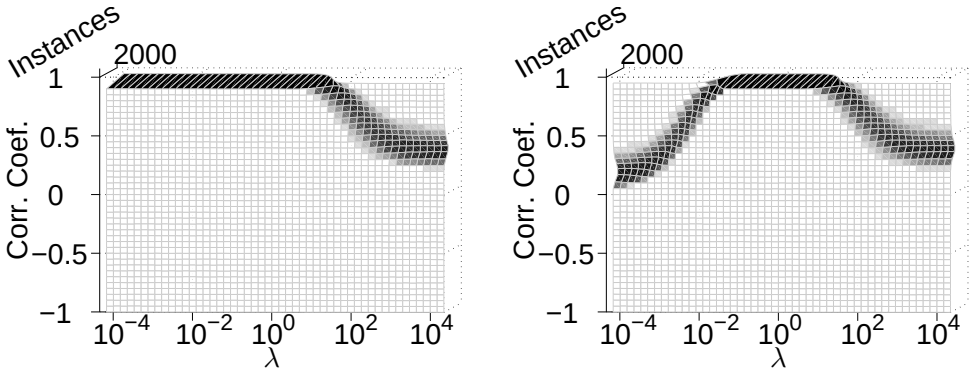
For all test cases in Figure 5.2, the weight of the additive noise is chosen as $a_3 \sim \mathcal{N}(0, 0.0025)$. There are experiments with three cases: (a) inputs contain equally a common component; (b) inputs contain approximately a common component; (c) inputs are approximately equal.

In Figure 5.2(a), the remaining mixing model parameters are chosen as $m_1 = 1$, $m_2 = 10$, $\sigma_1 = 0$ and $\sigma_2 = 0.05$. This situation fits the MIA assumption of an equally present component with an energy one order of magnitude smaller than the residue $\boldsymbol{f}_i + \boldsymbol{n}_i$. The results show that the common component is best extracted by MIA. In Figure 5.2(b), $m_1 = 1$, $m_2 = 10$, $\sigma_1 = 0.05$ and $\sigma_2 = 0.05$. This situation relaxes the strictly equal presence of the common component. Clearly, the simple MIA result and the mean do not represent $\boldsymbol{s}$. However, for some λ, GMIA succeeds in extracting the common component. Figure 5.2(c) illustrates the case $m_1 = 10$, $m_2 = 1$, $\sigma_1 = 0.05$ and $\sigma_2 = 0.05$. Here, all inputs are similar to the common component and therefore well represented by a signal plus noise model. The mean of the inputs is a good solution to this problem.

In summary, MIA and GMIA can be used to efficiently compute features in the data representing an invariant $\boldsymbol{s}$, or mutual feature to all inputs, whenever data fits the model from equation (5.1), even when the weight/energy of $\boldsymbol{s}$ is significantly smaller that the weight/energy of the other additive components in the model. Moreover, the computed feature $\boldsymbol{w}_{\mathrm{GMIA}}$ is radically different from the mean of the data in cases like (a) and (b) in Figure 5.2. The invariant feature $\boldsymbol{s}$ may have a physical interpretation of its own, depending on the problem as we will see next.

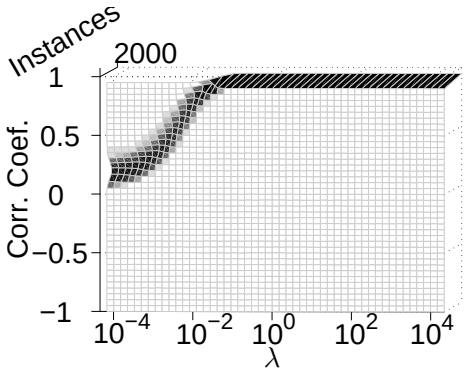
5.2 Two–Dimensional Input Examples

Face recognition on differently illuminated images is challenging. The reason is the high level of variance, in the gray levels of an image, that is accounted for by illumination information. For example if a face is illuminated from the left, large areas of the left face side are bright while most of the right face side is dark. In comparison, the fine structures are negligible that represent person dependent features such as a mole or the shape and position of the eyes, nose and mouth. Another difficulty is that illumination effects are dependent on the 3D information of the face. However, this information is usually not available causing difficulties to distinguish between face characteristics and shadows.



(a) Data fits MIA model

(b) Variation of $\boldsymbol{s}$



(c) High Intensity of $\boldsymbol{s}$

FIGURE 5.2: Statistical behavior of GMIA. Correlation of $\boldsymbol{w}_{\mathrm{GMIA}}$ and $\boldsymbol{s}$ for various λ values. Left vertical regions in the plots ($\lambda \rightarrow 0$) correspond to $\boldsymbol{w}_{\mathrm{GMIA}} = \boldsymbol{w}_{\mathrm{MIA}}$. Right vertical regions ($\lambda \rightarrow 10^4$) correspond to $\boldsymbol{w}_{\mathrm{GMIA}} \approx \boldsymbol{\mu}$, the mean of the inputs. (a) The common component intensity is invariant over the inputs and contributes little to their intensities. $\boldsymbol{w}_{\mathrm{MIA}}$ best represents the common component. (b) The common component intensity varies over the inputs with $\sigma_2 = 0.05$ and contributes little to their intensities. In this case, GMIA is preferable to MIA and the mean to learn a feature $\boldsymbol{w}_{\mathrm{GMIA}}$ that is best correlated with the common component. (c) The common component represents most of the energy in the inputs. In this case, the mean best represents the common component.

The intuition is that MIA captures an invariant from a set of inputs. Thus, the MIA representation of a face is also referred to as the mutual face. Is it possible to extract an illumination invariant mutual face from differently illuminated images? If this is the case, this mutual face can be used for illumination–invariant face recognition. For example, person characteristic features can be found without confusions caused by illumination conditions like shadows.

In the following, a synthetic model is defined that allows the artificial generation of differently illuminated faces. Thus, a large number of test cases can be generated enabling a statistical analysis of MIA for face recognition. Let the face be a Lambertian object (Foley et al., 1997, p. 723), where the object image has light reflected such that the surface is observed equally bright from different angles of the observer. Then, one can assume a face image $\boldsymbol{H}$ to be a linear combination of images from an image basis $\boldsymbol{H}_n$ with $n = 1, \ldots, K$ (Zhou et al., 2007):

FIGURE 5.3: Frontal images of the first person from the YaleB face database excluding the ambient and test image. The test image is illuminated frontally.

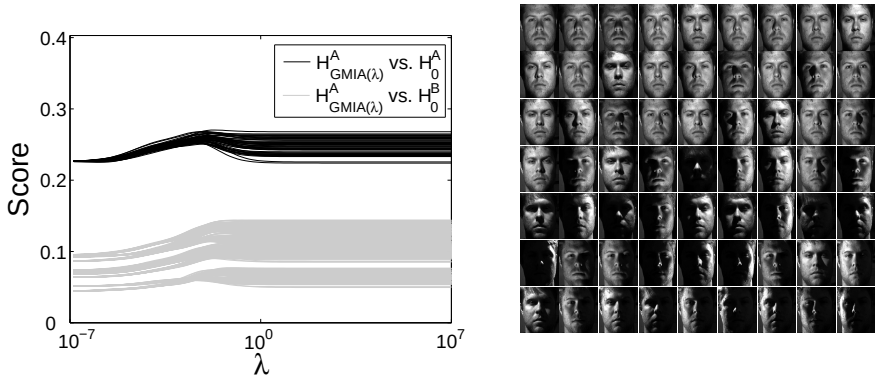
$$\boldsymbol{H} = \sum_{n=1}^{K} \alpha_n \boldsymbol{H}_n \tag{5.2}$$

where the α_n's are image weights. An appropriate set of basis images, to study illumination effects, is the YaleB database (Georghiades et al., 2001). This database contains 65 differently illuminated faces from 10 people and for 9 different camera angles to view a face. Each illuminated face image is obtained for a single light source at some unique but distinct position. In this report, we use only the frontal face direction but at various light source positions. The frontal illuminated faces are excluded from the basis and used as test images. Moreover, the images with ambient lighting conditions are excluded. The set of training images for the first person, A, of the YaleB database is illustrated in Figure 5.3. Additionally, the test image $\boldsymbol{H}_0^A$ of this person is shown in Figure 5.4(a).

Next, 20 images are synthetically generated as inputs to GMIA(λ). Each of these images is a combination of $J = 5$ randomly selected images $\boldsymbol{H}_i$ from the basis set $\boldsymbol{H}_n$. The basis images are combined according to equation (5.2) using weights $\alpha \sim \mathcal{U}(0, 1)$. To retain the image scaling: $\boldsymbol{H} = \frac{\sum_{i=1}^{J} \alpha_i \boldsymbol{H}_i}{\sum_{i=1}^{J} \alpha_i}$.

(a) Test Image $\boldsymbol{H}_0^A$ (b) MIA Image $\boldsymbol{H}_{\mathrm{MIA}}^A$

FIGURE 5.4: Images used for testing. (a) Frontal illuminated image of the first person from the YaleB face database. (b) Mutual image that is extracted from 20 randomly generated inputs. Each input is a combination of 5 randomly selected images of a person.



(a) Correlation result with $\boldsymbol{H}_{\mathrm{GMIA}(\lambda)}^A$ (b) Sorted YaleB Database

FIGURE 5.5: Results of synthetic MIA experiments with various illumination conditions. (a) Similarity scores of GMIA(λ) representation (mutual face) and the test image of the same and different people from the YaleB database in 50 random experiments. (b) Images of the YaleB database, ordered from high to low by their similarity score with the mutual face. The score becomes lower line after line from the top left to the bottom right.

An 'invariant' face signature is extracted to represent each person using MIA as follows. Images are 2D Fourier transformed. Thereafter, GMIA(λ) is separately computed for rows and columns resulting in $D = 250$ and $N = 20$. In a final step, GMIA representations for rows and columns are added and the data is processed using an inverse 2D Fourier transform to obtain a face signature of the person. This signature is called a mutual face and is e.g., denoted $\boldsymbol{H}_{\mathrm{MIA}}^A$ for person A. Figure 5.4(b) illustrates a GMIA representation that is generated using the previously described procedure.

A measure is defined to evaluate the similarity between test and GMIA images for the purpose of

face recognition. First, the images are filtered on their boundary. Second, the mean correlation scores of both images are computed separately for rows ($\mathcal{S}_1$) and columns ($\mathcal{S}_2$). A combined score is generated as: $\mathcal{S} = \frac{\sqrt{\mathcal{S}_1^2+\mathcal{S}_2^2}}{\sqrt{2}}$. Thus, the score is upper–bounded by the value one.

Now we test if MIA is able to capture illumination invariant facial features and can aid face recognition. Figure 5.5 illustrates results of synthetic MIA experiments with various illumination conditions. In particular, we show similarity scores between GMIA(λ) representations of 50 randomly generated input sets from person A and the test images from both A and other persons $B \neq A$. MIA (for $\lambda = 0$) results in an invariant image representation (all 50 scores are almost equal). Note that there is a λ dependent trade–off between the score value and the variance. For all cases of λ, the person A scores higher than person B. Figure 5.5(b) shows the training database from Figure 5.3 sorted by the score with the MIA representation (mutual face) of the same person. The score becomes lower line after line from the top left to the bottom right. The mutual face achieves the highest scores with evenly illuminated images, i.e., where the illumination does not distort the image.

Results support the intuition that the mutual image is an illumination–invariant representation of a set of images of one person. The MIA method will be used in a face recognition application described in a later chapter of this thesis.

5.3 Summary

In this chapter, MIA and GMIA were evaluated using synthetically generated data. Analyses were performed on both one- and two–dimensional inputs acting as pilot test for following experiments on real data. In a first step, a synthetic signal model was defined. GMIA was shown to extract an invariant component from the inputs that is hidden to ubiquitous methods such as PCA, ICA and the mean. Moreover, simulations were presented that demonstrate when and how MIA, GMIA or the mean represent a common signature in the inputs. MIA and GMIA work well even if the mutual signature accounts only for a small part of the signal energy. It was shown that GMIA continues to extract meaningful information if the common signature is not equally present in the inputs. Another mixing model was defined for differently illuminated face images. In this way, random illumination conditions were generated synthetically. MIA was shown to extract an illumination invariant ‘mutual face’. That is, MIA extracted a similar representation for different randomly generated input sets. This representation correlates maximally with evenly illuminated faces of the same class.

Chapter 6

Speaker Verification

Speech is not only a carrier of meaning but contains a large amount of information about the speaker. For instance it is possible to recognize the gender, language, dialect and emotional state of the speaker from only a few words. Furthermore, the identity of the speaker can be recognised or classified as unknown (Holmes and Holmes, 2001, p. 220). This information can be used in many applications e.g., for security, surveillance or human–machine interaction. Although it seems intuitive for a person to extract this information from speech of even unknown speakers, it is a challenging task for a machine. Reasons for this are the lack of prior information and the overall complexity of the task. For example, it is possible for a person to concentrate on a speaker in a noisy environment e.g., by knowledge of the surroundings, prior expectation of the disturbance nature, direction of the speech or visual inputs like lip reading. To counter these advantages, specialised machine models are designed for specific environments and tasks. Thus, the complexity can be reduced enabling comparable performance of the machine and the human (Schmidt-Nielsen and Crystal, 2000). However, it is the intuition that tasks involving e.g., a high amount of memory, speed, continuity or time can be performed better by a machine.

Possible constrained/specialised applications are identification and verification tasks. Verification compares the task–relevant features of a new speech utterance with previously learned ones of the claimed class. The goal is to decide if the class claim is correct or not. On the other hand, identification compares the features of the new utterance with all stored features in the database. The task is to decide the category of the utterance. Here, it is distinguished between open and closed set identification. While open set identification considers unknown categories, i.e., automatically creates a new class or rejects the utterance, closed set identification chooses the category with the highest score in the training set. For speaker identification/verification, one can further distinguish between text–dependent and text–independent approaches. While for text–dependent approaches the speaker has to input cooperatively a phrase that is known to the machine, text–independent approaches deal with any available information. Text–dependent identification/verification can be further classified into fixed–text, text–prompted and password–phrase methods. While text independence is preferable for, e.g., surveillance, the higher accuracy of text–dependent methods is advantageous for, e.g., security applications.

In this chapter, the application of MIA for text–independent speaker verification is discussed. As previously explained, the signal quality and background noise are major challenges in automated speaker verification. For example, telephone signals are nonlinearly distorted by the channel. In real applications, this distortion can be expected to vary for every connection. Furthermore, telephone signals are band–limited which cancels part of the speaker characteristic information. Moreover, background noise like wind, traffic or conversations of other people are challenges to modern speaker verification systems. The difficulty can be visualized by assuming the task where a person that is only known from a tape recording should be verified from a cellphone call with bad reception in a noisy pub. As noted in Schmidt-Nielsen and Crystal (2000), humans are robust to such changes in environmental conditions. In the following, the problem of background noise cancellation is disregarded. However, it is assumed that there exists no undistorted or equally distorted recording for the training or testing phase of the speaker verification system. The goal of MIA is to extract a signature that represents the speaker mutually in recordings from different nonlinear channels. Therefore, this feature represents the speaker but is invariant to the channels. The intuition is that this signature provides a robust feature for speaker verification in unknown channel conditions.

Section 6.1 provides the background in physical speech production for algorithmic concepts of later chapters. Thereafter, Section 6.2 gives a short introduction to prominent approaches for speaker verification. Section 6.3 discusses the choice and properties of the used databases. In Section 6.4, the preprocessing steps are motivated and explained. Section 6.5 covers the feature extraction step using MIA. Thereafter, Section 6.6 discusses the design and importance of a background model. Section 6.7 introduces the tools for evaluation and compares the results of MIA with alternative speaker verification approaches. Section 6.8 completes this chapter with its summary.

6.1 Speech Production Background

For the development and comprehension of speaker verification methods, it is important to understand the basic concepts of speech production. Therefore, this section discusses the physiological processes that are involved in speech production, introduces important terms and motivates a linear speech production model. The main organs that are involved in speech production are the lungs, larynx, pharynx, the nasal cavity and various parts of the mouth (Holmes and Holmes, 2001, p. 11). A cross–sectional view of a human head visualises these organs in Figure 6.1. Generally, sounds are acoustic pressure waves that are produced by modulation of the air flow between the lungs and the lips/nose. The source of this wave is expelled air from the lungs to the trachea that is forced through the vocal folds in the larynx. The opening between the vocal folds is known as the glottis. During inhalation, the glottis is fully opened allowing the air to fill the lungs. However, for sound production, the vocal folds are either periodically opening and closing (creating voiced sounds) or relaxed and open (for unvoiced sounds). In the following let $/ \cdot /$ denote a phoneme, the smallest linguistic unit,

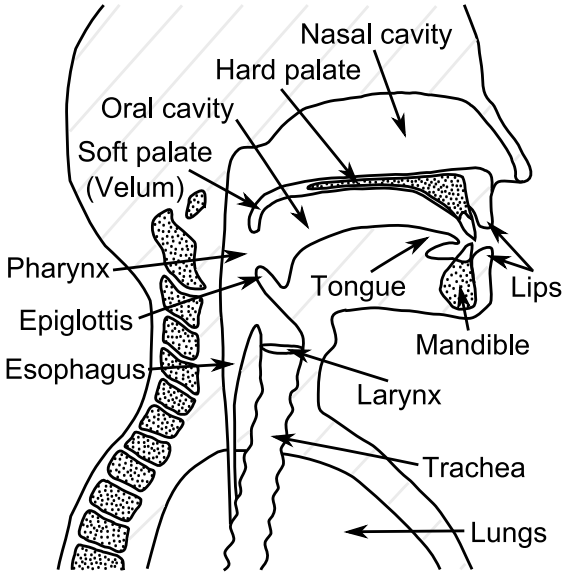
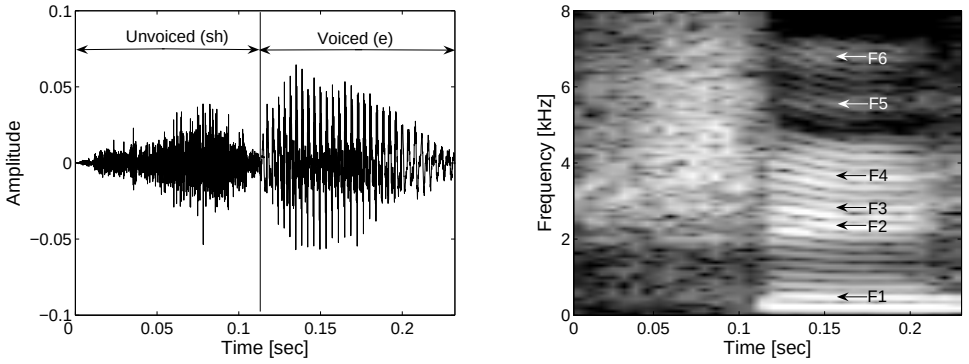


FIGURE 6.1: Cross–sectional view of a human head showing the anatomical structure of organs involved in speech production.

e.g., /f/ in *fill*. For voiced sounds like /i/ (in *she*), the vocal folds open and close cyclically because of muscle tension and the forced airflow. The vocal folds can be varied in length, thickness and position by muscular contraction. The frequency of the vibration, also called fundamental frequency (denoted f_0), is determined by these dimensions, the mass of the vocal folds and the air pressure from the lungs. This fundamental frequency lies typically in the region between 50-200 Hz for an adult male and about one octave higher for an adult female. In case the vocal folds are nearly closed but do not open and close cyclically, the airflow from the lungs is constrained creating a turbulent noise. Speech created in this way is perceived as whisper (O'Shaughnessy, 1987, p. 44). This case is similar for the production of other unvoiced sounds like /f/. Here, the vocal folds are open leaving the constriction of the airflow to the upper teeth that are pressed against the lower lip. A comparison of the waveform structures from voiced and unvoiced sounds is shown in Figure 6.2. In the case of swallowing, the epiglottis folds down directing food to the esophagus and preventing it from entering the trachea and lungs. The resonant structures between the larynx and lips/nose are also known collectively as the vocal tract. The acoustic energy that drives the vocal tract is the glottal volume velocity (Markel and Gray, 1980, p. 2). The volume velocity is a function of the subglottic air pressure and the time variation of glottal area, i.e., an increased pressure or a decreased acoustic impedance causes an increase in volume velocity.

In addition to the excitation effects, resonances in the vocal tract play an important role in the production of sounds. By muscular contraction, the structure of the vocal tract can be modified i.e., the position of the tongue changes the resonances in the oral cavity and the lowering of the velum adds the nasal cavity to the vocal tract resonance system. In this way, different sounds are formed from possibly constant glottal excitation. The resulting resonance frequencies are called formants. The formants are most prominent for voiced sounds. They are



(a) The word *she* in the time domain

(b) The word *she* in the time-frequency domain

FIGURE 6.2: Structure of voiced versus unvoiced sounds. (a) The unvoiced part / ∫ / of the word *she* appears like amplitude modulated noise. The voiced part /i/ has a clear periodic structure. (b) The time frequency representation of the same waveform unveils the formants of the voiced /i/. In contrast, the unvoiced sounds are smoothly structured over the whole frequency range lacking the horizontal line–structure of the voiced sounds. Note, in contrast to this example, there is not always a clear contour between the voiced and unvoiced sounds.

usually numbered in accenting order with their representing frequency as F1, F2, F3, etc. An example of formants and their notation is illustrated in Figure 6.2(b). The location and intensities of the formants are dependent on the physical characteristics of the speaker. Therefore, they are important features for speaker identification and verification. Only the subset of voiced sounds, containing the formant structure, is used for speaker verification in this chapter. The large structural variability of the vocal tract and the complex dampening of the different tissue densities including mandible, soft palate, hard palate, tongue, nasal cavity etc., make its exact modelling intractable. Additional modelling difficulties are introduced by coarticulation caused by transitions/overlap of phonemes. Therefore, the vocal tract characteristic is phoneme context–dependent cancelling the possibility to model each phoneme with a representing vocal tract model. Common simplified models assume the vocal tract as a string of filters or a combination of lossless tubes with different diameters. For example, let E, G, V and L represent the models of the excitation, glottal shaping, vocal tract and the lip radiation respectively in the z–transform domain. Then, as discussed in Markel and Gray (1980, p. 6), the speech S can be modelled as:

$$S = E \cdot G \cdot V \cdot L \tag{6.1}$$

Related to their vocal tract properties, sounds can be divided into two classes: vowels and consonants. Vowels like /a/ (in *stack*) are generated without major vocal tract restriction. On the other hand, consonants like /n/ (in *nice*) are typically generated with substantial restriction of the air flow in some part of the vocal tract. Therefore, while vowels can be modelled with a constant speaker dependent vocal tract, consonant models must represent the high number of possible vocal tract restrictions. For some consonants, the air flow is fully stopped for a

few milliseconds. These phonemes, like /p/ in *pack*, are called stop consonants or plosives because of the following rapid release of the air. Another group of consonants is called fricatives. Fricatives like /f/ are turbulent noises that are produced by forcing air through a tight channel i.e., the gap between the upper teeth and the lower lip or the tongue and the soft palate when they are pressed against each other. A phoneme family that includes both vowels and consonants is the nasal sounds. Nasalisation occurs when the velum is lowered allowing the nasal cavity to act as an additional resonator. In this case, the air and sound are partly or fully emitted though the noise. In English, nasal sounds like /n/ are commonly consonants. However, in other languages like French or Mandarin, nasal vowels are also linguistically relevant. An important linguistic property over all languages is that vowels and consonants are alternating i.e., it is uncommon that more than three vowels or consonants are consecutive. This prior knowledge about the phoneme structure of speech can be used, e.g., to aid the design of hidden Markov models (HMM's) (Baum and Petrie, 1966) in speech recognition.

The property of alternation between vowels and consonants is also used to define syllables, a combination of multiple phonemes, which are higher level phonological building blocks of words. Usually, a syllable is a combination of a vowel with possibly multiple consonants. Additional speech information is transmitted by emphasis, also called stress, of certain syllables in a word. This allows to distinguish identically spelled words with different meaning like *re-cord* and *rec-ord*. The stress can be provided by changes in pitch, loudness and timing. The pitch of a sound is closely related to its fundamental frequency. However, while the fundamental frequency is an accurate quantifiable measure (in Hz), the pitch is defined as an auditory attribute that allows the qualitative ordering of sounds on a scale from low to high. Thus the term pitch can be used for instance to describe relative speaker–independent changes in frequency that are characteristic for the pronunciation of a certain word. Commonly, real sounds are more complex containing harmonics, an integer multiple of the fundamental frequency. In this case, the most prominent frequency of a segment is considered its pitch. Comparable to the relation between pitch and fundamental frequency, the loudness is an subjective measure of the sound pressure level (in dB). The loudness, measured in phon, considers psychoacoustic effects like the frequency and timing dependent perception of sound intensity. These effects, illustrating the sensitivities of the human auditory system, are also used to increase the performance of speech processing approaches i.e., it is common to pre–emphasise significant frequencies in speaker recognition applications.

However, the speech content is not only recognised by the concatenation of phonemes using different stress and other acoustic features. Human languages are highly redundant methods of information transfer. Thus, even if not all 'phonemes', syllables or words are acoustically identified, it is typically possible to retrieve the meaning of the sentence. The prior information that is used for this retrieval is, e.g., the current topic of the conversation, knowledge of the speaker, the language, phonotactics (i.e., which speech sounds can combine with which other), grammatical structure, gesture and lip movement. Often, the correction or prediction of 'misunderstood' words is performed subconsciously. Prior knowledge also helps

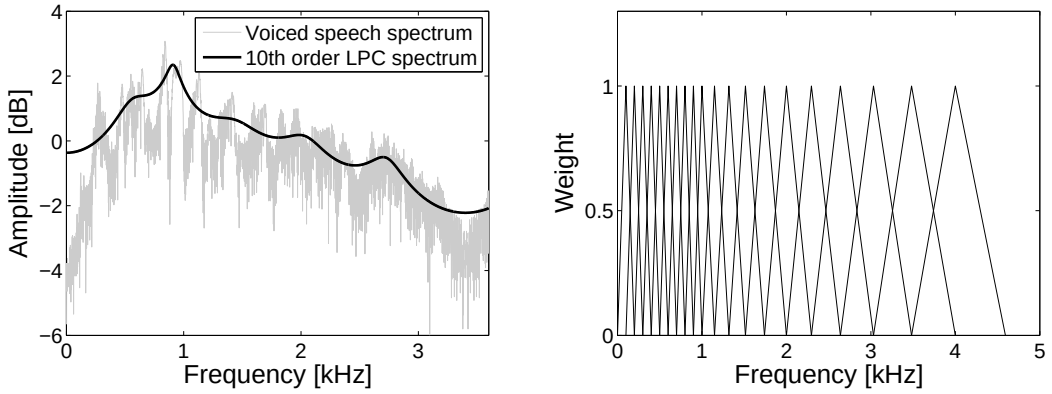
to identify/verify a person acoustically, i.e., a family member that calls every Sunday noon or a foreign friend that talks in the mother tongue are easily verified on bad telephone channels. Moreover, humans can recognise the direction of a sound using the different times of arrival, intensities, pinna reflections etc. in the ears. Thus, if the locations of all possible speakers are known, this ability can help to differentiate between them.

6.2 Introduction to Speaker Verification

This section provides a general background to a selection of common approaches used for speaker verification. It is intended to visualize the differences to procedures used in combination with MIA for text–independent speaker verification. Moreover, it gives an introduction to readers that are not familiar with the field of speech processing.

A prominent and established method for extraction of speaker–dependent information is based on linear predictive coding (LPC). An early application of this method to the field of speech analysis is discussed in Saito and Itakura (1968). LPC uses a weighted sum of previous signal values to predict future sample values. The weights are learned from the data using a minimum mean square error criteria. For speech analysis, LPC is commonly used as spectral estimator. In the Fourier domain, the LPC model can be viewed as an all–pole filter describing the spectral characteristic of a speech input. That is, the minimized error signal results by filtering the speech input with the inverse LPC model. Figure 6.3(a) illustrates the approximation of a spectral envelope using a 10th order LPC filter. The approximated data is one second of voiced speech from the NTIMIT database (Fisher et al., 1993) in the spectral domain that is band limited to 3.4 kHz. One property of LPC is that it models the peaks of the spectrum more accurately than the valleys. This is desirable because of its good correspondence to the human auditory perception (Deng and O'Shaughnessy, 2003, p. 47).

Ubiquitous speech features are the mel–frequency cepstral coefficients (MFCCs). The goal of the mel-cepstrum is to compress the spectral information of short speech segments such that critical auditory information is retained. Auditory experiments show that the sensitivity of the human ear is nonlinear with the frequency. Starting with 1 kHz, listeners were asked to define tones of equal distance in frequency. Based on these results, Stevens et al. (1937) define a mel–scale that is linear in the acoustic perception but nonlinear over frequencies. The conversion from the frequency f to the mel domain m is approximated as: $m = 1127 \log_e \left(1 + \frac{f}{700}\right)$. In a first step of the MFCC extraction, the input is windowed and transfered to the Fourier domain. It is common to use window sizes in the order of 20 ms to capture the phonetic information of speech. In the following, let $S(i, j)$ with $j = 1, \ldots, D$ and $i = 1, \ldots, N$ represent the windowed speech input from window number i of length D in the Fourier domain. Next, the acoustic perception information is utilized to compress the frequency spectrum using MFCC filters. This method was suggested by Davis and Mermelstein (1980). The used filters are triangular and linearly spaced in steps of 100 Hz below 1 kHz. The band edges are aligned



(a) Spectral estimation with LPC

(b) MFCC filters

FIGURE 6.3: Methods for modeling of spectral speech characteristics. (a) Approximation of the spectral envelope of a down–sampled voiced signal from the NTIMIT database using a 10th order LPC model. (b) MFCC filters that model the sensitivity of the human auditory system. The filters are equally spaced below 1 kHz and above follow the mel-scale.

with the center frequencies. In higher frequencies, the spacing between the band edges and center frequencies follow the logarithmic mel–scale. The 20 MFCC filters that were suggested in Davis and Mermelstein (1980) are illustrated in Figure 6.3(b). Note that, in this illustration, the spacing between the center frequencies above 1 kHz does not correspond to equidistant steps of 100 mel. The particular spacing is chosen dependent on the application, e.g., correspondent to bandwidth constraints of the signal. In the following, let $M(k, j)$ with $k = 1, \ldots, K$ and $j = 1, \ldots, D$ be the spectral representation of the MFCC filter number k. Then, the normalized spectral energy of the MFCC filtered inputs $\mathcal{E}_{mel}$ is given as:

$$\mathcal{E}_{mel}(i,k) = \frac{\sum_{j=1}^{D} \|M(k,j)\, S(i,j)\|^2}{\sum_{j=1}^{D} \|M(k,j)\|^2}$$

As discussed in Quatieri (2001, p. 714), the MFCCs are computed using a discrete cosine transform on the logarithmic $\mathcal{E}_{mel}$:

$$\text{MFCC}(i,j) = \frac{1}{K} \sum_{k=1}^{K} \log\left(\mathcal{E}_{mel}\,(i,k)\right) \cos\left(\frac{2\,\pi}{K}\, k\, j\right)$$

One disadvantage of the MFCCs is that the input signal is filtered twice using different filter characteristics. This has a negative effect on the spectral and temporal resolution of the result. A variation of the previously discussed mel–cepstral analysis is the sub–cepstrum. Here, the MFCC filters are directly used on the input signal. Further information on this method can be found in Erzin et al. (1995) and Quatieri (2001, p. 715).

Commonly, a statistical speaker model is learned from a large number of MFCCs to apply these

features to speaker verification and identification. Methods that are successfully used for these tasks are vector quantization (VQ) (Linde et al., 1980), Gaussian mixture models (discussed in Section 2.9) and support vector machines (SVMs) (Cortes and Vapnik, 1995; Shawe-Taylor and Cristianini, 2004).

6.3 Databases

One important component that aids the design of a speaker verification application is a database for training and testing. There are many desirable properties of such a database. For example, it is important that it is widely used to enable a comparison with other speaker verification methods or implementations. This can be ensured by using one of the standardized speech databases of the National Institute of Standards and Technology (NIST) or the Linguistic Data Consortium (LDC). While NIST is an United States government institute providing various scientific databases, LDC is an open consortium of universities, companies and government research laboratories that focuses on speech databases. Another desirable property of a database is usually that a large number of speakers from different regional and social backgrounds as well as of different gender are included. Thus, there is a higher confidence that the achieved verification results are representative for the method rather than the special case that is covered by the database. Assuming an application of the speaker verification in combination with the telephone network, it is desirable to use speech instances in the database containing different, realistic, nonlinear channel distortions and noise. However, in the design phase of a speaker verification approach, it may be preferable to exclude such nonlinear channels and focus on a simplified problem. Other requirements may be the amount of data that is available for training and testing, conversational speech, particular background noises, cocktail party recordings, healthy versus sick speaker, speech from different emotional states, availability of the spoken text or phoneme maps, presents of different languages, speech from kids etc.

Various portions of the NTIMIT database (Fisher et al., 1993) are used in this thesis. The NTIMIT database contains speech from 630 speakers (438 males and 192 females) that is nonlinearly distorted by real telephone channels. Thus, the intuition can be tested that MIA extracts an invariant, channel independent representation of each speaker. The database is relatively large and widely used enabling a comparison to other methods. NTIMIT is a distorted version of the TIMIT (Garofolo et al., 1993) database which is recorded with a high–quality microphone in a low noise studio environment. Therefore, the verification problem can be simplified using TIMIT. As previously discussed, this can be advantageous in the design phase of an algorithm. However, the invariant recording scenario of TIMIT is not desirable to test the assumption of MIA's invariant representation of one speaker in various channels. Thus, only experiments on NTIMIT are discussed. The databases are segmented into a test and train portion which both include different speakers. The reason for this is that TIMIT was designed for speech recognition and the acquisition of acoustic–phonetic knowledge. The test portion includes 168 speakers (112 males and 56 females) and is used to test smaller size speaker

verification problems. Additionally available information about the phoneme and text structure of the recordings is not utilized in this thesis. In both databases, each speaker is represented by 10 utterances that are subdivided into three text types: Type one represents two dialect sentences that are the same for all speakers in the database, type two contains five sentences per speaker that are in common with seven other speakers and type three includes three unique sentences. The segmentation of the data in a training and testing set is alternately done by sentence, i.e., the first sentence is used for training, the second for testing, the third for training etc. The resulting training and testing set includes a mix of all text types.

6.4 Preprocessing

As illustrated in equation (6.1), a speech signal can be modeled as an excitation that is convolved with linear dynamic filters. These filters represent various parts of the vocal tract. As discussed in Section 6.1, unvoiced sounds are generated using a noise–like excitation and various constrictions of the vocal tract. On the other hand, voiced sounds are generated at a constant location, the glottis, and propagate through a speaker dependent vocal tract. This results in a characteristic formant structure. Thus, by excluding silence and unvoiced sounds, each input instance can be assumed to contain similar speaker–dependent information. The detection of these non–voiced intervals is described in this section.

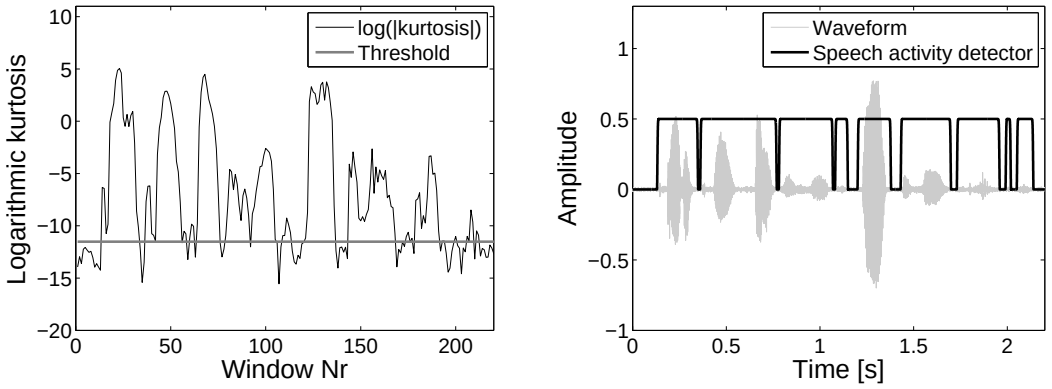
Let $E^{(p)}$, $G^{(p)}$, $V^{(p)}$ and $L^{(p)}$ represent the models of the excitation, glottal shaping, vocal tract and the lip radiation of person p respectively in the z–transform domain. Moreover, let SV and M represent the voiced speech and speaker–independent signal parts respectively in the z–transform domain (e.g., recording equipment, environment, etc.). The data can be modeled as: $SV^{(p)} = E^{(p)} \cdot G^{(p)} \cdot V^{(p)} \cdot L^{(p)} \cdot M$. By cepstral deconvolution, the model is represented as a linear combination of its basis functions, for each instance i:

$$\boldsymbol{x}_i^{(p)} = \log SV_i^{(p)} = \log E_i^{(p)} + \log G_i^{(p)} + \log V^{(p)} + \log L_i^{(p)} + \log M_i$$

This additive model suggests that we can use MIA to extract a signature that represents the speaker's vocal tract $\log V^{(p)}$. Several preprocessing steps are necessary to transform the raw data such that the additive model holds. To prevent cross interference, each speech utterance of the training and testing set is preprocessed separately.

6.4.1 Speech Activity Detection

As a first preprocessing step, silence and background noise are excluded from the wave data. Speech activity detection (SAD) is usually performed in speech coding or speech recognition to prevent unnecessary processing in non–speech intervals. There are various features used to



(a) SAD by threshold

(b) SAD result

FIGURE 6.4: Higher order statistics based SAD procedure. (a) Logarithmic kurtosis value of the waveform for half overlapping intervals of 20ms. Periods of more than two consecutive intervals under the threshold are classified as silence (b) Original waveform and the found segmentation between speech and non-speech.

indicate if speech is present in a wave segment e.g., zero–crossing–rate, low–band–energy, full–band–energy, cepstral coefficients, higher–order statistics etc. One advantage of these features is that they can be efficiently computed for short time intervals enabling a fine scale for the rejection of non–speech segments. However, SAD is a non–essential preprocessing step in the discussed off–line speaker verification approach on the NTIMIT database. It is primarily used to improve the speed of the following voiced speech segmentation. Therefore it is important to use a conservative SAD approach that ensures that no speech is cut. To achieve this, a threshold–based approach is used on the logarithmic absolute kurtosis values for 20 ms, half overlapping data intervals. The threshold is empirically chosen. If the values of more than two consecutive intervals fall below this threshold, all but the first and last interval are cut. The two retained intervals are exponentially smoothed preventing discontinuities at the cutting ends. This approach is illustrated in Figure 6.4 on a waveform example form the NTIMIT database. The logarithmic absolute kurtosis value was also used for speech activity detection in Li et al. (2005).

6.4.2 Voiced Speech Detection

In a second preprocessing step, the unvoiced speech segments are eliminated using a short–time autocorrelation (STAC) like approach. In the following, the STAC approach is reviewed to discuss its properties, motivate the used variation and aid a comparison of both methods. Let $w(k)$ represent a window function with nonzero elements for $k = 0 \ldots K-1$. The STAC, which is commonly used for voiced/unvoiced speech separation, is defined as (Deng

and O'Shaughnessy, 2003, p. 35):

$$\text{STAC}_n\,(i) = \sum_{m=-\infty}^{\infty} x\,(m)\,w\,(n-m)\,x\,(m-i)\,w\,(n-m+i)$$

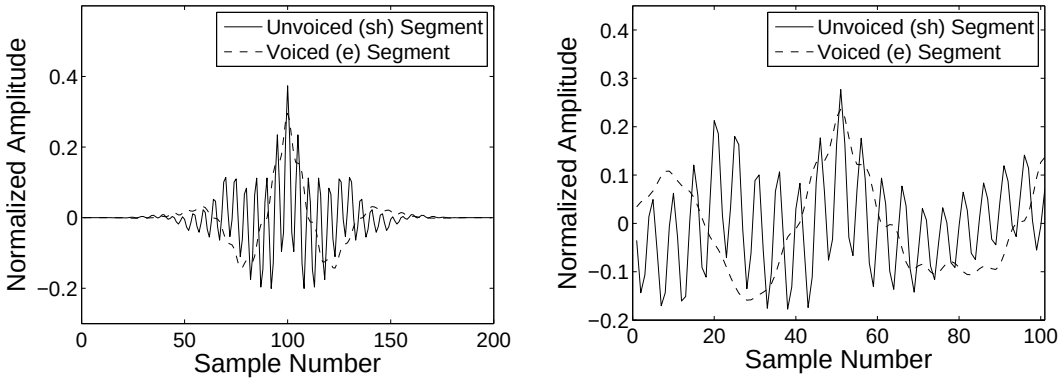
The range of the summation is limited by the window $w\,(k)$. Furthermore, $STAC$ is even, $STAC_n\,(i) = STAC_n\,(-i)$, and tends toward zero for $\|i\| \rightarrow K$. A disadvantage of this method is that, assuming a finite window function of length K, only the center of the STAC result is a combination of maximal K non–zero elements i.e., the overlap of the two window functions reduces as $\|i\|$ increases. The resulting filter effect makes it necessary to use long windows (Deng and O'Shaughnessy, 2003, p. 46). However, short windows are important to ensure accurate voiced/unvoiced segmentation. Thus, a different windowing procedure is employed that reduces this effect and prevents the trend toward zero. In the following, a Hann window function is used:

$$w\,(k) = \begin{cases} 0.5\left(1-\cos\left(\frac{2\pi k}{K-1}\right)\right), & \text{for } 0 \leq k \leq K-1 \\ 0, & \text{otherwise.} \end{cases}$$

The modified short–time autocorrelation (MSTAC) function is given by:

$$\text{MSTAC}_n\,(i) = \sum_{m=-\infty}^{\infty} x\,(m)\,w\,(m-n)\,x\,(m+i)\,w\,(m-n)$$

The result of this function is computed for $i = -\frac{K}{2}\ldots\frac{K}{2}$ and steps in n of size $\frac{K}{2}$. Note that in contrast to the STAC, these results are not necessarily even. However, quasi–periodic signals $x\,(m)$, e.g., voiced sounds, unveil their periodicity in this domain. The voiced and unvoiced segments are separated using an empirical decision function that compares the low and high frequency energies of each MSTAC result. That is, the input segment is assumed to be voiced if the low frequency energies outweigh the high frequencies and vice versa. For good resolution, a short window length can be chosen, e.g., 6.3 ms for the discussed examples. Note that this is possible because information of neighboring windows is utilized. Assuming this window length, quasi–periodicity frequencies of above approximately 160 Hz are captured. As discussed in Section 6.1, this is sufficient to capture most of the formants and a large part of fundamental frequencies. Figure 6.5 illustrates the STAC and MSTAC results for voiced and unvoiced sounds. In a final preprocessing step, the *a priori* knowledge is used that NTIMIT utterances are band limited by the used telephone channels. Thus, the voiced speech is down sampled to 6.8 kHz. The signal to noise ratio is increased by exclusion of noise above the speech channel.



(a) STAC Segments of the word *she*

(b) MSTAC Segments of the word *she*

FIGURE 6.5: STAC and MSTAC results of voiced versus unvoiced sound segments. (a) Unvoiced sounds are represented by a high frequency, voiced sounds by a low frequency STAC signal. (b) The MSTAC representations trade off the even signal shape against additional information content from neighboring instances. Note that for steps in n of size $\frac{K}{2}$, the end of one MSTAC segment equals the beginning of the next one.

6.5 Speaker Signature Extraction

For the design of a speaker characteristic feature, it is necessary to make assumptions regarding speaker–dependent information that is hidden in speech. A way to derive these assumptions is to analyze available information on the speech production process. However, this gives little information that can help with certain trade–offs, e.g., window length vs. frequency resolution, number vs. size of used frequency bands etc. One can assume that sound perception is an indicator to answers for some of these questions. This implies that the machine uses similar information to the human to distinguish between people. Although this is not necessarily true, the perceptual information can be used as a starting point.

In a psychophysical experiment designed to find the sensitivity of humans to changes in frequency, listeners were asked if two sounds of equal intensity differ. In the F1–F2 range, 70 % of the people noticed a difference of 2 Hz. This variation is defined as a just–noticeable difference (JND) (Deng and O'Shaughnessy, 2003, p. 241). JNDs represent the resolution of human audition and are therefore used, e.g., for coding.

Note that to find a speaker characteristic representation in the Fourier domain with a resolution of 2 Hz, it is necessary to use a window size of minimum 0.5 seconds. However, speech is clearly not well modeled as a constant signal. Thus, for example in a speaker recognition application using GMMs, the speaker is modeled in multiple states. The hypothesis is that GMMs represent a phoneme dependent map of a speaker. To distinguish between different phonemes, it is common to use window sizes of approximately 10 ms. This window size results in a frequency resolution of 100 Hz. The resolution is further reduced using MFCC filters. Clearly, there is an application dependent trade–off between those two extremes. For example,

one could imagine that for speech recognition, the time resolution is critical. On the other hand, to detect the invariant identity of a speaker, it may be advantageous to use a higher frequency resolution resulting in a larger window size. Possibly the performance of a speaker verification or identification system can be optimized using multiple window sizes to capture both time and frequency resolution.

In this section, the extraction of a common speaker invariant signature using MIA is discussed. The assumption is that each speech segment contains such a common characteristic. Obviously, this is not the case for infinitely small window lengths. Therefore, to show data segmentation effects, multiple experiments are performed where the data are windowed using different window sizes. Each utterance is segmented separately to comply with the data model in equation (6.4). An equal overlap between segments is introduced if more than half of a segment would be disregarded at the end of an utterance. This step limits the loss of signal energy for short utterances and long window sizes.

The segmented voiced speech $\boldsymbol{x}^{(p)}$ is nonlinearly transformed to fit the linear model in equation (5.1). In MIA, correlation coefficients are used as measure of similarity between two vectors. This measure is sensitive to outliers. Also, low signal values result in negative peaks in the logarithmic domain. A nonlinear filter and offset are used, before the logarithmic transformation, to reduce the focus on these signal parts. First, the inputs are transfered to the absolute of their Fourier representation. Second, each sample is reassigned with the maximum of its original and its direct neighboring sample values. Third, an offset is added to limit the sensitivity to low signal intensities that are affected by noise. The resulting signals are transfered to the logarithmic domain.

Speech has a speaker independent characteristic with maximum energy in the lower frequencies. As our goal is to extract signatures to distinguish speakers, it is sensible to disregard information that is common between them. Furthermore, the step prevents the effect illustrated in Figure 5.2(c). To achieve this, the mean of the original inputs of all speakers is decorrelated from them. The new inputs are then used to compute the final GMIA signatures for each speaker. The procedure used to extract GMIA speaker signatures is illustrated in Figure 6.6.

As in the artificial example, the GMIA parameters are $\boldsymbol{C_w} = \boldsymbol{I}$, $\boldsymbol{C_n} = \lambda\,\boldsymbol{I}$ and $\boldsymbol{\mu_w} = \underline{\mathbf{0}}$. Thus, the GMIA result is a weighted sum of the high dimensional inputs. For example, a window size of 250 ms and 10 seconds of speech data result in $D = 1700$ and $N = 40$. In the nonlinear logarithmic space it is not meaningful to subtract two features from each other. Therefore, the parameter λ is chosen to the smallest value that ensures positive weights. Note that in the limit ($\lambda \to \infty$), all weights are equal and positive.

A goal of the discussed speaker verification application is to evaluate the performance of a GMIA–based approach on real world data. Therefore, it is of interest to show results using only GMIA signatures. This enables a clear interpretation of GMIA without affects from preprocessing or other methods. However, for an implementation it is important to design a system that achieves 'optimal' results. This is commonly achieved by using preprocessed data

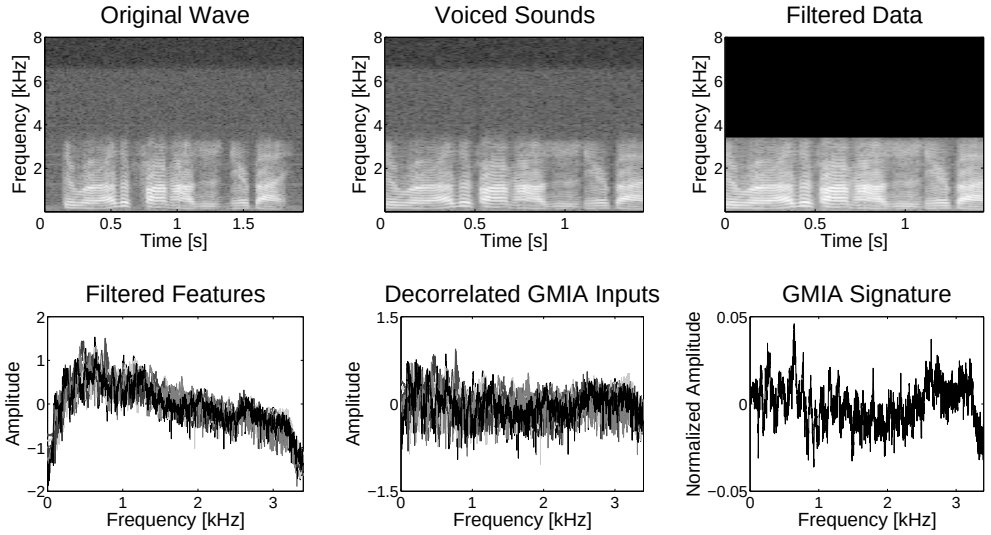


FIGURE 6.6: Processing chain for text–independent speaker verification using GMIA

and additional features. To satisfy both criteria, GMIA is analyzed alone and with additional components. For example, a pitch feature is used in combination with the GMIA signature to improved the overall system performance:

$$f_0^{(p)} = \underset{n}{\text{median}} \left(\arg\max_i \left(\text{abs} \left(\text{FFT} \left(\text{abs} \left(s_n \left(i \right)^{(p)} \right) \right) \right) \right) \right)$$

This simple pitch frequency detector can be motivated by a diode based envelope demodulation circuit. In a first step, the signal is rectified in the time domain using the absolute operator. Thereafter, the frequency is detected that contributes maximally to the signal energy. Here, the search over i is constrained to the realistic pitch interval 80–300 Hz (Deng and O'Shaughnessy, 2003, p. 31). In a final step, the median result over all input windows is selected to increase the robustness of the estimator.

6.6 Background Model

For speaker verification, it is necessary to define a similarity measure between the input and the learned features of the claimed identity. The goal is to compare the current score with a baseline that represents the minimum score necessary for the acceptance of the claimant. In this section, the procedure is described that is used to compute the similarity score from both pitch and GMIA features. Thereafter, the baseline score level is discussed that results from a speaker world model called the background model.

In a first step, the similarity scores of both GMIA signature and the pitch feature are computed separately. As discussed in Section 6.5, a database of features and correspondent identities is built from the training data and claimant features are extracted from the testing data. The GMIA

distance of a person from the training data and a claimant is given by the sum of square distances of their signatures. On the other hand, the pitch distance equals the norm one distance between the estimated f_0 values of both individuals divided by 100. The possible range of the GMIA distance is $[0, 4]$ and of the pitch distance is $[0, 2.2]$. This results because the GMIA signatures are norm one and the pitch detection is constrained to frequencies between 80–300 Hz. The distances are added to a single measure for an implementation with both GMIA and pitch feature. Otherwise, only the GMIA distance is used. To simplify the following discussions, the similarity score is defined as this distance measure with a negative sign.

Next, the design of the used background model is discussed. The standard approach is to find a model for all known impostors and compare the probability that the current instance is from the claimed identity with the probability that it is fitting the impostor model. A threshold is set that represents the minimum ratio between these probabilities that is necessary for acceptance. For noisy conditions, the performance of such a universal background model (UBM) can be improved by inclusion of noise. However, often it is not realistic to assume *a priori* knowledge of the particular kind of background noise or to model all possible backgrounds.

In this chapter, a different philosophy is followed. One difference is that not probabilities but scores are compared. Recollect that the goal of a background model is to act as comparison enabling the decision if the score between the input and the claimed identity is sufficient for acceptance. An input can be distorted by noise which results in a reduction of its maximum possible score with the undistorted features. Therefore, the goal is to find a comparison score that represents a high score given the quality of the input. To achieve this, the pair of GMIA signature and pitch features, from the training data, is used as background model that results in the highest score with the inputs. The speaker identity is accepted if the score of the input with the claimed speaker identity is larger than the maximum score minus a threshold. Note that at least one identity will be accepted for any input. This may be a problem for small databases and open set tests where no training data are present for an impostor. On the other hand, note that if the speaker is correctly identified in a speaker identification experiment, he/she will always be accepted with this speaker verification approach.

One disadvantage of the current implementation is that the scores have to be computed with all speakers in the database to perform verification. This is no problem because of the necessity to compute all scores for the following analysis of the approach. However, for an implementation in a product it is advantageous to select a subset of speakers in the training database as background speakers. The goal is to select speakers with different characteristics that represent the space of features. A similar selection procedure for the background model is suggested in Reynolds (1995).

6.7 Evaluation of Results

One area of critical importance is the evaluation of the results. All feature pairs (GMIA signature and pitch value) from the training and testing data are compared to enable a representative measure. This results in 396900 verification tests for a single experiment on the full NTIMIT database of 630 speakers. In this section, the used measures for the result evaluation are introduced. Thereafter the selection of the threshold described in Section 6.6 is motivated. Concluding, the results of GMIA for text–independent speaker verification are discussed.

Let P, CA, WA, IR, FAR and FRR denote the number of speakers in the database, number of correctly accepted speakers, number of wrongly accepted speakers, identification rate, false acceptance rate and false rejection rate respectively. The IR, FAR and FRR are given by:

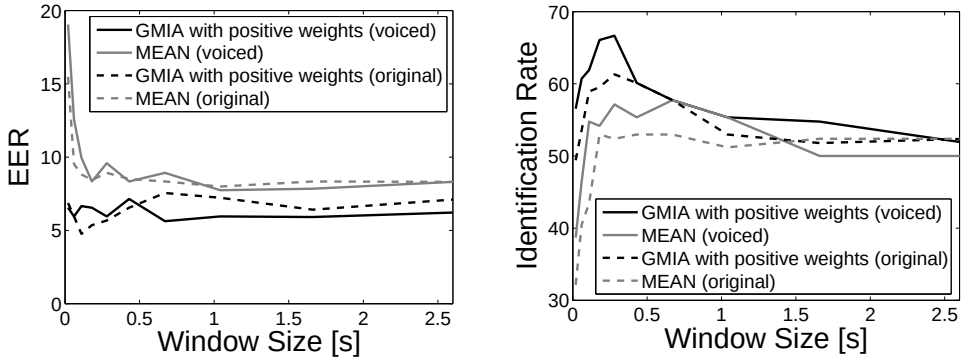
$$\begin{aligned} \text{IR} &= 100\,\frac{\text{CA}}{\text{P}}\,[\%] \\ \text{FRR} &= 100\left(\frac{\text{P}-\text{CA}}{\text{P}}\right)[\%] \\ \text{FAR} &= 100\left(\frac{\text{WA}}{\text{P}\,(\text{P}-1)}\right)[\%] \end{aligned}$$

The IR is of main interest for the evaluation of a speaker identification system. Here, the identity of the speaker with the highest score is assigned to the current input. On the other hand, in speaker verification, a speaker is accepted if the score between its own and the claimed identity signature $\mathcal{S}_C$ exceeds the one with a background speaker model $\mathcal{S}_B$ by more than a defined threshold Δ. In this way, different speakers from the database could be accepted for a single claimed identity. The following decision function is used to constrain the possible threshold values:

$$\frac{\mathcal{S}_C - \mathcal{S}_B}{\mathcal{S}_C + \mathcal{S}_B} < \Delta \tag{6.2}$$

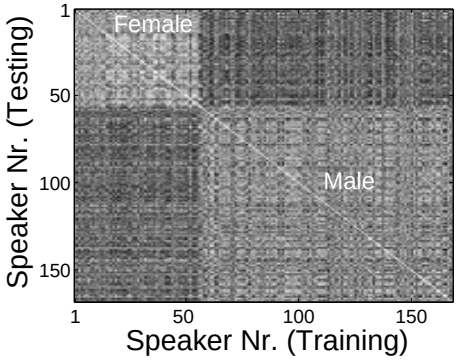
Note that $0 \geq \frac{\mathcal{S}_C - \mathcal{S}_B}{\mathcal{S}_C + \mathcal{S}_B} \geq 1 \quad \forall \quad \|\mathcal{S}_C\| \geq \|\mathcal{S}_B\|$. This is the case for the scores discussed in Section 6.6. Clearly, the threshold has a direct effect on the FRR and FAR. It is application–dependent if low FAR or FRR are desirable. For example, in a security–relevant application it is important to achieve low FAR preferably rejecting a correct speaker rather than accepting an impostor. However, to enable a comparison of different speaker verification approaches, it is common to select the threshold such that both FAR and FRR are equal. This point is called equal error rate (EER).

Figure 6.7(a) illustrates the results of the discussed speaker verification approach using the NTIMIT test portion of 168 speakers. Only GMIA signatures were used to exclude effects of the pitch feature. This enables a clear interpretation of the results. Additionally, window size



(a) Verification Results

(b) Identification Results



(c) Score Matrix

FIGURE 6.7: Comparison of speaker verification results using GMIA and mean features. Optimal performance is achieved for window lengths between 100 – 500 ms. Note that the performance drops sharply for shorter window sizes. a) GMIA clearly outperforms the mean based feature. b) The results are affected if not only voiced speech is used. c) The sets of male and female speakers are clearly separated.

effects and the advantage of GMIA over mean features are shown. The performance is optimal for windows between 100–500 ms and drops sharply for shorter lengths. This aligns with the intuition discussed in Section 6.5 that there exists a trade off between the frequency and time resolution of a speaker–dependent feature. The results of unprocessed speech are compared to the ones using only voiced speech. In all cases, GMIA is contrasted to the mean input feature. Figure 6.7(b) shows that voiced speech extraction is important to achieve 'optimal' results. Figure 6.7(c) illustrates the classification performance of GMIA using a window of 280 ms. Note that the sets of male and female speakers are clearly separated.

Table 6.1 line 2, column 2 presents EER results of GMIA against previous approaches and other representative results from the literature. The speaker verification performance can be improved by inclusion of the previously discussed pitch feature. Results of this procedure are illustrated in Table 6.1 line 1, column 2. The identification rates of the algorithms are included for comparison with previous results in the literature. The assumption of differently distorted inputs results in the chosen data partitioning where the utterances are alternatively separated in a training and testing set, i.e., five utterances are used to train a speaker signature and five to

extract a test signature. Note that this partitioning—and therefore the results—are not exactly comparable to the other methods. Many test procedure variations are used in the literature. For example, Reynolds (1995) employs eight utterances for training and each of the remaining two utterances for testing. Furthermore, Reynolds (1995) defines 10 background speakers per person, that achieve an extreme score (high or low) with the person's feature set and are maximally spread, that are excluded from the respective testing set. As this variability prevents an exact comparison, the goal is not to claim best performance but to demonstrate that MIA extracts an invariant in the data that can be successfully used to represent a speaker independent of the spoken text and channel conditions.

6.8 Summary

This chapter discussed the implementation of a GMIA–based approach for text–independent speaker verification. In a first step, a general speech production background was provided. This illustrated the concepts, physiological processes and terminology in this field. Thereafter, methods were described that are commonly used in the speech processing domain, e.g., linear prediction coding and mel–cepstral coefficients. Moreover, the choice of the database for training and testing was motivated. Note the previous intuition that a mutual feature is a speaker signature under varying channel conditions. The NTIMIT database was used to enable both channel variation and a wide use of the database. Subsequently, the preprocessing of the data, e.g., voice activity and voiced speech detection, were described. Thereafter, the extraction processes of a speaker signature using GMIA and a pitch feature were discussed. Furthermore, the measure of similarity or score between a signature in the database and an input instance was defined. A background model was used that estimates the quality of each input, i.e., the highest score between the input with all signatures in the database. The decision on the acceptance was dependent on the difference between this high score and the score with the signature of the claimed identity. This implementation achieved an equal error rate of 4.0% on the full NTIMIT database of 630 speakers.

TABLE 6.1: MIA and GMIA performance comparison using various NTIMIT database segments.

Method	EER [%]	Identification [%]	NTIMIT Database Section
GMIA + pitch	4.8	70	Test section with 168 speakers
GMIA	6.0	67	
GMIA (Claussen et al., 2009)	6.0	52	
MIA (Claussen et al., 2009)	6.9	48	
MIA (Claussen et al., 2008) [a]	6.8	56	
GMM (Wildermoth and Paliwal, 2003)	12.4	N/A	
GMM (Sanderson, 2002)	9.6	N/A	
GMM (Reynolds, 1995)[b]	7.2	69	
GMIA + pitch	4.5	52	Selection of all 438 male speakers
GMIA	5.7	47	
GMIA (Claussen et al., 2009)	6.9	39	
MIA (Claussen et al., 2009)	8.4	35	
Phoneme GMM (Gutman and Bistritz, 2002)	15.7	N/A	
GMIA + pitch	4.0	51	Full database of 630 speakers
GMIA	5.1	44	
GMIA (Claussen et al., 2009)	6.5	37	
MIA (Claussen et al., 2009)	7.5	32	
GMM (Wildermoth and Paliwal, 2003)	8.8	N/A	

[a] All utterances of a person were concatenated, jointly preprocessed and 50-50 partitioned for training and testing.

[b] The background model for each person (i.e., the 10 speakers with similar/dissimilar features) were excluded from the testing respectively.

Chapter 7

Illumination Robust Face Recognition

Person identification, verification, tracking etc. is of increasing interest in our interconnected world. With the advances in technology, these tasks can now be reliably performed using different biometrics. Besides reliability, a goal is to use approaches that are least intrusive for the tested person. That is, identification should be performed unnoticeably while the tested person passes by at a distance. One possible approach that promises these capabilities is face recognition. Additional to its non-intrusiveness, this method can be easily employed due to the low cost of camera surveillance and its widespread routine deployment nowadays. Face recognition can be performed on both 2D and 3D representations of a face. Clearly, the 3D representation captures additional information that results in an improved recognition performance. For example, an algorithm that uses the 3D shape information of a face is less prone to errors from different appearances resulting from illumination or make-up. However, much research effort focuses on face recognition on 2D images. The motivation for this is the previously mentioned easy employment using available infrastructure.

While face recognition seems intuitive for a human its automatic application in a machine is challenging. The reason is the high variability of the input information in an uncontrolled environment. First, a machine needs to detect if a face is present in an image. Here it is possible that either no, one or multiple faces are visible. If multiple faces are present, they have to be processed separately. Second, the scaling of the face, that is dependent on the distance of the person from the camera, needs to be normalized. Third, the system has to account for the direction of the head. For example, the head may be tilted, the face may be recorded frontally, from a higher position or from the side. Moreover, the system has to account for occlusions, for instance, objects covering the face, rain, facial hair, glasses, hats, make–up etc. Additionally, variations in illumination are challenging for modern face recognition application as already discussed in Section 5.2. Also, different facial expressions have an impact on the system. Furthermore, the background of the recording is different for each camera location and the resolution of the face image varies dependent on the camera and the distance of the person. Finally, there are many application–dependent effects that result in additional problems. For example, it may be desirable to find a person based on old pictures that do not account for facial

(a) Original image (b) Distorted image (c) Distorted image upside down

FIGURE 7.1: The Thatcher effect. (a) Original face image. (b) Face image with the eyes and mouth flipped around a horizontal axis. The distortion is clearly visible. (c) Upside down representation of the distorted image. Note that the distortion is less obvious than in (b).

changes resulting from the aging process.

The starting point in designing a system that aims to automate tasks that are successfully performed by humans is to study and model the human approach. This is similarly done for speaker verification discussed in Chapter 6, e.g., the design of MFCC features based on human auditory perception. For example, people were asked to classify modified face images to learn about the features used by humans for face recognition. The hypothesis is that the face recognition performance is not influenced if feature–independent image information is deleted. A short summary of various results is given in Jain et al. (2006, pp. 67,68). For instance, humans use both global and local features. Global features like low spatial frequency–band representations of a face are sufficient for gender classification. However, local features, that are captured by the high frequency spatial bands, are necessary for face recognition. A further result shows that it is harder for humans to recognize faces that are not perceived as 'attractive' nor 'unattractive'. This suggests that there exist specific, observer–dependent features that enable a classification in these subclasses. This pre–classification thereafter supports the recognition. Furthermore, the face recognition performance was found to depend on the familiarity of the observer with similar face images, e.g., from different races or genders. An experiment that underlines the necessary familiarity of the observer to the recognized image class is demonstrated by the 'Thatcher effect' (Thompson, 1980), illustrated in Figure 7.1. Here, the eyes and mouth of a person are rotated by 180 degrees leaving the rest of the face unchanged. While the altered image seems grotesque when viewed normally, this distortion is less obvious if the head is viewed upside down. This result can be interpreted in two ways. First, larger differences to a normal image are accepted by the observer if a lower familiarity to the image class is present. Second, not only the position of local features is important for face recognition but also their appearance, e.g., their shape. This suggests that face recognition is done in a hierarchical manner. Another result shows that young children typically recognize unfamiliar people based on ambiguous clues, e.g., clothing, glasses, a walking stick, hairstyle, hats etc. This clearly shows that for a human, face recognition is not as simple as usually perceived.

Taking this into consideration, it is sensible to break down the face recognition task into different subproblems, e.g., face detection, scaling etc. In this chapter, MIA is used for feature extraction under various illumination conditions. In contrast to Section 5.2, the goal is to tackle a more realistic problem without perfect alignment of faces from the same class. Furthermore, variations in facial expressions and occlusions like glasses are included.

Section 7.1 provides a short background to face recognition to familiarize the reader with related approaches. Section 7.2 discusses the choice of the database and compares properties of different alternatives. Thereafter, Section 7.3 describes the preprocessing and extraction of 'mutual faces' that represent the commonalities of a set of inputs from one person. Subsequently, Section 7.4 discusses the results of MIA for illumination robust face recognition. Finally, the chapter is summarized in Section 7.5.

7.1 Face Recognition Background

Face recognition algorithms can be classified into feature- and appearance–based approaches. Feature–based approaches use for example positions or angles and distances between landmarks, e.g., eye corners, ends of the mouth, eyebrows, chin, tip of the nose, nostrils etc. to distinguish between faces. It is assumed that local features capture information that is necessary for face recognition. This is motivated by results of previously–discussed human face recognition experiments. A further advantage of feature–based approaches is that it is possible to detect and compensate for variations in the camera angle. For example one can deduce that a face is rotated from a difference in distances of the eye corners to the tip of the nose. Using an average 3D model of a head, it is possible to approximate the feature positions for the frontal view. However, a disadvantage of feature based methods is the difficulty of finding local landmarks reliably. The reason for this are, e.g., variations in lighting, facial expressions or occlusions.

The second class of face recognition approaches are appearance–based. These approaches are widely used and successfully applied in modern face recognition systems (Turk and Pentland, 1991; Belhumeur et al., 1997; Moghaddam and Pentland, 1997). The MIA approach for extraction of 'mutual faces', that is discussed in this chapter, falls in this category. Ubiquitous methods of the same category are the eigenface (Turk and Pentland, 1991) and fisherface (Belhumeur et al., 1997) approaches. Both methods view each image as a single vector, i.e., the lines of the image are concatenated forming a long vector.

The eigenface approach first subtracts the mean of all images in the database from each image instance. Thereafter, PCA is used to find the vectors or principal components that capture the maximum variance in the database. PCA is discussed in Section 2.1. Each principal component represents a linear combination of all mean–subtracted images in the database. These face image mixtures are called eigenfaces. An example of the first six eigenfaces of the Yale database (Belhumeur et al., 1997) is given in Figure 7.2. Note that the original images can be recovered using a combination of the eigenfaces. An approximation of the original images is possible

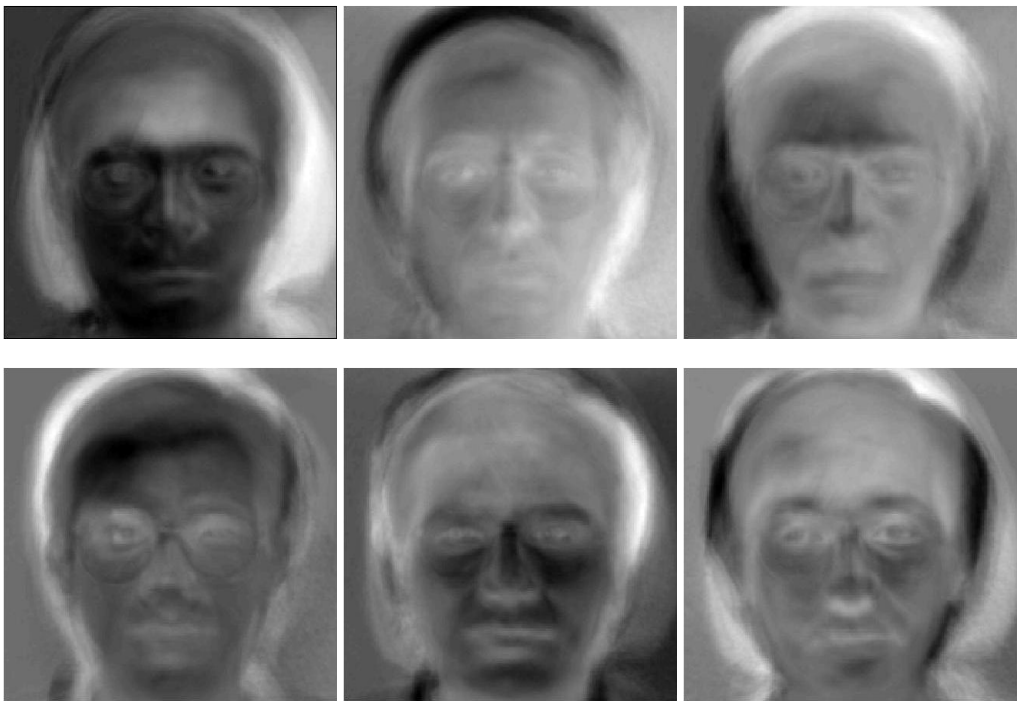


FIGURE 7.2: The first six eigenfaces of the Yale database. The eigenvalues of the eigenfaces reduce line–by–line from the top left to the bottom right.

using a mix of the eigenface subset with the highest eigenvalues. This is discussed in more detail in Section 2.1.1. A simple face recognition algorithm using eigenfaces can be described as follows. First, the face images from one person in a training database are represented using a subset of eigenfaces. The weights of the eigenfaces that were used to represent these images are statistically modeled using a GMM (discussed in Section 2.9). This procedure is repeated for all individuals. For testing, the unknown input face is again represented by a mixture of eigenfaces. The identity is selected that corresponds to the modeled weights that are maximally similar to the ones of the unknown face.

The fisherface approach uses the available class label information in the training data for feature extraction. The goal is to find a set of features that are most discriminant between the classes. For this task, the fisherface approach uses LDA which is discussed in Section 2.4. The most discriminant vectors that result from LDA are called 'fisherfaces' when reshaped to a 2D image. A face recognition algorithm using fisherfaces can be implemented analogous to the previousely discussed eigenface approach by simply replacing the eigenfaces with fisherfaces. This method was shown to be more robust to variations in illumination conditions (Belhumeur et al., 1997).

7.2 Databases

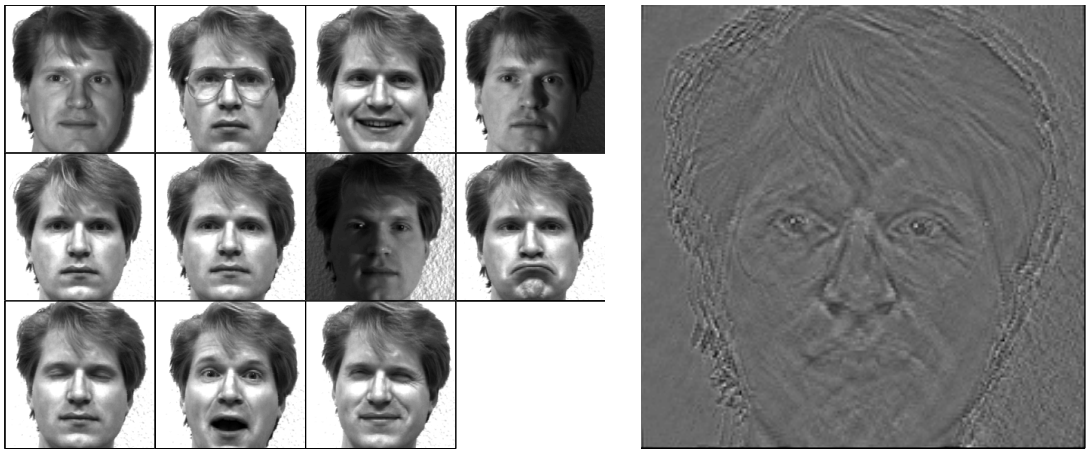
The selection of the database is a critical task for the evaluation of a face recognition algorithm. It is important to select a large and widely–used database as discussed in Section 6.3. This way it is possible to cover a wide variety of possible cases in the field and to enable a comparison to results from other methods. In the following, two databases are discussed that are not used in this chapter. The reason for this is their special prominence in the field of face recognition and research on illumination conditions. One of these large, realistic and well known databases is FERET (Phillips et al., 2000). This database contains 14,126 images of 1,199 individuals and is provided by NIST. The database was separately collected from algorithm developers and includes a variety of conditions, e.g., aging, illumination, facial expressions, scaling, camera angles etc. However, the goal of this chapter is to analyze the effectiveness of MIA for illumination independent face recognition. Hence, it is important to choose a database that is commonly used for the evaluation of this particular effect on algorithms. Therefore FERET was not used. A database that is widely used to analyze the effect of illumination conditions is the YaleB database (Georghiades et al., 2001). The database contains 5850 images of 10 people. The faces were illuminated from 64 different angles with a single light source. Each experiment was repeated for 9 different angles of the head. Additionally, an image with ambient illumination is included for each viewing condition. This database was used in Section 5.2. The images of one viewing condition were taken nearly instantaneously using strobe lights. This results in an exact alignment and equal facial expressions between all images of a viewing condition. Thus, when ignoring other than frontal views, it is simple to recognize the person despite the illumination conditions. Therefore this database was not used.

Another database that is designed to evaluate the effect of varying illumination conditions is the Yale database (Belhumeur et al., 1997). The difference to the YaleB database is that this earlier version includes misalignment, different facial expressions and variations in scaling and camera angles. However, in comparison to the FERET database, the variations in scaling and camera angle are minor. Thus, no additional component is necessary to account for these effects. Although, by allowing these variations the algorithm can be tested in a more realistic face recognition scenario. A disadvantage of the Yale database is its size of only 165 face images of 15 people. Therefore, the database can not be split in a fixed training and testing section. The testing is done in a leave one out fashion as discussed in Section 7.4 to ensure that the training and testing data do not overlap.

The original Yale database includes a large amount of background. This is illustrated on one image instance in Figure 7.3(a). Thus, it is necessary to crop the images to focus on the face information. Furthermore, it is important to align all images using a defined procedure. For simplicity, a half automatic cropping and alignment procedure was used. The original images are 243×320 pixels in size. A 200×200 pixel image is cut around a manually defined center on the nose and between the eyes. The cropped image instance is illustrated in Figure 7.3(b). In the following, it is shown how a mutual face is extracted from the cropped images of each person.

(a) Original image (b) Cropped image

FIGURE 7.3: Yale database preprocessing. (a) Original image from the Yale database. (b) Cropped image with the center point on the nose and between the eyes.



(a) Yale face database class (b) Mutual image information

FIGURE 7.4: (a) Image set of one individual in the Yale database. The set contains 11 images of the person taken with various facial expressions and illuminations, with or without glasses. (b) MIA result, or mutual face estimated from all images of the set.

7.3 Mutual Face Extraction

As discussed in Foley et al. (1997), the reflected light intensity I of each image pixel can be modeled as a sum of an ambient light component and directional light source reflections. Let I_a and I_p be the ambient/directional light source intensities. Also, let k_a, k_d, $\vec{\boldsymbol{n}}$ and $\vec{\boldsymbol{l}}$ be ambient/diffuse reflection coefficients, surface normal of the object, and the direction of the light source respectively. Hence,

$$I = I_a k_a + I_p k_d (\vec{\boldsymbol{n}} \cdot \vec{\boldsymbol{l}}) \ .$$

(a) Eigenface input (b) Fisherface input (c) MIA input

FIGURE 7.5: Examples of training instances used in (a) eigenfaces, (b) fisherfaces and (c) MIA: (a) Mean-subtracted face obtained as difference between a face instance and the mean of all images in the database. (b) Mean-subtracted face obtained as difference between a face instance and the mean image of all instances for the same person. (c) "Centered" face image, obtained by subtraction of the mean column value from each image column.

More complex illumination models, including multiple directional light sources, can be captured by the additive superposition of the ambient and reflective components for each light source (Foley et al., 1997, see Equation 16.20).

The intuition is that MIA can extract an illumination–invariant mutual image, perhaps including $I_a k_a$, from a set of aligned images of the same object (face) under various illumination conditions. In the following, mutual faces were used in a simple appearance–based face recognition experiment. Prominent methods of this widely researched area include the eigenface and fisherface approaches as discussed in Section 7.1. Most use mean image subtraction for preprocessing, which reduces the image space dimensionality compared to the original image set. This effect is illustrated in Figure 4.3. Therefore, this step cancels potentially discriminant image information. In contrast, MIA uses centered images ($\boldsymbol{x}_i^T \cdot \underline{\mathbf{1}} = 0 \quad \forall i$) as inputs. Note that $\boldsymbol{x}_i$ represents an input instance of MIA. Usually, this is not equivalent to a whole image, as for the eigenface and fisherface approach. For the discussed MIA approach, the input instances are defined as single image columns or rows of the 2D Fourier transformed images. The mean subtraction centers each image line separately. Figure 7.5 illustrates the difference between a mean–face–subtracted input instance in the eigenface/fisherface approach and the centered MIA input.

The procedure to extract the mutual face from the face set of one person can be defined as follows: First, images are 2D Fourier transformed. Second, each row of the images is centered and windowed. The windowing represents a high–pass filter operation. Thus, the approach focuses on the high frequency image information. In the previously discussed results of experiments on human perception, these frequencies were found to be of critical importance for face recognition. Thereafter, MIA is performed separately on each set of rows. After normalizing and reassembling the rows, the procedure is repeated with the columns of the original images. Thus, two mutual faces are generated, added, and the result is transformed

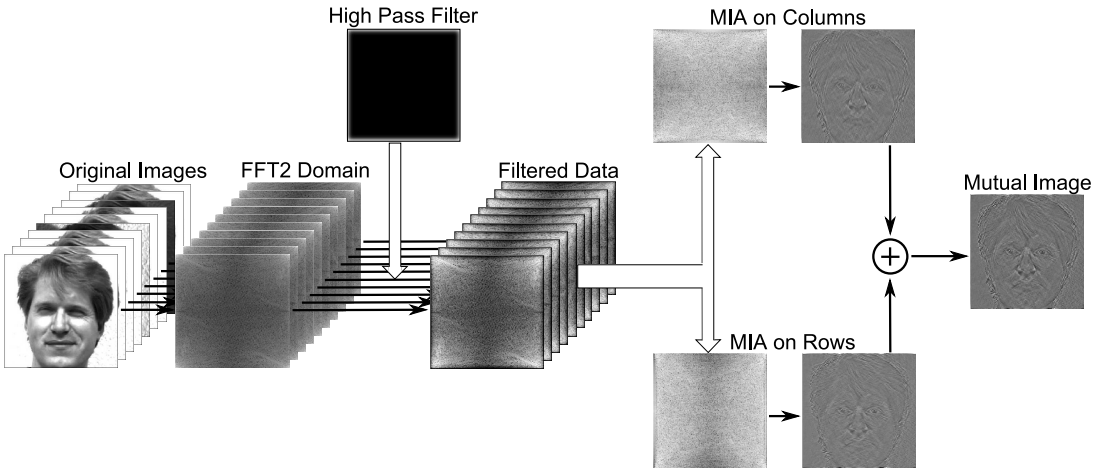


FIGURE 7.6: Extraction process of the mutual image representation.

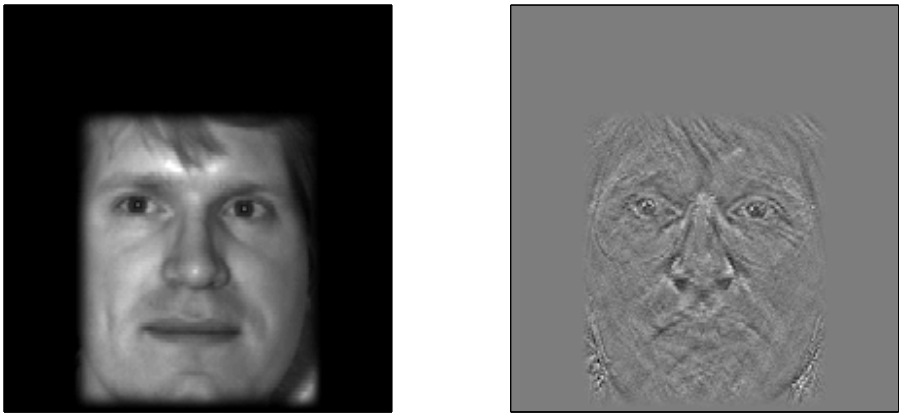
through the 2D inverse FFT. The procedure is shown in Figure 7.6. Note that, for illustrative purposes, the 2D inverse FFT transformed images of the MIA results on columns and rows are illustrated before they are combined to a single 'mutual image'. The procedure to generate a mutual image representation is repeated for each individual in the database. Thus, a codebook of mutual faces is generated. In the following, the procedure is discussed that is used to evaluate the similarity between a new image and the mutual image representations in this database. The identity of the new face image is selected dependent on this similarity.

7.4 Evaluation of Results

In this section, the method is discussed that is used to estimate the similarity between the mutual face representations and a new input instance. Furthermore, the testing procedure is described and the results are presented and compared to other approaches on the Yale database.

The Yale database used with 165 images is small. This results in problems with the testing as discussed in Section 7.2. To make optimal use of the database, mutual faces are learned on all but a single test image using the "leave–one–out" method discussed in Duda and Hart (1973, p. 75). The "leave–one–out" method can be summarized as follows. All images but one are used for training. The remaining image is used for testing against all trained 'mutual faces'. Thereafter, the 'mutual faces' are trained again with a different left–out image for testing. This procedure is repeated until each image has been left out once for testing. In this way, a large training database is available while it is guaranteed that the testing image is not represented in the training set.

Both 'mutual faces' and the test image are cropped to focus on information that is captured in the face features rather than hairstyle, misalignment or background. The filter used for cropping is empirically found and applied for all tests. The cropped face representations of both testing and the correspondent mutual face are illustrated in Figure 7.7.



(a) Cropped test image (b) Cropped mutual face

FIGURE 7.7: Cropped face representations for test of similarity. One cropping filter was used for all image instances. The hypothesis is that the remaining face image contains less information about hairstyle, background or misalignment of the face shape. (a) Test image segment that is used to compute the similarity to the mutual faces (b) Cropped 'mutual face' from the same person as the test image. The test image was not used for training.

As for the generation of the mutual image, the similarity of two images is computed separately on the corresponding centered rows and columns. The measure of similarity between a test image and the MIA representation of a person is the mean cosine distance. The resulting scores $\mathcal{S}_1$ and $\mathcal{S}_2$, from the mean cosine distances of the rows and columns respectively, are fused using $\mathcal{S} = \sqrt{{\mathcal{S}_1}^2 + {\mathcal{S}_2}^2}$.

In case the left–out image is one of the three illumination variant cases of the Yale database (centered light, left light and right light), this approach leads to an identification error rate (IER) of 2.2%. Overall, in exhaustive leave–one–out tests, the mutual face method results in an error rate of 7.4%. This shows that the approach using mutual features is robust against different illumination conditions but suffers from variations in facial expressions and occlusions. Recognition performance for unknown illumination is comparable or beyond various reported results obtained with similar data (Table 7.1). Note that a mutual face can be used to extract landmarks like eye corners, ends of the mouth, eyebrows, chin, tip of the nose, nostrils, etc. The advantage over extraction of these points from the original training image is that the mutual face represents the invariant representation excluding, e.g., variations in illumination. It is expected that this approach can lead to a more robust detection of these landmarks. Thus, the MIA approach can be used to enhance both feature- and appearance–based methods, only requires minimal training, and appears insensitive to multiple illumination sources and diffuse light conditions.

TABLE 7.1: Comparison of the identification error rate (IER) of MIA with other methods using the Yale database. Full faces include some background compared to cropped images.

Method	IER [%]	Evaluation	Comments
MIA (Claussen et al., 2008)	7.4	leave–one–out	Cropped face (see Figure 7.7)
Minimax Probability Machine (Hoi and Lyu, 2004)	21.2	k–fold cross validation	Cropped face test
Kernel PCA (Yang et al., 2000)	26.0	leave–one–out	Cropped face test
Fisherface (Belhumeur et al., 1997)[a]	7.3		
Eigenface (Belhumeur et al., 1997)[b]	24.4		
Eigenface w/o 1-3 (Belhumeur et al., 1997)[bc]	15.3		
MIA (Claussen et al., 2008)	2.2	leave–one–out	Only illumination
Minimax Probability Machine (Hoi and Lyu, 2004)	10.1	k–fold cross validation	Without illumination
Fisherface (Belhumeur et al., 1997)[a]	0.6	leave–one–out	Full face test
Eigenface (Belhumeur et al., 1997)[b]	19.4		
Eigenface w/o 1-3 (Belhumeur et al., 1997)[bc]	10.8		

7.5 Summary

This chapter discussed an implementation of a MIA–based method for illumination–robust face recognition. In a first step, a general background to face recognition was provided. This illustrated some domain specific challenges and exemplified how human perception is used to define discriminant features. Next, prominent methods, e.g., the eigenface and fisherface approach, were discussed. The Yale database was chosen to include different illumination conditions and minor misalignments, occlusions etc. Thereafter, a semi–automatic preprocessing approach was described. The procedure that is used to extract a mutual face was discussed. Moreover, a similarity metric or score between an input instance and the mutual faces from the database was defined. Mutual faces for illumination–robust face recognition achieve an identification error rate of 7.4% in exhaustive leave–one–out tests on the Yale database.

[a] Classification was performed using 15 FLDA directions.
[b] Classification was performed using 30 principal components.
[c] The first three principal components that represent eigenvectors with maximal eigenvalues were disregarded.

Chapter 8

Conclusions

The goal of this thesis was to research methods that find a unique invariant or signature of a dataset which can be used for pattern recognition problems. If it is possible to extract such an essence of an input set it is the intuition that this signature will adequately represent new inputs of the class despite unseen variability. A novel method, called mutual interdependence analysis (MIA), was designed in Chapter 4 to extract such mutual features from a set of inputs. By definition, the MIA invariant is a linear combination of class examples that has equal correlation with all the training samples in the class. An equivalent view is to find a direction to project the dataset such that projection lengths are maximally correlated. Both goals have MIA as their unique solution. It is shown that the scatter of the inputs onto this unique direction is zero for linearly independent inputs. Using experiments with synthetic data, it was verified in Chapter 5 that MIA results in the desired unique, invariant class representation.

There exist many signal processing methods to extract features for pattern recognition problems. Some of these were reviewed in Chapter 2. MIA was compared to related work to evaluate its novelty and contribution. For instance, MIA and MCA share the goal to extract an invariant in the data. In contrast to MCA, MIA assumes the result in the span of linearly independent inputs. In this case, the common step of mean feature subtraction reduces the span of the data for MCA. The invariant in the data is not represented by the mean subtracted space. Thus, only MIA can extract this invariant. Moreover, this thesis discusses similarities of MIA with Bayesian estimation and a modified CCA approach. Each brings insights into the value and properties of MIA. As a result, a generalized MIA solution (GMIA) is found. It is shown in Section 5.1 that GMIA finds a signal component that is not captured by ubiquitous signal processing methods. Moreover, simulations are presented that demonstrate when and how MIA, GMIA or the mean represent a common signature in the inputs. In a nutshell, MIA and GMIA work well if a signal is equally present in the inputs, even if it accounts only for a small part of the signal energy. Real world problems usually do not exactly fit a synthetic signal model. Therefore, it is important to analyze the behavior of MIA in situations where its model does not exactly fit. For these situations it was shown that GMIA continues to extract meaningful information. GMIA is preferable over MIA if the assumption of an equally present component in the inputs does not

hold. Clearly, MIA and GMIA extend the current signal processing tools capturing previously hidden information.

After the theoretical analysis of MIA and verification of the approach on synthetic data it was shown that mutual signatures are of interest in real world applications. In this thesis, the performance of MIA and GMIA was demonstrated on both text–independent speaker verification and illumination–independent face recognition applications. For example, GMIA achieved an EER of 4.0% in the text–independent speaker verification application on the full NTIMIT database of 630 speakers. This result supports the assumption that MIA extracts a common representation from speech recordings over different nonlinearly distorted channels. Furthermore, data segmentation effects were evaluated indicating a trade–off between the time and frequency resolution of the mutual signatures. For illumination–independent face recognition, the mutual face method achieved an error rate of 7.4% in exhaustive leave–one–out tests on the Yale face database. It was shown that MIA is more robust to differences in illumination conditions than to occlusions and differences in facial expressions. Overall, MIA and GMIA are found to achieve competitive pattern recognition performance to other modern algorithms.

Chapter 9

Future Work

The utilization of mutual features which are invariants in the training data showed promising results. While this thesis explores many aspects of this topic, there remain various areas with potential for further research. This further work can be classified into algorithmic and application based explorations. On the one hand, the algorithmic work could seek to further the understanding of methods for the extraction of mutual features. On the other hand, the application related work could aim to improve results and application dependent know–how in problems characterized by e.g., high dimensional data, low number of input samples etc. We discuss these directions in more detail below.

The resolution and amount of sensor data is rapidly increasing with advances in technology. Thus, modern signal processing applications deal with increasing amounts of data. Future work could investigate statistical properties of MIA for a large number of inputs and computational tractability in large dimensions. Signal processing algorithms like PCA, CCA, FLDA etc., use the dependence between inputs in the form of covariance or scatter matrices. They suffer from growth of these matrices, e.g., the computational complexity of PCA is $O(N^2D)$ (Bach and Jordan, 2002). It is often possible to use dual formulations of the algorithms to select if the matrices grow with the number of dimensions or samples. This is exemplified in equations 4.19 and 4.20. A problem arises if both are too high to store all information in the memory. One possible solution is to approximate the GMIA result iteratively. This could be achieved by providing an estimate of $\boldsymbol{\mu}_{\boldsymbol{w}}$ from the GMIA results of previous input subsets. The procedure will converge by increasing the weight of this component. Future work could analyze the properties of such a procedure, i.e., if it is sensible to estimate a mutual signature iteratively in the previously sketched manner.

Mutual interdependence analysis is a linear method for extraction of invariants in high–dimensional spaces. A prominent approach that uses linear classification in high–dimensional spaces is the kernel support vector machine (KSVM). This method is commonly used for nonlinear classification in a lower dimensional space. The projection of the lower dimensional inputs to a possibly infinite dimensional space is performed virtually. KSVMs take advantage

of the fact that nonlinear problems can be transformed to linear ones by transformation of the inputs to a specific, higher dimensional space. Figuratively, KSVMs use a 'key function', or kernel, to unlock a particular nonlinear problem. For example, all quadratic problems in the two–dimensional space are linear in the three-dimensional space: $x_{new} = x_{old}^2, y_{new} = y_{old}^2$ and $z_{new} = x_{old} \cdot y_{old}$. Another property of KSVMs is that only a sparse subset of the training data is used for classification. This results in an improved classification speed over non sparse kernel methods. A sparse kernel method that is related to MIA is the relevance vector machine (RVM) (Tipping, 2001). In contrast to the outputs of KSVMs, that represent decisions, the outputs of RVMs model posterior probabilities. While RVMs achieve comparable performance to KSVMs they typically lead to sparser models (Bishop, 2006, p. 345). Further research could analyze the similarities between MIA, KSVMs and RVMs. The goal is to find new applications and to transfer knowledge from these domains. For example, the MIA performance could be improved using a sparse solution to its problem.

In the following, future work on the application side is proposed. Generally, there exists a trade–off between the frequency resolution and the time resolution as demonstrated by the Gabor uncertainty principle (Gabor, 1950). However, additionally to this fundamental principle, there exists a conceptual trade–off between the time and frequency resolution in text–independent speaker verification applications as discussed in Chapter 6. That is, while the just noticeable frequency resolution of the human ear is approximately 2 Hz (Deng and O'Shaughnessy, 2003, p. 241), suggesting a window size of 0.5 s, the 'phoneme' length is in the order of 10 ms. Common approaches model the phoneme structure of each person by multiple states. On the other hand, the GMIA approach finds a single speaker representation with high frequency resolution. Both methods yield promising results. Therefore, future work is planned to combine both philosophies to obtain a more powerful classifier. Furthermore, issues such as the robustness of both approaches to different noise conditions, recording lengths for training and testing, computational requirements and the minimum number of background speakers could be analyzed. Moreover, it is possible to increase the confidence in the results by implementing also algorithms that model the 'phoneme' structure, e.g., with the same preprocessing.

Speaker verification approaches can usually not assume that all possible input speakers are enrolled in the training phase. That is, there is the possibility that an unknown person tries to gain access under a false identity. Speaker verification tests that assume such conditions are called open–set tests. For simplicity, such tests were excluded in Chapter 6. However, this test scenario is of interest in real world applications. Thus, further work could evaluate GMIA based speaker verification in an open–set test scenario. Additionally, it should be analyzed if these different assumptions have an effect on the currently chosen background model.

In this thesis, the background speaker model includes all speaker signatures in the database. A score of an input and a claimed identity is evaluated by comparison to the highest score of the input with all signatures. However, this exhaustive computation of scores for all enrolled speakers is undesirable for large scale applications. Hence, future work could include the design of a search procedure for a representative subset of speaker signatures. The goal is to find a

small number of signatures that are maximally different from each other such that at least one of them achieves a high score with potential speech inputs. New background speaker models could be explored if this approach is not feasible.

There are multiple possibilities to advance the current analysis and procedure for illumination–independent face recognition. For example, the disadvantage of the used Yale database in Chapter 7 is its small size of 165 images. This is a drawback that limits the statistical interpretability and representativeness of the results. Future work could utilize the FERET database for large scale tests. Multiple preprocessing tools, e.g., for face detection, rescaling, alignment etc., have to be designed prior to this evaluation. A three–dimensional head model could be used to recover hidden information resulting from different orientations of the head.

A mutual face was shown to be an invariant representation that excludes illumination conditions. One disadvantage of features like positions or angles and distances between landmarks, e.g., eye corners, ends of the mouth, eyebrows, chin, tip of the nose, nostrils etc. is the lack of robustness of their automatic detection. The reason for this is the variability in the face images, e.g., by differences in illumination. Future work could evaluate if mutual faces can be used as a basis to improve the robustness of detection at this scale. If this is successful, a MIA supported, landmark–based face or object recognition approach should be implemented and evaluated. Further research could also include the extraction of mutual representations of local features like mouth, eyes, nose, etc. The goal is to implement a hierarchical approach for face recognition that simulates the human face recognition behavior.

MIA was shown to perform worse than GMIA in situations where its model does not hold. Also, GMIA outperformed MIA in the text–independent speaker verification application if the parameter λ was small but prevented negative combinations of the logarithmic features. It is expected that this performance gain can also be observed for the face recognition application. However, it is currently unclear how the parameter λ should be selected for optimal performance. In the future, the effects of λ on the illumination–independent face recognition application could be analyzed. The goal is to design a method that estimates the optimal parameter given the current set of inputs. Ideally, λ should be chosen automatically based on information theoretic principles, cross–validation, etc. Finally, alternatives ways of setting $\boldsymbol{C_f}$ and $\boldsymbol{C_w}$ could be explored to model better the problem.

This chapter indicated some starting points for future explorations. It is expected that most of these paths of future work will unveil additional questions and increase the understanding of MIA. One goal is to inspire you, the reader, to help further the knowledge in this field.

Bibliography

R. Adcock. Note on the method of least squares. *Analyst*, 4:182–184, 1877.

F. R. Bach and M. I. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2002.

F. R. Bach and M. I. Jordan. A probabilistic interpretation of canonical correlation analysis. Technical Report 688, Department of Statistics, University of California, Berkeley, 2005.

M. S. Bartlett. Further aspects of the theory of multiple regression. *Proceedings of the Cambridge Philosophical Society*, 34:33–40, 1938.

L. E. Baum. An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. In *Inequalities III: Proceedings of the Third Symposium on Inequalities*, pages 1–8, Los Angeles, CA, 1972.

L. E. Baum and T. Petrie. Statistical Inference for Probabilistic Functions of Finite State Markov Chains. *The Annals of Mathematical Statistics*, 37(6):1554–1563, 1966.

T. Bayes. An essay towards solving a problem in the doctrine of chances. By the late Rev. Mr. Bayes, F.R.S. communicated by Mr. Price, in a letter to John Canton, A.M.F.R.S. *Philosophical Transactions*, 53:370–418, 1763.

P. N. Belhumeur, J. Hespanha, and D. J. Kriegman. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.

A. J. Bell and T. J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159, 1995.

R. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961.

C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, New York, NY, 2006.

J. F. Cardoso. Infomax and maximum likelihood for blind source separation. *IEEE Signal Processing Letters*, 4(4):112–114, 1997.

E. C. Cherry. Some experiments on the recognition of speech, with one and with two ears. *Journal of the Acoustical Society of America*, 25(5):976–979, 1953.

H. Claussen, J. Rosca, and R. Damper. Mutual interdependence analysis. In *Independent Component Analysis and Blind Signal Separation*, pages 446–453, Heidelberg, Germany, 2007. Springer-Verlag.

H. Claussen, J. Rosca, and R. Damper. Mutual features for robust identification and verification. In *International Conference on Acoustics, Speech and Signal Processing*, pages 1849–1852, Las Vegas, NV, 2008.

H. Claussen, J. Rosca, and R. Damper. Generalized mutual interdependence analysis. In *International Conference on Acoustics, Speech and Signal Processing*, volume in press, Taipei, Taiwan, 2009.

C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

S.B. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4):357–366, 1980.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum-likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

L. Deng and D. O'Shaughnessy. *Speech Processing: A Dynamic and Optimization-Oriented Approach*. Signal Processing and Communications. Marcel Dekker, Inc., 2003.

L. Devroye. *Non-Uniform Random Variate Generation*. Springer-Verlag, New York, NY, 1986.

D. Donoho. High-dimensional data analysis: The curses and blessings of dimensionality. Lecture delivered at the conference "Math Challenges of the 21st Century" held by the American Mathematical Society organised in Los Angeles, August 6-11, August 2000.

R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York, 1973.

E. Erzin, A.E. Cetin, and Y. Yardimci. Subband analysis for robust speech recognition in the presence of car noise. In *International Conference on Acoustics, Speech and Signal Processing*, volume 1, pages 417–420, Detroit, MI, 1995.

R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7: 179–188, 1936.

W. M. Fisher, G. R. Doddington, K. M. Goudie-Marshall, C. Jankowski, A. Kalyanswamy, S. Basson, and J. Spitz. NTIMIT. CDROM, 1993.

J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics (2nd ed. in C): Principles and Practice*. Addison-Wesley Longman Publishing, Boston, MA, 1997.

D. Gabor. Communication theory and physics. *Philosophical Magazine*, 4:1161–1187, 1950.

J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, N. L. Dahlgren, and V. Zue. TIMIT acoustic-phonetic continuous speech corpus. CDROM, 1993.

A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):643–660, 2001.

G. Golub and C. F. Loan. An analysis of the total least squares problem. *SIAM Journal on Numerical Analysis*, 17:883–893, 1980.

A. Gretton, R. Herbrich, A. Smola, O. Bousquet, and B. Schölkopf. Kernel methods for measuring independence. *Journal of Machine Learning Research*, 6:2075–2129, 2005.

D. Gutman and Y. Bistritz. Speaker verification using phoneme-adapted Gaussian mixture models. In *European Signal Processing Conference*, volume 3, pages 85–88, Toulouse, France, 2002.

H. Hartley. Maximum likelihood estimation from incomplete data. *Biometrics*, 14:174–194, 1958.

T. Hastie, R. Tibshirani, and A. Buja. Flexible discriminant analysis by optimal scoring. *Journal of the American Statistical Association*, 89(428):1255–1270, 1994.

T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning*. Springer-Verlag, 2001.

A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

C. Hoi and M. R. Lyu. Robust face recognition using minimax probability machine. In *International Conference on Multimedia and Expo*, pages 1175–1178, Taipei, Taiwan, 2004.

J. Holmes and W. Holmes. *Speech Synthesis and Recognition*. Taylor & Francis, Inc., London, UK, 2nd edition, 2001.

R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1999.

H. Hotelling. Analysis of a complex of statistical variables into principle components. *Journal of Educational Psychology*, 24:417–441, 498–520, 1933.

H. Hotelling. Relation between two sets of variates. *Biometrika*, 28:322–377, 1936.

A. Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, 1999.

A. Hyvärinen and Erkki Oja. A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9(7):1483–1492, 1997.

A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. John Wiley and Sons, 2001.

A. Jain, R. Bolle, and S. Pankanti. *Biometrics: Personal Identification in Networked Society*. Springer, New York, NY, 2006.

C. Jutten and J. Herault. Blind separation of sources, part 1: an adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24(1):1–10, 1991.

I. Kant. *Critique of Pure Reason*. Palgrave Macmillan, Basingstoke, UK, 2003. Originally published 1781.

K. Karhunen. Über lineare methoden in der wahrscheinlichkeitsrechnung. *Annales Academiae Scientiarum Fennicae*, Ser. A.1, Math.-Phys. 37:1–79, 1947. Translated by I. Selin, On linear methods in probability theory, Doc. T-131, Rand Corp. Santa Monica, CA, 1960.

S. M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice-Hall, Inc., Upper Saddle River, NJ, 1993.

P. S. Laplace. Mémoire sur les approximations des formules qui sont fonctions de très grands nombres et sur leur application aux probabilités. In *Mémoires de l'Académie Royale des Sciences de Paris Année 1809*, pages 353–414, 559–565. 1810. Reprinted in Oeuvres Complètes 12, 301–351.

K. Li, M.N.S. Swamy, and M.O. Ahmad. An improved voice activity detection using higher order statistics. *IEEE Transactions on Speech and Audio Processing*, 13:965–974, 2005.

Y. Linde, A. Buzo, and R. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, 28:84–95, 1980.

K. V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate Analysis*. Academic Press, Padstow, Cornwall, UK, 1979.

J. D. Markel and A. H. Gray. *Linear Prediction of Speech*. Springer, Berlin, Germany, 2nd edition, 1980.

I. Markovsky and S. V. Huffel. Overview of total least–squares methods. *Signal Processing*, 87: 2283–2302, 2007.

G. J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. John Wiley, New York, New York, 1997.

B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):696–710, 1997.

T. K. Moon and W. C. Stirling. *Mathematical Methods and Algorithms for Signal Processing*. Prentice-Hall, Upper Saddle River, NJ, 2000.

E. Oja. Principal components, minor components, and linear neural networks. *Neural Networks*, 5:927–935, 1992.

D. O'Shaughnessy. *Speech Communication: Human and Machine*. Addison–Wesley, Reading, MA, 1987.

K. Pearson. On lines and planes of closest fit to points in space. *Philosophical Magazine*, 2: 559–572, 1901.

J.P. Phillips, H. Moon, S.A. Rizvi, and P.J. Rauss. The FERET evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22 (10):1090–1104, 2000.

T.F. Quatieri. *Discrete-Time Speech Signal Processing: principles and practice*. Prentice Hall Press, Upper Saddle River, NJ, 2001.

D. A. Reynolds. Speaker identification and verification using Gaussian mixture speaker models. *Speech Communication*, 17(1-2):91–108, 1995.

S. Saito and F. Itakura. The theoretical consideration of statistically optimal methods for speech spectral density. Technical report, NTT, Tokyo, Japan, 1968.

C. Sanderson. Speech processing & text-independent automatic person verification. Technical Report 08, IDIAP, Martigny, Switzerland, 2002.

A. Schmidt-Nielsen and T. H. Crystal. Speaker Verification by Human Listeners: Experiments Comparing Human and Machine Performance using the NIST 1998 Speaker Evaluation Data. *Digital Signal Processing*, 10:249–266, 2000.

B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, NY, 2004.

S.S. Stevens, J. Volkman, and E. Newman. A scale for the measurement of the psychological magnitude of pitch. *Journal of the Acoustical Society of America*, 8(3):185–190, 1937.

P. Thompson. Margaret Thatcher: a new illusion. *Perception*, 9(4):483–484, 1980.

P. A. Thompson. An adaptive spectral analysis technique for unbiased frequency estimation in the presence of white noise. In *Proceedings of the 13th Asilomar Conference on Circuits, Systems and Computer*, pages 529–533, 1979.

M. E. Tipping. Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244, 2001.

K. Todros and J. Tabrikian. Application of Gaussian mixture models for blind separation of independent sources. In *Independent Component Analysis and Blind Signal Separation*, pages 382–389, Granada, Spain, 2004.

M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1): 71–86, 1991.

M. Welling, F. Agakov, and C. K. I. Williams. Extreme components analysis. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems*, volume 16. MIT Press, Cambridge, MA, 2004.

B. Wildermoth and K.K. Paliwal. GMM based speaker recognition on readily available databases. In *Microelectronic Engineering Research Conference*, Brisbane, Australia, 2003. pagination unknown.

C. Williams and F. Agakov. Products of Gaussians and probabilistic minor components analysis. *Neural Computation*, 14(5):1169–1182, 2002.

L. Xu, E. Oja, and C. Y. Suen. Modified Hebbian learning for curve and surface fitting. *Neural Networks*, 5:441–457, 1992.

M. Yang, N. Ahuja, and D. J. Kriegman. Face recognition using kernel eigenfaces. In *International Conference on Image Processing*, volume 1, pages 37–40, Vancouver, Canada, 2000.

S.K. Zhou, G. Aggarwal, R. Chellappa, and D.W. Jacobs. Appearance characterization of linear Lambertian objects, generalized photometric stereo, and illumination-invariant face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(2):230–245, 2007.