Scalable Probabilistic Models for Face and Speaker Recognition

THIS IS A TEMPORARY TITLE PAGE
It will be replaced for the final print by a version
provided by the service academique.

Thèse n. 6175 2014
présentée le 24 Mars 2014
à la Faculté des Sciences et Techniques de l'Ingénieur
laboratoire de l'IDIAP
programme doctoral en Génie Électrique

École Polytechnique Fédérale de Lausanne

pour l'obtention du grade de Docteur ès Sciences
par

Laurent El Shafey

acceptée sur proposition du jury :

Prof Jean-Philippe Thiran, président du jury
Prof Hervé Bourlard, directeur de thèse
Dr Sébastien Marcel, codirecteur de thèse
Prof Josef Kittler, rapporteur
Dr Jan Černocký, rapporteur
Prof Pascal Fua, rapporteur

Lausanne, EPFL, 2014

Only a life lived for others
is a life worthwhile.
— Albert Einstein

To “my others”, my beloved family and friends. . .

Acknowledgements

In memory of Simone Klein for her unconditional love and support.

The completion of this Ph.D. dissertation has been a long journey.

First, I would like to thank my thesis advisor, Dr. Sébastien Marcel, for giving me the opportunity to carry out my doctoral studies at Idiap in a great research environment and for his guidance during four years. I am also grateful to my thesis director Prof. Hervé Bourlard, as well as the other members of my jury, Prof. Pascal Fua, Prof. Josef Kittler, Prof. Jan Honza Černocký and Prof. Jean-Philippe Thiran, for doing me the honor to supervise my oral exam.

I would also like to thank Prof. David van Leeuwen and Dr. Henk van den Heuvel for managing the BBfor2 project, as well as Dr. Didier Meuwly from the Netherlands Forensics Institute and Prof. Josef Bigun from Halmstad University for hosting me and supervising me during my Ph.D. secondments.

Working at Idiap was nice and easy, thanks to the great administrative support of Nadine, Sylvie, Antoine, Christophe, Ed, Chantal, Corinne, and the technical support of Bastien, Cédric, Frank, Louis-Marie, Norbert and Vincent.

I would also like to thank my office mates and colleagues who made my stay successful and fun : Abhishek, Abhijit, Adolfo, Afsaneh, Aleksandra, Alexandre, Alexandros, André, Andrei, Anindya, Arjan, Ashtosh, Barbara, Billy, Blaise, Braida, Carl, Charles, Chidansh, Chris, Christine, Cosmin, Daira, Danil, Darshan, David, Deepu, Dimitri, Elie, Fabio, Fen, Flavio, Florent, François, Gokul, Gulcan, Gwénolé, Hainan, Hari, Harsha, Hugo, Hui, Ilja, Ivana, Jagan, James, Javier, Jean-Marc, Jesus, Joan, Joel, Jukka, Kate, Kenneth, Lakshmi, Laurent, Leo, Lucia, Majid, Manuel, Marc, Marco, Maryam, Matthias, Matthieu, Mehdi, Minh-Tri, Mohammad, Moussa, Murali, Najeh, Nesli, Nicolae, Niklas, Nikolaos, Olivier, Oya, Paco, Paul, Pedro, Petr, Phil, Philip, Philippe, Pierre-Edouard, Pranay, Radu, Rakesh, Ramya, Raphaël, Riwal, Roger, Romain, Ronan, Roy, Rui, Rémi, Salim, Samira, Samuel, Serena, Sriram, Stefan, Tatiana, Teodora, Thiyagarajan, Thomas, Tiago, Valérie, Vasil, Venkatesh, Xingyu, Yann, Yang, Youssef, Zoltan.

Acknowledgements

Being involved in the BBfor2 project was a great experience that I shared with all the fellows : Abhishek, Ahilan, Erica, Mira, Mitch, Mohamed, Natalie, Ram, Rahim, Rudolf, Ruifang, Tauseef, Vasileios and Vikram.

Finally, I would like to thank my parents, my brother, my sister, the rest of my family as well as my friends for their constant support.

My work was supported by the European Community's Seventh Framework Programme (FP7) under grant agreements 238803 (BBfor2) and 284989 (BEAT). I gratefully thank the European Union for its financial support, and all project partners for a fruitful collaboration. More information about BBfor2 and BEAT is available from the project websites www.bbfor2.net and www.beat-eu.org.

Martigny, March 24th 2014 L. E.S

Abstract

In the biometrics community, face and speaker recognition are mature fields in which several systems have been proposed over the past twenty years. While existing systems perform well under controlled recording conditions, mismatch caused by the use of different sensors or a lack of cooperation from the subject still significantly affects performance, especially in challenging scenarios such as in forensics. Furthermore, existing methods suffer from scalability issues, which prevents them from taking advantage of increasingly large amounts of training data. This is otherwise a promising approach to improve accuracy in such challenging scenarios.

In this thesis we address these problems of mismatch and complexity by developing scalable probabilistic models that we apply to face, speaker and bimodal recognition. Our contributions are four-fold. First, we propose a unified framework for session variability modeling techniques based on Gaussian mixture models (GMM), that encompasses inter-session variability (ISV) modeling, joint factor analysis (JFA) and total variability (TV) modeling. Second, we propose a novel exact and scalable formulation of probabilistic linear discriminant analysis (PLDA), which is a probabilistic and generative framework that models between-class and within-class variations. This formulation solves a major scalability issue, by improving both the time complexity of the training procedure from cubic to linear with respect to the number of samples per class, and the complexity of the scoring procedure. Furthermore, the implementations of all the proposed techniques are integrated into a novel collaborative open source software library called Bob [1] that enforces fair evaluations and encourages reproducible research. Fourth and finally, large-scale experiments are conducted with all of the above machine learning algorithms on several databases such as FRGC for face recognition, NIST SRE12 for speaker recognition and MOBIO for bimodal recognition, showing competitive performances.

Keywords : face recognition, speaker recognition, bimodal recognition, inter-session variability modeling, joint factor analysis, total variability modeling, probabilistic linear discriminant analysis

1. http://www.idiap.ch/software/bob

Résumé

La reconnaissance automatique du visage et du locuteur constituent deux applications technologiquement mûres dans le domaine de la biométrie. De nombreux systèmes ont été proposés au cours des vingt dernières années, lesquels ont une précision relativement élevée lorsque les conditions d'enregistrement des échantillons biométriques sont strictement contrôlées. En revanche, ces systèmes deviennent peu performants en cas de disparités causées par l'utilisation de différents types de capteurs ou par le manque de coopération de la personne à reconnaître, comme c'est le cas avec des applications ambitieuses telles que l'investigation criminelle. Par ailleurs, de nombreuses méthodes pâtissent d'un manque d'extensibilité, leur empêchant de tirer profit de base de données d'apprentissage sans cesse plus grandes, qui pourraient permettre d'améliorer leur robustesse.

Dans cette thèse, nous abordons ces problèmes de disparité et d'extensibilité en développant des modèles probabilistes extensibles que nous appliquons à la reconnaissance automatique du visage, du locuteur et bimodale. Nos contributions comportent quatre volets essentiels. Premièrement, nous proposons un cadre unifié pour un ensemble de trois techniques reposant sur le modèle de mélange gaussien (GMM) : modélisation de la variabilité intersession (ISV), analyse factorielle jointe (JFA) et modélisation de la variabilité totale (TV). Deuxièmement, nous proposons une formulation exacte et extensible de l'analyse discriminante linéaire probabiliste, une approche probabiliste et générative qui modélise les variations inter et intra classes. Cette formulation novatrice résout un problème majeur d'extensibilité, en améliorant à la fois la complexité en temps du processus d'apprentissage de cubique à linéaire par rapport au nombre d'échantillons par classe, ainsi que la complexité du processus d'évaluation. Troisièmement, les implémentations de toutes les techniques proposées ont été intégrées dans une bibliothèque logicielle au code source ouvert dénommée Bob, [2] qui permet des évaluations équitables et encourage la reproductibilité des expériences scientifiques. Enfin, des expériences à grande échelle sont effectuées avec plusieurs bases de données telles que FRGC pour la reconnaissance du visage, NIST SRE12 pour la reconnaisance du locuteur, et MOBIO pour la reconnaissance bimodale, et ce en utilisant les algorithmes d'apprentissage automatique mentionnés ci-dessus.

Mots-clés : reconnaissance du visage, reconnaissance du locuteur, reconnaissance bimodale, modélisation de la variabilité intersession, analyse factorielle jointe, modélisation de la variabilité totale, analyse discriminante linéaire probabiliste

2. http://www.idiap.ch/software/bob

Contents

List of Figures

List of Algorithms

List of Tables

Glossary

Technical acronyms in this thesis are listed below in alphabetical order.

CAR	correct acceptance rate
CCTV	closed-circuit television
DCT	discrete cosine transform
DET	detection error tradeoff
EER	equal error rate
EM	expectation-maximization
FA	false acceptance
FAR	false acceptance rate
FFT	fast Fourier transform
FR	false rejection
FRR	false rejection rate
GMM	Gaussian mixture model
HOG	histogram of oriented gradients
HTER	half total error rate
ISV	inter-session variability modeling
JFA	joint factor analysis
LDA	linear discriminant analysis
LBP	local binary pattern
LLR	log-likelihood ratio
MAP	maximum a posteriori
MFCC	Mel frequency cepstrum coefficient
ML	maximum likelihood
MLP	multilayer perceptron

PCA	principal component analysis
PIE	pose, illumination and expression
PLDA	probabilistic linear discriminant analysis
ROC	receiver operating characteristic
SIFT	scale-invariant feature transform
SVM	support vector machine
UBM	universal background model
VAD	voice activity detection
TV	total variability modeling

1 Introduction

Humans have the ability to recognize people from various cues, such as face, voice or gait. This plays a central role in our social relationships, as it is a fundamental step in human interaction. Furthermore, we are now surrounded by electronic devices, with which we interact. The rapid development of ubiquitous computing and smart environments will take human-machine interaction to the next level. This requires automatic systems able *to recognize people* from various modalities. *Biometrics* is the field that addresses the problem of identifying humans by their traits or characteristics [Jain et al., 2007]. Reliable biometrics traits such as DNA, fingerprints or iris are available and widely used nowadays. Nevertheless, they rely on the active participation of the person. On the other hand, face and speaker recognition require very little cooperation from the person and are, thus, said to be non-intrusive.

Automatic face and speaker recognition are topics that have been under active research for more than two decades. The current state-of-the-art consists of systems that work well under controlled laboratory conditions, but are still severely impacted under the wildly varying conditions encountered in many real world scenarios.

This thesis is a step towards the development of more robust systems for automatic face and speaker recognition.

1.1 Background and Motivations

Automatic face and speaker recognition have recently received significant attention [Campbell, 1997, Li and Jain, 2005, Beigi, 2011]. This trend can be explained at least by the following two reasons. First, the progress in technology has encouraged the development and the deployment of such automatic systems not only on powerful servers or workstations, but also on mobile devices with limited hardware resources such as smartphones and tablets. Second, there is a broad range of possible commercial and law enforcement applications.

Some of these applications are listed below to highlight the wide applicability of these technologies.

1. **Access control:** Biometrics has been used to control access to physical facilities in high security areas. For instance, face recognition systems have been deployed in few airports [Spreeuwers et al., 2012], comparing the picture of a person taken on site with the one stored on his biometric passport.
   More recently, other applications have emerged, which control access to secure systems or services. Considering speaker recognition, automatic systems have been employed to verify the identity of a person, when accessing a service such as banking by telephone, voice mail or, more recently, electronic and mobile commerce [Reynolds, 2002]. Authentication on a mobile phone can also be achieved using face and speaker recognition technology [Marcel et al., 2010].
2. **Surveillance:** The deployment of closed-circuit television (CCTV) systems has led to a huge amount of information to be stored and processed. This is of particular interest in forensic science, since face recognition technology can be employed to reduce the quantity of information to be processed manually, when criminal or terrorist investigations are performed [Klontz and Jain, 2013].
   Similarly, speaker recognition can help police and government agencies to link recordings made in connection with criminal or terrorist activities (anonymous calls or telephone tapping) [Champod and Meuwly, 2000].
3. **Law enforcement:** Speaker recognition technology can help to enforce the law [Reynolds, 2002]. For instance, this can be used for home-parole monitoring, which consists of calling parolees to verify that they are at home when they have to. Another possible application is prison call monitoring, to verify the identity of an inmate prior to his outgoing call.
   It is also commonly admitted that CCTV contributes to law enforcement, although recent studies suggest that this has a larger effect on deterring thieves than violent crimes [Welsh and Farrington, 2002].
4. **Data management:** Person identification is particularly useful to automatically tag photos, audio and/or video content, or at least to reduce the amount of manual work required from the user. Companies such as Google, Microsoft, Facebook or Apple are already providing this feature in their image organizer and image viewer software [Kapoor et al., 2012]. Similarly, audio-visual mining applications are now able

to annotate recorded meetings or video with speaker labels to allow quick indexing and retrieval [Reynolds and Torres-Carrasquillo, 2005]. This process is called *speaker diarization*.

5. **Personalization:** Content personalization is commonly employed by online shops, to help them to propose offers that customers may find interesting. Similarly, recognition techniques could be used for directed advertisement, where former customers could be either identified or categorized based on facial and vocal characteristics (e.g., gender or age) [Reynolds, 2002].
   Another possible application would be the identification of a car driver to be able to automatically adjust his personal settings (seat, wheel and mirrors). This could be achieved by connecting visual or audio sensors to the onboard computer of the vehicle [Stallkamp et al., 2007]. More generally, smart environments are geared towards this idea [Ekenel et al., 2007].

If we formalize the goal in these applications, it appears that *recognition* refers to two different tasks: (1) The first one is *verification* (also called authentication), in which a person claims a particular identity (known as the *subject*'s identity), and the system has to verify this claim based on a biometric trait (e.g., his face or his voice). This means that the system has to decide whether the person is the true claimant or an impostor, which is a two-class classification problem. (2) The second task is *identification*, in which the system has to identify a person from a set of N possible persons, given a biometric trait. In this case, this is a N-class classification problem. This distinction is depicted in fig. 1.1. In addition, the terms "class", "subject", "client" and "identity" are used interchangeably in this thesis.

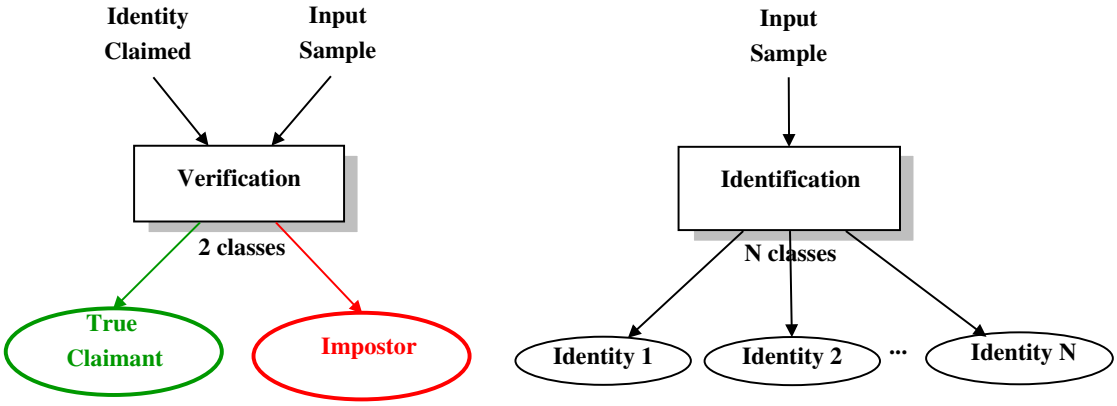


Figure 1.1 – PERSON RECOGNITION: VERIFICATION VS. IDENTIFICATION.

Humans have a strong ability to recognize people from their voice or their face [Balas et al., 2006]. In contrast, machines are usually explicitly programmed, which is not a suitable scheme for tasks as complex as face or speaker recognition. *Machine learning* is a branch of artificial intelligence that considers the construction and study of systems that can learn from data. This gives to computers the ability to learn to recognize without being explicitly programmed. In this case, data are usually distinguished in two groups: (1) *training data* are employed to learn the classification model, while (2) *test data* are used to query the system in a real-world

application or to evaluate the performance of the system before its deployment.

Many challenges are encountered in both face and speaker recognition, which often result in difficult classification problems. In automatic face recognition, changes in pose, illumination, expression (PIE) or image acquisition as well as accessories (e.g., glasses, hat) and occlusions significantly affect the appearance of the digital samples captured by a camera. Similarly, in automatic speaker recognition, the variability of the environment (e.g., acoustic of the room, background noise), the channels (e.g., using different microphones) and human factors (e.g., emotional state, aging) impact the resulting audio samples.
These challenges can be attributed to the problem of *session variability*, which is anything that causes a mismatch between samples of the same person (client). Session variability is also coined as *within-class variability* (and more rarely as intra-class variability). In the following, the two terms "session variability" and "within-class variability" will, hence, be used interchangeably.
In addition, the overall appearance of human faces (or voices) are similar. If we consider that each identity corresponds to a class, this means that these recognition problems have a low *between-class variability*.

More generally, solving a problem with a high within-class variability and a low between-class variability is a very challenging task. Fig. 1.2 highlights these difficulties, with concrete examples on face recognition.

Figure 1.2 – SAMPLES FROM THE LFW DATABASE. *This figure shows samples with high within-class variations from the Labeled Faces in the Wild (LFW) database. The first three samples are from the same female subject, and the last three are from the same male subject.*

1.2 Objectives and Contributions

The tasks of automatic face and speaker recognition are often addressed in completely different ways. Standard approaches for face recognition typically rely on discriminative techniques, such as *Eigenfaces* [Turk and Pentland, 1991b] or discriminant face descriptors [Lei et al., 2013]. In contrast, the standard approach for speaker recognition models audio segments using *Gaussian mixture models* (GMM).

The main objective of this thesis is to investigate the use of *probabilistic models* to address the problem of session variability in automatic face and speaker recognition. Specifically, we focus on a particular subset of techniques, which are *latent variable models* for classification tasks, that we apply separately to face, speaker and bimodal recognition. In addition, a recent

trend in machine learning and pattern recognition research is to go large scale, processing increasingly large amounts of data in an effort to improve recognition accuracy. Therefore, these models are required to be *scalable*, both at training and test time.

The major contributions of this thesis are as follows.

1. **An exact and scalable formulation of *probabilistic linear discriminant analysis*** (PLDA) is proposed. PLDA is a probabilistic approach that models within-class and between-class variations and that can be used for various tasks such as classification and clustering. Our main goal is to address the issue of scalability that is encountered at both training and test time. This is achieved by reformulating the problem using a wise change of variable that allows us to diagonalize the model. This leads to a **significantly reduced complexity**:
   - At training time, the time complexity becomes linear instead of cubic, and the memory complexity becomes constant instead of quadratic, both with respect to the number of training samples per class.
   - At test time, when computing the joint likelihood of several samples, the time complexity becomes linear instead of quadratic and the memory complexity becomes constant instead of quadratic.

   In particular, this novel formulation allows the use of the PLDA model for **large-scale applications**.
   Related papers for this contribution: [El Shafey et al., 2013c, McCool and El Shafey, 2013]

2. We integrate **three session variability modeling techniques**, *inter-session variability modeling* (ISV), *joint factor analysis* (JFA), *total variability modeling* (TV) into a unified framework. These techniques all rely on Gaussian mixture models (GMM). We show the similarities and the differences between these techniques, both theoretically and visually by using a synthetic dataset specifically created for this purpose. Furthermore, we describe how the **training procedure** for these models can be **parallelized** to take advantage of recent multi-core processors and computer clusters for large-scale experiments.

3. We successfully **apply the proposed approach** to the following three tasks.
   First, **face recognition** experiments are conducted on a wide range of publicly available and challenging databases, with different recording conditions. In particular, the impact of pose, illumination and expression variations, as well as occlusions is empirically assessed. On the FRGC database, a relative improvement in the CAR at FAR = 0.1% (see sec. 2.6) of 33% is observed with the **TV-PLDA** system on experiment *2.0.1* when compared to the **GMM** baseline.
   Related papers for this contribution: [McCool et al., 2013, Günther et al., 2013]
   Second, **speaker recognition** experiments are conducted on a large-scale. In particular, we built a speaker recognition system based on inter-session variability modeling (ISV),

when we took part in the *NIST Speaker Recognition Evaluation 2012* (NIST SRE12). [1] This thesis encompasses and extends our submission to NIST SRE12, by considering and evaluating all the modeling techniques introduced above. A relative improvement in HTER (see sec. 2.6) of up to 47% is observed with **TV-PLDA** when compared to the **GMM** baseline.
Related papers for this contribution: [Khoury et al., 2012, Saedi et al., 2013]
Third and finally, **bimodal (face and speaker) recognition** is investigated, using high level fusion. We show that bimodal recognition is of particular interest when a modality is strongly affected by challenging conditions, since the other one is then often available to come to the rescue.
Related papers for this contribution: [Motlicek et al., 2012, Khoury et al., 2013a,b]

4. We present **Bob**, [2] **an open source framework for signal processing and machine learning**. The probabilistic models presented in this thesis are all implemented and integrated into this library. In addition, Bob aims to encourage reproducible research, by allowing any researcher:
   – to distribute source code in a satellite package that may rely on any feature of the Bob toolkit (e.g., machine learning or signal processing algorithms).
   – to provide step-by-step instructions for combining the source code and the data, which should then allow anyone to reproduce results or plots from the corresponding article.

   Finally, **the results and plots presented in this thesis can be easily regenerated**, by following a set of instructions mentioned in the satellite package `xbob.thesis.elshafey2014` [3] associated with this dissertation.
   Related papers for this contribution: [El Shafey et al., 2013a, Anjos et al., 2012]

In addition to these primary contributions, a secondary contribution of this thesis is as follows.

5. We adapt and **apply the session variability modeling techniques** to the two-class classification problem of **gender recognition**, using both visual and acoustic cues. Experimental results show that these approaches offer appealing performances, for both the visual and the acoustic modality, the resulting bimodal system achieving an accuracy of 99%. In addition, and consistent with our conclusions on bimodal person recognition, bimodal gender recognition provides large improvements when one of the modalities is affected by challenging conditions.
   Related paper for this contribution: [El Shafey et al., 2013b]

1.3 Thesis Outline

This thesis is composed of eight chapters.

1. http://www.nist.gov/itl/iad/mig/sre12.cfm
2. http://www.idiap.ch/software/bob
3. https://pypi.python.org/pypi/xbob.thesis.elshafey2014

In this first chapter, the motivations, objectives and contributions of this work were briefly summarized.

Chapter 2 gives an overview of related work, separately for the tasks of automatic face and speaker recognition. In addition, this chapter introduces a set of metrics and evaluation measures, which are used to compare the proposed systems in the experimental part.

In Chapter 3, the Gaussian mixture model (GMM) framework for classification is introduced. Next, three session variability modeling techniques that are built on top of this framework are described: inter-session variability modeling (ISV), joint factor analysis (JFA) and total variability modeling (TV). In particular, the similarities and the differences between these approaches are highlighted.

Chapter 4 introduces probabilistic linear discriminant analysis (PLDA), which is another technique to address the problem of session variability. In particular, we propose an exact and scalable formulation of this model, which drastically reduces the computational complexity at both training and test time. This novel formulation, hence, allows the use of this model for large-scale applications.

Chapter 5 describes the application of the proposed approach to the task of face recognition. Experimental studies on a wide range of publicly available databases and under different experimental conditions are reported. Evaluation protocols propose both verification and identification scenarios.

Chapter 6 describes the application of the proposed approach to the task of speaker recognition. The experiments are conducted on the large and recent NIST SRE12 corpus.

Chapter 7 describes the application of the proposed approach to the task of bimodal (face and speaker) recognition. The experiments evaluate both bimodal and multi-algorithm fusion at the score level. In particular, scenarios, where there is significant condition mismatch, are investigated.

Chapter 8 concludes this thesis by providing a summary of the major contributions and findings. Potential directions for future work are also discussed.

In the appendices, we first describe Bob (appendix A), our open source framework for signal processing and machine learning, as well as the satellite package associated with this thesis. Together, they are convenient companions of this dissertation, which allow to easily regenerate all the reported results and plots. Second, we describe the application of the proposed approach to the task of bimodal (face and speaker) gender recognition (appendix B).

1.4 Notation

Scalars are indicated with lower case letters (e.g., x and γ), vectors with bold lower case letters (e.g., $\boldsymbol{x}$ and $\boldsymbol{\gamma}$) and matrices with bold upper case letters (e.g., $\boldsymbol{X}$ and $\boldsymbol{\Gamma}$). Sets of values are denoted using the "\mathbb" type (e.g., $\mathbb{O}$ or $\mathbb{R}$). An exception is $\boldsymbol{\Theta}$, which is used to denote the set of parameters of a model.

2 Related Work

Face and speaker recognition are usually addressed in different ways by the biometric community. Nevertheless, they are classification tasks and existing approaches usually rely on three key components [Duda et al., 2000], which are depicted in fig. 2.1.

The first component is *segmentation*, which aims at localizing the useful information inside the captured input signal. For instance, an automatic face recognition system needs to determine the location of a face prior to recognition. Similarly, an automatic speaker recognition system needs to know when the person is speaking within an audio sample.

The second component is *feature extraction*, which maps the input signal (raw pixel image or digital audio signal) to a *feature vector* belonging to the so-called feature space. This step aims at enhancing the important information of the input, while suppressing redundant or irrelevant information. For instance, in the context of face recognition, feature extraction can reduce the impact of illumination variation, whereas in the context of speaker recognition, it can improve the signal-to-noise ratio.

The third step is performed by a *classifier*, which is in charge of assigning a class label (e.g., true claimant or impostor in a verification scenario) to the feature vector extracted from the input signal. Explicit programming is not a suitable scheme for such complex tasks, which have a high within-class (session) variability and a low between-class variability. Therefore, classification is usually achieved using a machine learning algorithm. These algorithms first learn a model on a specific training set of samples. The model can then be applied to make the decision.

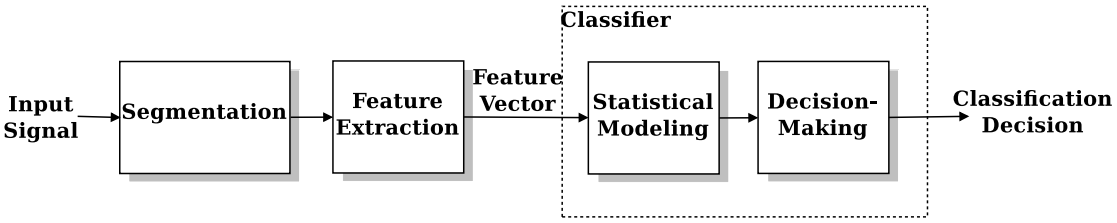


Figure 2.1 – SIMPLIFIED STRUCTURE OF A TYPICAL RECOGNITION SYSTEM.

Many challenges in face and speaker recognition can be attributed to the problem of session variability. Session variability is anything that causes a mismatch between samples of the same class. For instance, in face recognition this includes changes in illumination, pose, expression or image acquisition, whereas in speaker recognition it consists of variations in microphones, acoustic environments or transmission channels.

The feature extraction step should ideally lead to a new representation, where the effect of these variations is not visible. In practice, this can often not be achieved by such a mapping in a feature space. Learning algorithms can then be of great help.

The fields of face and speaker recognition are very broad. In the remainder of this chapter, we review some of the existing work and discuss how we restrict this thesis to a specific subset of machine learning algorithms. First, we introduce standard approaches for face and speaker recognition. Second, we discuss classification in machine learning and highlight what is specific to these two tasks. Third, we review existing work in two categories of machine learning algorithms: subspace-based and latent variable-based models. Fourth and finally, we introduce a set of evaluation measures that are employed to empirically assess the performance of the proposed systems.

2.1 Standard Approaches

2.1.1 Face Recognition

The first popular approach in automatic face recognition was developed at the end of the eighties [Sirovich and Kirby, 1987]. It relies on an holistic representation, in which the whole face region is fed to a classifier. It is well known that natural images contain significant statistical redundancies [Ruderman and Bialek, 1993], especially after the normalization of face regions with respect to scale, translation and rotation. From a statistical perspective, this redundancy means that the pixel values of the input image are *correlated variables*.

One of the most famous representations that addresses this issue is called Eigenfaces [Sirovich and Kirby, 1987, Kirby and Sirovich, 1990, Turk and Pentland, 1991a,b], which is based on *principal component analysis* (PCA). PCA is a statistical procedure that computes a transformation to convert a set of possibly correlated input variables into linearly uncorrelated output

variables, called principal components. This transformation is linear and can be represented by a projection matrix $\boldsymbol{W}_{\mathbf{PCA}}$, which is learned from a set of training samples. Formally, a feature vector $\boldsymbol{o}$ of dimensionality $D_{\boldsymbol{o}}$ is transformed into a projected vector $\boldsymbol{t}$ of dimensionality $D_{\boldsymbol{t}} \leq D_{\boldsymbol{o}}$ as follows:

$$\boldsymbol{t} = \boldsymbol{W}_{\mathbf{PCA}} \boldsymbol{o} \,. \tag{2.1}$$

This model, hence, leads to a new lower-dimensional representation of a face. At the decision-making time, these new representations of images are compared using a metric such as the Euclidean distance or the cosine similarity measure. From a machine learning perspective, this approach is *discriminative*: given a feature vector $\boldsymbol{o}_{\text{test}}$ (the observed variable) extracted from a sample $\boldsymbol{\chi}_{\text{test}}$, the aim is to directly predict the corresponding class i (the unobserved variable), without specifying the joint distribution of $\boldsymbol{o}_{\text{test}}$ and i.

2.1.2 Speaker Recognition

In the field of automatic speaker recognition, a considerable amount of work has been conducted in industry, laboratories, research institutes and universities, since the beginning of the seventies (see [Campbell, 1997] for a survey). Early systems were operating in a text-dependent mode, where the speakers had to utter the same words in order to be recognized. Later, systems operating in text-independent mode were proposed, and this has now become the main trend.

Of particular interest, Reynolds' Gaussian mixture model [Reynolds, 1992, Reynolds and Rose, 1995, Reynolds, 1995a] has built the foundation of current state-of-the-art speaker recognition systems. After segmentation, this approach consists of decomposing the audio signal in a set of overlapping windows of short duration (in the order of tens of milliseconds). A feature vector $\boldsymbol{o}$ is then extracted from each window. For each class i, the probability distribution $P(\boldsymbol{o} \mid i)$ of these feature vectors is modeled using a mixture of Gaussians with parameters $\boldsymbol{\Theta}_i$. At test time, this distribution is used to determine the probability that a test sample $\boldsymbol{\chi}_{\text{test}}$ has been generated by the class i. From a machine learning perspective, this approach is *generative*: given a feature vector $\boldsymbol{o}$ extracted from a sample $\boldsymbol{\chi}$, the distribution of the inputs $P(\boldsymbol{o} \mid i)$ is explicitly modeled, before estimating the posterior probabilities $P(i \mid \boldsymbol{o})$.

2.2 Classification for Face and Speaker Recognition

Considering the previous two approaches, it appears that a discriminative model is employed for face recognition, whereas a generative model is used for speaker recognition. More generally, the choice of a specific machine learning algorithm depends on the particular problem.

For classification tasks, support vector machines (SVM) [Vapnik, 1995] and multilayer per-

ceptrons (MLP) [Bishop, 2007] are very popular in machine learning. SVMs have indeed been used by researchers for both face [Phillips, 1999, Heisele et al., 2001] and speaker recognition [Campbell et al., 2004, 2006a]. This is also the case of MLPs [Lin et al., 1997, Cardinaux et al., 2003, Farrell et al., 1994].

There are different ways of addressing the problem of face and speaker recognition. One possibility is to rely on a single binary classifier that compares pairs of samples and tells whether the samples are from the same class or not. A drawback is that such an approach does not generalize well for authentication or identification scenarios when several *enrollment* samples are available for a specific class. Indeed, the outputs of several comparisons between the test sample and the class-specific (enrollment) samples of a given subject (class) then need to be combined using a heuristic.

An alternative is to *enroll* a class-specific model for each subject. The classifier can then determine whether a test sample belongs to the same class as one of the enrollment samples or not. The number of enrollment samples for a specific class is typically small. In addition, the training set is *imbalanced* in this case, which means that it contains much more negatives (non-class) samples than class-specific samples. Training procedures for classifiers such as SVMs and especially MLPs, as mentioned above, are usually suboptimal in this case.

More generally, the comparison of sets of samples instead of pairs of samples is a current line of research in face recognition [Arandjelović et al., 2005, Kim et al., 2010]. This is also occurring when matching video data [Poh et al., 2010]. In speaker recognition, the same problem is intrinsically encountered. Several *observations* are typically extracted from audio samples of various duration, and comparisons between these sets of observations are then performed.

2.3 Variability Modeling using Subspaces and Manifolds

The PCA approach estimates a subspace that captures the main directions of variability. Several other machine learning techniques rely on subspaces or manifolds to model variability.

Similar to PCA, *linear discriminant analysis* (LDA) [Fisher, 1922] seeks for a subspace $\boldsymbol{W}_{\mathbf{LDA}}$ that best describes the variability of the samples. However, and in contrast to PCA, the estimation of the projection matrix $\boldsymbol{W}_{\mathbf{LDA}}$ is performed in a *supervised* way, by making use of the class label associated with each feature vector. Within-class and between-class scatter matrices are computed, before optimizing a criterion (called the Fisher's criterion) that aims at maximizing the separability between classes in the projected space.

Both PCA and LDA lead to a linear projection of the feature vector. There has been a significant amount of work to propose systems that offer nonlinear projections. In particular, Kernel LDA [Mika et al., 1999, Baudat and Anouar, 2000] is a natural extension of LDA for the nonlinear case, which employs the *kernel function* operator. The main idea is to map the input feature space into a more convenient space in which variables are nonlinearly related to the input

space. If ϕ is the function that describes this mapping between the input feature space and this higher dimensional feature space $\mathcal{F}$, the kernel function k computes the dot product in this space $\mathcal{F}$:

$$k(\boldsymbol{o}_1, \boldsymbol{o}_2) = \phi(\boldsymbol{o}_1) \cdot \phi(\boldsymbol{o}_2) \tag{2.2}$$

In practice, the formulation of Kernel LDA can be done in terms of only dot products of input feature vectors. An expression of the kernel function is then sufficient and there is no need to know the exact mapping ϕ. This is known as the *kernel trick*.

Nonlinear subspace methods have also been proposed to compare sets of samples [Hamm and Lee, 2008]. In this case, a set of samples is seen as a linear subspace and comparison can then be performed by exploiting the fact that these subspaces can be described within a Grassmann manifold.

2.4 Latent Variable Models

A *latent variable model* is a statistical model that seeks to relate a set of *observed variables* (or observations) to a set of *latent variables*. For instance, in a classification task the observed variables might be the feature vectors, whereas the latent variables describe the associated classes.

One of the most well known latent variable models is *factor analysis* [Bartholomew et al., 2011, Basilevsky, 2009], which assumes a linear relationship between the variables:

$$\boldsymbol{o} = \boldsymbol{\mu} + \boldsymbol{A}\boldsymbol{t} + \boldsymbol{\epsilon}. \tag{2.3}$$

In this model, the latent variable $\boldsymbol{t}$ is assumed to be independent and to follow a normal distribution $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. The latent variable $\boldsymbol{t}$ is of dimensionality $D_{\boldsymbol{t}}$, whereas the observation $\boldsymbol{o}$ is of dimensionality $D_{\boldsymbol{o}}$ (with $D_{\boldsymbol{t}} \leq D_{\boldsymbol{o}}$). The vector $\boldsymbol{\mu}$ allows the model to have a non-zero mean, while the matrix $\boldsymbol{A}$ of dimension $(D_{\boldsymbol{o}}, D_{\boldsymbol{t}})$ relates the two sets of variables. Assuming that the noise (or residual) $\boldsymbol{\epsilon}$ has a Gaussian distribution $\mathcal{N}(\boldsymbol{0}, \boldsymbol{\Psi})$, the observation $\boldsymbol{o}$ follows a Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{A}\boldsymbol{A}^\top + \boldsymbol{\Psi})$.

A particular case of special interest is when there is an assumption of isotropic Gaussian distribution $\mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$ for the noise $\boldsymbol{\epsilon}$. The corresponding model is then known as *probabilistic principal component analysis* [Tipping and Bishop, 1999, Roweis, 1998], which is a probabilistic formulation of the PCA model introduced earlier. With this additional assumption, the

conditional probability distribution of the observed variable $\boldsymbol{o}$ given the latent variable $\boldsymbol{t}$ is:

$$P(\boldsymbol{o} \mid \boldsymbol{t}) = \mathcal{N}(\boldsymbol{\mu} + \boldsymbol{A}\boldsymbol{t}, \sigma^2 \boldsymbol{I}). \tag{2.4}$$

There are several advantages of defining such a probabilistic model compared to regular PCA. First, PCA requires the diagonalization of a covariance matrix, which is computationally expensive when both the number of samples and their dimensionality are large. Second, PCA is not able to properly handle missing data. This means that incomplete samples must either be discarded or completed using an interpolation method. The probabilistic version of PCA is able to address these limitations.

More generally, latent variable models have the advantage of defining a proper probability model in the space of inputs. Results can then be interpreted in terms of probabilities, rather than using empirical measures. For instance, to measure how well a data point fits the training data used for PCA, a common criterion is to rely on the Euclidean distance of this data point to its projection into the principal subspace. In contrast, probabilistic PCA answers this question using probabilities.

Similarly, the Gaussian mixture model (GMM) introduced earlier in this section has a formulation in terms of a latent variable model. In this case, the latent variables are discrete [Bishop, 2007]. In the following chapter, we will introduce a different latent variable formulation for GMMs.

2.5 Latent Variable Models and Subspaces

In this thesis, we investigate a set of latent variable models that model variability with subspaces. These techniques are suitable to enroll class-specific models from a limited number of enrollment samples and to compare sets of samples.

Three of the most successful techniques in improving robustness of speaker recognition model variability with subspaces: *inter-session variability modeling* (ISV) [Vogt et al., 2005], *joint factor analysis* (JFA) [Kenny et al., 2007] and *total variability modeling* (TV) [Dehak et al., 2009, Dehak, 2009]. Another latent variable model known as *probabilistic linear discriminant analysis* (PLDA) [Prince and Elder, 2007] has been proposed in the field of face recognition. All of these models combine the strength of using subspaces to model variability (within-class, between-class or total) while retaining a probabilistic interpretation. They are described in details in the following chapters before being applied to the tasks of face and speaker recognition.

2.6 System Evaluation and Performance Measure

When evaluating a recognition system, samples of a database are typically employed for two different purposes: to train a model or to probe the system. As pointed in the previous section, it is often worthwhile to be able to enroll a model for a new class without the need of retraining an expensive model from scratch. This requires separate data for training a generic model and for enrolling class-specific models.

Evaluation protocols define which samples of a database should be used: (1) to train a generic model, (2) to enroll class-specific models and (3) to probe the system. As opposed to the *training set*, the *evaluation set* denotes enrollment and probe samples. Given a class-specific model $\mathscr{S}_i$ (generated from a set of enrollment samples) and a probe sample $\boldsymbol{\chi}_{\text{test}}$ (or several of them), a recognition system generates a score $h\left(\boldsymbol{\chi}_{\text{test}} \mid \mathscr{S}_i\right)$. To avoid biased evaluations, a good practice consists of having different classes in the training and evaluation sets.

Classification systems commonly have several parameters to tune. When optimizing parameters on the evaluation set, there is a risk of *overfitting*, which means that the system will offer very good performance on the evaluation set, but will not generalize well.

In order to avoid overfitting, different strategies are possible. A popular approach in biometrics relies on the definition of a *development set* (also called validation set). Similarly to the evaluation set, this development set consists of enrollment and probe samples and it should be used to optimize the parameters of a system. In contrast, the evaluation set is only employed for assessing the performance. Furthermore, if a model overfits, it will lead to good performance on the development set, but this will not generalize well on the evaluation set. Again, to avoid biased evaluations, the classes in the training, development and evaluation sets should ideally be disjoint (cf. tab. 2.1).

Another approach to avoid overfitting is *cross-validation*, a common variant being *n-fold cross-validation*. In this scenario, the database is randomly partitioned into n folds of equal size. The cross-validation process is repeated n times, by considering one fold for evaluating the model and the remaining $n-1$ for training. The n results from the folds can then be combined to produce a single performance measure.

Table 2.1 – TYPICAL EVALUATION PROTOCOL IN BIOMETRICS. *This table shows how classes (identities) are split into different subsets in a typical evaluation protocol for biometric systems. In an unbiased protocol, there is no overlap between classes of the three subsets: training, development and evaluation.*

Classes (Identities)		
Training	**Development**	**Evaluation**
	Enrollment Probe	Enrollment Probe

Once an evaluation protocol has been defined, the performance of a classifier can be assessed.

Several metrics have been proposed for this purpose, and their popularity typically depends on the field.

2.6.1 Verification Measures

In a verification scenario, the decision-making process consists of comparing a score $h\left(\boldsymbol{\chi}_{\text{test}} \mid \mathscr{S}_i\right)$ with a threshold θ to output a decision of *acceptance* or *rejection*. When a score is higher than the threshold, it is accepted and the class of the probe sample $\boldsymbol{\chi}_{\text{test}}$ is considered to be the same as the one of the model $\mathscr{S}_i$.

A verification system can produce two different types of errors:
– *false acceptance* (FA) if the system has wrongly accepted an *impostor*,
– *false rejection* (FR) if a *true claimant* (also called true, genuine or legitimate client) has been rejected.
By splitting up the scores into *true claimant scores* and *impostor scores*, we can define the *false acceptance rate* (FAR) and *false rejection rate* (FRR):

$$\text{FAR}(\theta) = \frac{|\text{FA}|}{|\text{impostor accesses}|}, \tag{2.5}$$

$$\text{FRR}(\theta) = \frac{|\text{FR}|}{|\text{true claimant accesses}|}, \tag{2.6}$$

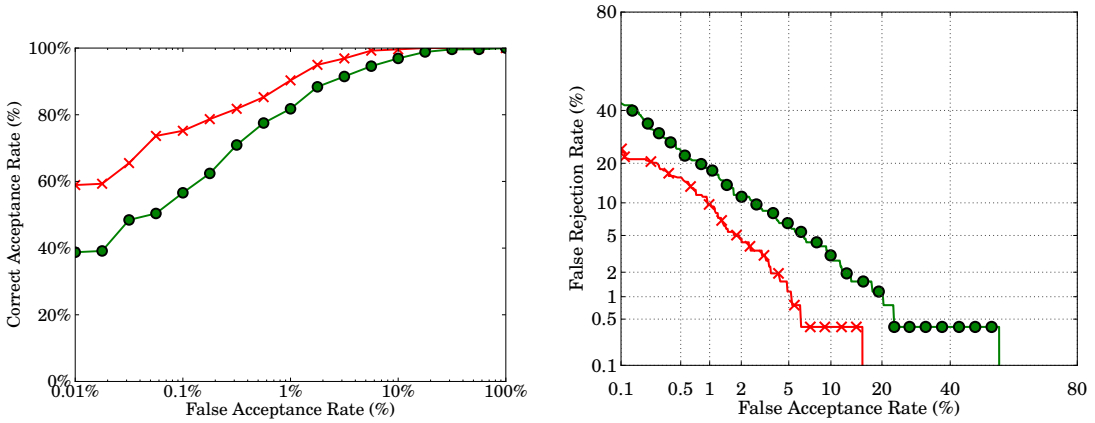
where θ is the decision threshold.

Another widely used measure is the *half total error rate* (HTER), which summarizes the FAR and the FRR into a single value as follows:

$$\text{HTER}(\theta) = \frac{\text{FAR}(\theta) + \text{FRR}(\theta)}{2}. \tag{2.7}$$

A common way to select a decision threshold θ is to look for the *equal error rate* (EER), which is the rate, at which the FAR is equal to the FRR.

A limitation when reporting EER or HTER values is that they describe the performance of a system for a specific operating point. In addition, the FAR and the FRR are highly correlated: depending on the threshold θ, increasing the FAR will reduce the FRR, and vice versa. A possible representation to show the performance of a system at various operating points is the *receiver operating characteristic* (ROC) curve. An ROC curve plots the *correct acceptance rate* (CAR = 1 − FRR) against the FAR. To improve readability, a logarithmic scale for the x-axis is often used. Another related representation is the *detection error tradeoff* (DET) curve, which plots the FAR against the FRR, using axes that are scaled nonlinearly by their standard normal deviates [Martin et al., 1997]. This yields tradeoff curves that are more linear than ROC ones. Examples of ROC and det curves are shown in fig. 2.2.



(a) Example of an ROC curve

(b) Example of a DET curve

Figure 2.2 – EXAMPLES OF ROC AND DET CURVES. *This figure shows examples of ROC curves (a) and of DET curves (b).*

2.6.2 Identification Measures

In an identification scenario, every probe sample is compared with all the class-specific models stored within the system. The decision-making process then consists of returning the set of n classes (models) that are similar to the one of the probe sample. In practice, this is achieved by returning the n classes (models) corresponding to the n largest scores $h\left(\boldsymbol{\chi}_{\text{test}} \mid \mathscr{S}_i\right)$ obtained with the probe sample $\boldsymbol{\chi}_{\text{test}}$. The identification of a probe sample is correct when its class belongs to the returned set of n classes. If the model corresponding to the probe sample gives the r^{th} largest score, the *rank* of this probe sample is said to be equal to r.

The identification performance of a system can be represented using a *cumulative match characteristics* (CMC) curve. For each value r, the CMC curve displays how many probe samples have a rank r or lower, normalized by the total number of probe samples. When $r = 1$, the corresponding measure is known as the *recognition rate* (RR). To improve legibility, a logarithmic scale for the x-axis might be used. Example of CMC curves are shown in fig. 2.3.

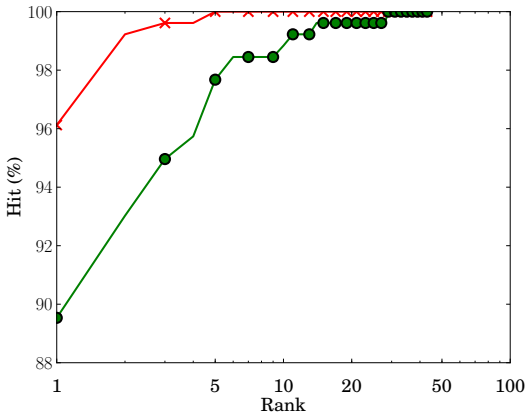


Figure 2.3 – EXAMPLES OF CMC CURVES.

3 GMM-based Latent Variable Models

In this chapter, we describe several latent variable models that are built on top of a Gaussian mixture model (GMM) classification framework. These probabilistic models are scalable and address the problem of session variability that is encountered in several classification tasks.

We present a self-contained description of these techniques and explain how to apply them to face and speaker recognition, despite the very distinctive nature of the signals considered (visual and audio, respectively). Furthermore, the similarities and the differences between these approaches are highlighted both theoretically and empirically using a synthetic dataset.

3.1 Introduction

The problem of session variability is common to several fields such as face or speaker recognition. While face and speaker recognition systems offer good performance under controlled conditions, there is still room for improvement in more realistic uncontrolled scenarios. In particular, this has been part of the motivation for the collection of recent face and speaker databases such as MOBIO [McCool et al., 2012] and Multi-PIE [Gross et al., 2008], which contain several types of variabilities across recordings.

A large variety of approaches have been proposed in both fields. In the speaker recognition community, one of the most popular approaches is the Gaussian mixture model (GMM) framework [Reynolds et al., 2000], which is a generative modeling technique. Recently, several techniques were proposed on top of this approach, which attempt at modeling detrimental session variations to improve the robustness of recognition systems. We investigate three such methods in the remainder of this chapter: *inter-session variability* (ISV) modeling [Vogt and Sridharan, 2008], *joint factor analysis* (JFA) [Kenny et al., 2007] and *total variability* (TV) modeling [Dehak et al., 2011]. ISV and JFA aim to explicitly model and remove within-class variation using a low-dimensional subspace. JFA can be considered to be an extension of ISV as it additionally utilizes a between-class subspace to capture important discriminative class-specific information. Similarly to ISV and JFA, TV (also known as *i-vector modeling*) is

built on top the GMM framework. However, this is an unsupervised technique, and session compensation is carried out separately in a low-dimensional space after the extraction of low-dimensional i-vectors.

Miris Synthetic Dataset

To visualize the behavior of these techniques, in particular the subspaces learned, we generate a synthetic dataset. This toy example is inspired from the Iris flower dataset introduced in [Fisher, 1936], considering only two dimensions. It consists of four classes, three of them being part of the training set, while the last one is used to enroll a class-specific model. This dataset is depicted in fig. 3.1. In the following, all the figures that are generated using this dataset are reproducible using the satellite package associated with this dissertation (see sec. A.3.2).

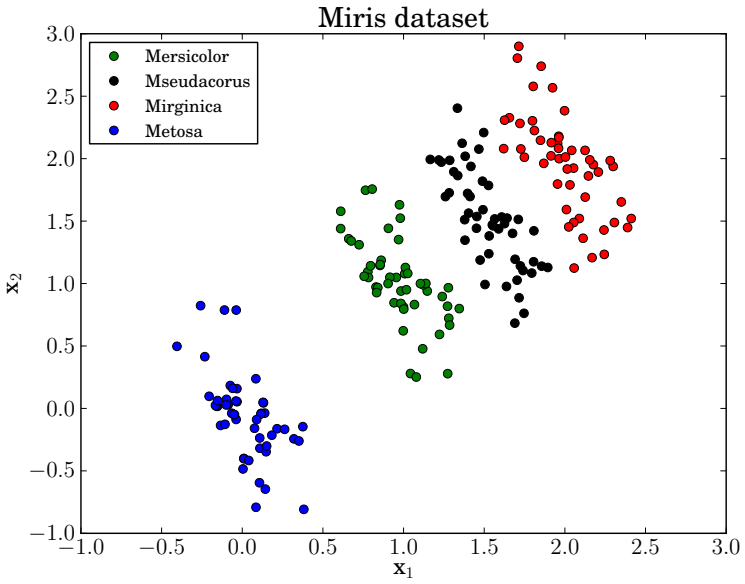


Figure 3.1 – MIRIS SYNTHETIC DATASET. *This figure shows the Miris synthetic dataset. It consists of four classes, three of them being used for training (Mersicolor, Mseudacorus and Mirginica) and the remaining one for enrollment (Metosa). Each sample consists of five feature vectors (observations/dots). The samples are not shown on the plot.*

3.2 Feature Representation

When using latent variable models for face and speaker recognition, the same underlying approach is taken. The main difference is how the feature vectors are extracted from the image (face) and audio (speech) samples. For both modalities, a biometric sample $\boldsymbol{\chi}$ (image or audio) is decomposed into a set $\mathbb{O}$ of K feature vectors ($\mathbb{O} = \{\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_K\}$), where each feature vector is of dimensionality $D_{\boldsymbol{o}}$. This decomposition is performed in the spatial domain for the image

data, and in the time domain for the audio data. For instance, visual features might consist of a set of 2D *discrete cosine transform* (DCT) coefficients extracted from a regular grid of blocks in an image [Sanderson and Paliwal, 2003] (see also sec. 5.3.3). Considering the audio modality, features might consist of a set of *Mel frequency cepstrum coefficients* (MFCCs) extracted on overlapping time frames (see sec. 6.3).

Whenever required, the notation $\boldsymbol{\chi}_{i,j}$ will be used to describe the j^{th} sample of the class i. The corresponding set of $K_{i,j}$ extracted feature vectors is then written $\mathbb{O}_{i,j} = \left\{\boldsymbol{o}_{i,j,1}, \boldsymbol{o}_{i,j,2}, \ldots, \boldsymbol{o}_{i,j,K_{i,j}}\right\}$.

3.3 Gaussian Mixture Model

3.3.1 Model Formulation

A *Gaussian mixture model* (GMM) is a probabilistic model for density estimation. It corresponds to a mixture (or weighted sum) of Gaussian distributions that is used to represent the distribution of a set of feature vectors.

More formally, a GMM consists of C multivariate Gaussian components, each of dimensionality $D_{\boldsymbol{o}}$. Each component is defined by its weight, ω_c, mean vector, $\boldsymbol{\mu}_c$, and covariance matrix, $\boldsymbol{\Sigma}_c$, so that the corresponding probability density function is given by:

$$P\left(\boldsymbol{o} \mid \boldsymbol{\Theta}\right) = \sum_{c=1}^{C} \omega_c \mathcal{N}\left[\boldsymbol{o} \mid \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c\right], \tag{3.1}$$

where $\boldsymbol{\Theta} = \{\omega_c, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c\}_{c=\{1,\ldots,C\}}$ are the parameters of this model.

In addition, the mixtures weights, ω_c, must satisfy the constraints $0 \leq \omega_c \leq 1$ and $\sum_{c=1}^{C} \omega_c = 1$.

Given a set of M-dimensional feature vectors $\mathbb{O} = \{\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_K\}$, an interesting property of multivariate Gaussian distributions is that the sample mean (first-order statistic) and the sample covariance (second-order statistic) of these points define a *sufficient statistic*. A statistic is said to be sufficient for a family of probability distributions, if the points from which it is calculated give no additional information than does the statistic [Fisher, 1922]. This is of particular interest when looking for *maximum likelihood* estimators [Bishop, 2007] that are computed for the models described in this chapter. For a Gaussian mixture model and given a sample $\boldsymbol{\chi}$ with corresponding feature vectors $\mathbb{O}$, the sufficient statistics consist of the zeroth-,

first- and second-order statistics wrt. to each component c:

$$n_c(\mathbb{O}) = \sum_{k=1}^{K} \gamma_c(\boldsymbol{o}_k), \tag{3.2}$$

$$\boldsymbol{f}_c(\mathbb{O}) = \sum_{k=1}^{K} \gamma_c(\boldsymbol{o}_k)\boldsymbol{o}_k, \tag{3.3}$$

$$\boldsymbol{S}_c(\mathbb{O}) = \sum_{k=1}^{K} \gamma_c(\boldsymbol{o}_k)\left(\boldsymbol{o}_k - \boldsymbol{f}_c(\mathbb{O})\right)\left(\boldsymbol{o}_k - \boldsymbol{f}_c(\mathbb{O})\right)^{\top}, \tag{3.4}$$

respectively, where the term $\gamma_c(\boldsymbol{o}_k)$ is the occupation probability (also called responsibility or posterior probability) of the k^{th} observation of the sample $\boldsymbol{\chi}$ for the c^{th} component:

$$\gamma_c(\boldsymbol{o}_k) = \frac{\omega_c \mathcal{N}\left[\boldsymbol{o}_k \mid \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c\right]}{\sum_{c=1}^{C} \omega_c \mathcal{N}\left[\boldsymbol{o}_k \mid \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c\right]}. \tag{3.5}$$

The occupation probability express how well a component c of a GMM represents a given feature vector (or observation).

A classification approach based on GMMs has been used successfully for both face [Sanderson and Paliwal, 2003, Lucey and Chen, 2004, Cardinaux et al., 2006] and speaker recognition [Reynolds, 1992, Reynolds and Rose, 1995, Reynolds, 1995a, Reynolds et al., 2000]. This approach decomposes the input signal (face or speech) into a set of overlapping observations (in the spatial domain or temporal domain), which are considered to be separate observations of the same signal. One of the main motivations is that a linear combination of Gaussian functions allows to represent a wide class of sample distributions, ranging from smooth to arbitrarily-shaped densities. In addition, it was found to offer a good trade-off in terms of complexity, robustness and discrimination [Cardinaux et al., 2003, 2006].

GMMs can have several different organizations and types of covariance matrices [Reynolds and Rose, 1995]. The most general case is to have one full covariance matrix per Gaussian component per class-specific model (nodal covariance), but alternatives were investigated. One possibility is to consider a single covariance matrix shared by all Gaussian components in a class-specific model (grand covariance). Another option, which is even more restrictive, is to consider a single covariance matrix shared by all class-specific models (global covariance). Another aspect is the shape of the covariance matrix, full rank and diagonal matrices being very popular. In this work, we consider diagonal covariance matrices. This is motivated by previous work, as it was shown that the density modeling of a full covariance GMM can equally well be achieved using a diagonal covariance GMM with more Gaussian components. In addition, diagonal covariance matrices are very efficient to invert compared to full rank matrices, which is usually required several times when training a GMM.

The key aspects for applying GMMs to face or speaker recognition are: (1) how to obtain the

features (preprocessing and feature extraction), (2) how to generate a model of each class (enrollment), and (3) how to perform authentication given a probe sample and a claimed class (testing).

Step (1) has been succinctly described in the previous section (sec. 3.2), and steps (2) and (3) are described below.

3.3.2 Training

When using GMMs for recognition, the model $\mathscr{S}_i$ for the class (subject, identity or client) i is a GMM. This GMM is learned from a set of enrollment samples. Several techniques have been proposed to estimate the parameters of a GMM [Mclachlan and Basford, 1988, Mclachlan and Peel, 2000]. One of the main challenges is that the number of enrollment samples per class is usually limited, possibly to a single sample. To address this issue, it has been shown that for both speaker [Reynolds et al., 2000] and face authentication [Lucey and Chen, 2004, Cardinaux et al., 2006] an efficient enrollment method is to use a class-independent prior GMM $\mathscr{M}$, called the *universal background model* (UBM), and to adapt this prior to the enrollment samples of the class i to generate the class-specific model $\mathscr{S}_i$.

Training a Universal Background Model (UBM)

One of the most popular techniques to estimate the parameters of a GMM is *maximum likelihood* (ML) estimation. ML is applied to learn a UBM $\mathscr{M}$ by maximizing the likelihood of the GMM parameters given observations extracted from a large independent training set of several identities. Given a training set $\mathbb{O}_{\text{train}} = \{\boldsymbol{o}_1, \ldots, \boldsymbol{o}_{K_{\text{train}}}\}$ of K_{train} feature vectors and assuming that these feature vectors are independent, the GMM likelihood is given by:

$$P(\mathbb{O}_{\text{train}} \mid \boldsymbol{\Theta}) = \prod_{k=1}^{K_{\text{train}}} P(\boldsymbol{o}_k \mid \boldsymbol{\Theta}). \tag{3.6}$$

There is no closed form solution for the maximization of this function of $\boldsymbol{\Theta}$. Nevertheless, the ML estimate of $\boldsymbol{\Theta}$ can be obtained using the iterative *expectation-maximization* (EM) algorithm [Dempster et al., 1977]. EM has a broad applicability and it is used for several statistical models described in this thesis. At a maximum of the likelihood function eq. (3.6), the derivatives must be set to zero with respect to the parameters $\boldsymbol{\Theta}$. When writing down these conditions in the context of GMM, several useful quantities appear, such as the sufficient statistics given by eq. (3.2), eq. (3.3) and eq. (3.4). The only difference is that these statistics are then computed over the full training set $\mathbb{O}_{\text{train}}$.

The basic idea of EM is to begin with an initial GMM with parameters $\boldsymbol{\Theta}^0$. In practice, this initial GMM is obtained by initializing the means with a clustering algorithm, such as k-means [Steinhaus, 1957, MacQueen, 1967] or its variant k-means++ [Arthur and Vassilvitskii, 2007] that seeds the initial centers more carefully. EM alternates between the following two

steps: expectation and maximization. During the expectation step (E-step), the occupation probabilities of the training samples are evaluated using the current GMM parameters. During the maximization step (M-step), the parameters of the GMM are updated using the current occupation probabilities. These steps are repeated for a given number of iterations or until some convergence criterion is fulfilled. A more detailed procedure of the EM algorithm for GMM is given in alg. 1, and its application to the Miris dataset is shown on fig. 3.2.

Furthermore, this procedure can be refined when diagonal covariance matrices are used. The second-order statistics can then be stored in vectors instead of matrices, and multiplications are then performed element-wise (see eq. (3.4)).

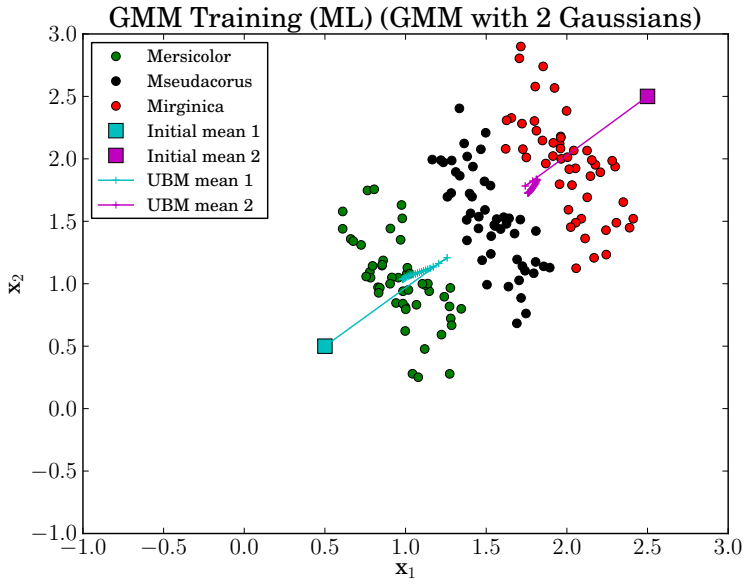


Figure 3.2 – MAXIMUM LIKELIHOOD ESTIMATION FOR A GMM ON THE MIRIS DATASET. *This figure shows the application of the maximum likelihood estimation procedure to train a GMM on the Miris dataset. The GMM consists of two Gaussian components, and its means are initialized by hand to* $(0.5, 0.5)$ *and* $(2.5, 2.5)$*, respectively. The evolution of the GMM means during EM with respect to the training iterations is displayed.*

When training a UBM on a large dataset, the computation of the statistics needed for the M-step might require a significant amount of time. These statistics involve three sums over all the samples of the training set (see eq. (3.2), eq. (3.3) and eq. (3.4)). To take advantage of multi-core processors or computer clusters, it is interesting to notice that the computation of these sums can be parallelized on a per-*chunk* basis, as follows. First, the training set is split according to a partition, each subset of the partition being a "chunk". During the E-step, the statistics for each subset of the partition are computed on separate cores or nodes. During the M-step, the statistics are first summed over the subsets, giving the statistics over the full training set, and the parameters of the models are then updated.

Algorithm 1 Maximum Likelihood Estimation for GMM using Expectation-Maximization

Initial GMM: $\boldsymbol{\Theta}^0 = \{\omega_c^0, \boldsymbol{\mu}_c^0, \boldsymbol{\Sigma}_c^0\}_{c=\{1,\ldots,C\}}$ *and* **a training set:** $\mathbb{O}_{\text{train}} = \{\boldsymbol{o}_1, \ldots, \boldsymbol{o}_{K_{\text{train}}}\}$
for it = 1 to maximum number of expectation-maximization iterations **do**
E-step: *Evaluate the responsibilities using the current GMM parameters*

$$\gamma_c^{\text{it}}(\boldsymbol{o}_k) = \frac{\omega_c^{\text{it}-1}\mathcal{N}\left[\boldsymbol{o}_k \mid \boldsymbol{\mu}_c^{\text{it}-1}, \boldsymbol{\Sigma}_c^{\text{it}-1}\right]}{\sum_{c=1}^{C}\omega_c^{\text{it}-1}\mathcal{N}\left[\boldsymbol{o}_k \mid \boldsymbol{\mu}_c^{\text{it}-1}, \boldsymbol{\Sigma}_c^{\text{it}-1}\right]} \qquad \texttt{\# eq. (3.5)}$$

and the sufficient statistics:
Count

$$n_c^{\text{it}}(\mathbb{O}_{\text{train}}) = \sum_{k=1}^{K_{\text{train}}} \gamma_c^{\text{it}}(\boldsymbol{o}_k) \qquad \texttt{\# eq. (3.2)}$$

First-order moment

$$\boldsymbol{f}_c^{\text{it}}(\mathbb{O}_{\text{train}}) = \sum_{k=1}^{K_{\text{train}}} \gamma_c^{\text{it}}(\boldsymbol{o}_k)\boldsymbol{o}_k \qquad \texttt{\# eq. (3.3)}$$

Second-order moment

$$\boldsymbol{S}_c^{\text{it}}(\mathbb{O}_{\text{train}}) = \sum_{k=1}^{K_{\text{train}}} \gamma_c^{\text{it}}(\boldsymbol{o}_k)\left(\boldsymbol{o}_k - \frac{\boldsymbol{f}_c^{\text{it}}(\mathbb{O}_{\text{train}})}{n_c^{\text{it}}(\mathbb{O}_{\text{train}})}\right)\left(\boldsymbol{o}_k - \frac{\boldsymbol{f}_c^{\text{it}}(\mathbb{O}_{\text{train}})}{n_c^{\text{it}}(\mathbb{O}_{\text{train}})}\right)^{\top} \qquad \texttt{\# eq. (3.4)}$$

M-step: *Weights:*

$$\omega_c^{\text{it}} = \frac{n_c^{\text{it}}(\mathbb{O}_{\text{train}})}{K_{\text{train}}}$$

Means:

$$\boldsymbol{\mu}_c^{\text{it}} = \frac{1}{n_c^{\text{it}}(\mathbb{O}_{\text{train}})}\boldsymbol{f}_c^{\text{it}}(\mathbb{O}_{\text{train}})$$

Variances:

$$\boldsymbol{\Sigma}_c^{\text{it}} = \frac{1}{n_c^{\text{it}}(\mathbb{O}_{\text{train}})}\boldsymbol{S}_c^{\text{it}}(\mathbb{O}_{\text{train}})$$

Evaluate the likelihood, check for convergence and stop the loop if the convergence criterion is satisfied
end for
return GMM ML estimate $\boldsymbol{\Theta}^{\text{ML}} = \{\omega_c^{\text{ML}}, \boldsymbol{\mu}_c^{\text{ML}}, \boldsymbol{\Sigma}_c^{\text{ML}}\}_{c=\{1,\ldots,C\}}$

Adapting a Class-specific Model

Once a UBM has been learned, a class-specific model is derived by adapting the parameters $\boldsymbol{\Theta}^{\text{UBM}} = \boldsymbol{\Theta}^{\text{ML}}$ of the UBM, which is seen as a prior distribution, to the enrollment samples for this class. The main motivation is to adapt the well-trained parameters of the UBM to the enrollment samples rather than learning all the parameters of the class-specific model from a possibly limited number of enrollment samples. In practice, a form of Bayesian adaptation called *maximum a posteriori* (MAP) estimation [Gauvain and Lee, 1994] is employed.

Similarly to the EM algorithm, this adaptation is a two step process. The sufficient statistics of

the GMM are first computed, which consists of the zeroth-, first- and second-order moments of the GMM. In the second step, these statistics are combined with the parameters of the UBM prior distribution, using an adaptation mixing coefficient.

Algorithm 2 Maximum A Posteriori Estimation for GMM

1: **Prior UBM**: $\boldsymbol{\Theta}^{\text{UBM}} = \{\omega_c^{\text{UBM}}, \boldsymbol{\mu}_c^{\text{UBM}}, \boldsymbol{\Sigma}_c^{\text{UBM}}\}_{c=\{1,\dots,C\}}$ *and* **enrollment samples:** $\mathbb{O}_{\text{enroll}} = \{\boldsymbol{o}_1, \dots, \boldsymbol{o}_{K_{\text{enroll}}}\}$

2: **Step 1:** *Evaluate the responsibilities using the UBM parameters*

$$\gamma_c(\boldsymbol{o}_k) = \frac{\omega_c^{\text{UBM}} \mathcal{N}\left[\boldsymbol{o}_k \mid \boldsymbol{\mu}_c^{\text{UBM}}, \boldsymbol{\Sigma}_c^{\text{UBM}}\right]}{\sum_{c=1}^{C} \omega_c^{\text{UBM}} \mathcal{N}\left[\boldsymbol{o}_k \mid \boldsymbol{\mu}_c^{\text{UBM}}, \boldsymbol{\Sigma}_c^{\text{UBM}}\right]} \qquad \texttt{\# eq. (3.5)}$$

and the sufficient statistics:
Count

$$n_c^{\text{MAP}}(\mathbb{O}_{\text{enroll}}) = \sum_{k=1}^{K_{\text{enroll}}} \gamma_c(\boldsymbol{o}_k) \qquad \texttt{\# eq. (3.2)}$$

First-order moment

$$\boldsymbol{f}_c^{\text{MAP}}(\mathbb{O}_{\text{enroll}}) = \sum_{k=1}^{K_{\text{enroll}}} \gamma_c(\boldsymbol{o}_k) \boldsymbol{o}_k \qquad \texttt{\# eq. (3.3)}$$

Second-order moment

$$\boldsymbol{S}_c^{\text{MAP}}(\mathbb{O}_{\text{enroll}}) = \sum_{k=1}^{K_{\text{enroll}}} \gamma_c^{\text{MAP}}(\boldsymbol{o}_k) \left(\boldsymbol{o}_k - \frac{\boldsymbol{f}_c^{\text{MAP}}(\mathbb{O}_{\text{enroll}})}{n_c^{\text{MAP}}(\mathbb{O}_{\text{enroll}})}\right)\left(\boldsymbol{o}_k - \frac{\boldsymbol{f}_c^{\text{MAP}}(\mathbb{O}_{\text{enroll}})}{n_c^{\text{MAP}}(\mathbb{O}_{\text{enroll}})}\right)^{\top} \qquad \texttt{\# eq. (3.4)}$$

3: **Step 2:** *Update the parameters of the model:*
Weights:

$$\omega_c^{\text{MAP}} = \left(\alpha_c^w \frac{n_c^{\text{MAP}}(\mathbb{O}_{\text{enroll}})}{K_{\text{enroll}}} + \left(1 - \alpha_c^w\right) \omega_c^{\text{UBM}}\right) \rho$$

4: *Means:*

$$\boldsymbol{\mu}_c^{\text{MAP}} = \alpha_c^m \frac{\boldsymbol{f}_c^{\text{MAP}}(\mathbb{O}_{\text{enroll}})}{n_c^{\text{MAP}}(\mathbb{O}_{\text{enroll}})} + \left(1 - \alpha_c^m\right) \boldsymbol{\mu}_c^{\text{UBM}}$$

5: *Variances:*

$$\boldsymbol{\Sigma}_c^{\text{MAP}} = \alpha_c^v \boldsymbol{S}_c^{\text{MAP}}(\mathbb{O}_{\text{enroll}}) + \left(1 - \alpha_c^v\right)\left(\boldsymbol{\Sigma}_c^{\text{UBM}} + \left(\boldsymbol{\mu}_c^{\text{UBM}}\right)^2 - \left(\boldsymbol{\mu}_c^{\text{MAP}}\right)^2\right)$$

6: **return** GMM MAP estimate $\boldsymbol{\Theta}^{\text{MAP}} = \{\omega_c^{\text{MAP}}, \boldsymbol{\mu}_c^{\text{MAP}}, \boldsymbol{\Sigma}_c^{\text{MAP}}\}_{c=\{1,\dots,C\}}$

This technique is described in alg. 2. The scale factor, ρ, is set such that the weights sum to unity. The adaptation coefficients $\{\alpha_c^w, \alpha_c^m, \alpha_c^v\}_{=\{1,\dots,C\}}$ control the balance between the prior UBM distribution and the new estimates, for the weights, the means and the variances, respectively. In [Reynolds, 1997, Reynolds et al., 2000], it is proposed to use a single data-dependent adaptation coefficient α_c for each GMM component. In particular, this means that the same adaptation factor $\alpha_c = \alpha_c^w = \alpha_c^m = \alpha_c^v$ is used for the weight, the mean and the

variance of a GMM component. This data-dependent adaptation coefficient is defined by:

$$\alpha_c = \frac{n_c^{\text{MAP}}}{n_c^{\text{MAP}} + \tau}, \tag{3.7}$$

where τ is a fixed parameter called the *relevance factor* and $n_c^{\text{MAP}}(\mathbb{O}_{\text{enroll}})$ the zeroth-order statistics of the enrollment samples (see eq. (3.2)). There are several advantages to use such a data-dependent adaptation coefficient. First, the components that are far from the enrollment samples (with a low probabilistic count value) are not adapted, as there is a high ambiguity due to the low amount of samples in this area of the feature space. In this case, the parameters of the old and well-trained UBM are emphasized. In contrast, components with high probabilistic counts are adapted more, according to the enrollment samples. The relevance factor τ is the parameter that allows to control the balance between these two effects.

Considering the Miris dataset, the impact of increasing the relevance factor is shown on fig. 3.3(a). In this case, the UBM learned on three classes (cf. fig. 3.2) is adapted to the enrollment samples of the class Metosa. The higher τ is, the less the UBM means are adapted. Similarly the impact of the number of enrollment samples for a fixed relevance factor is shown on fig. 3.3(b). The larger is the number of enrollment samples, the more the UBM means are adapted.

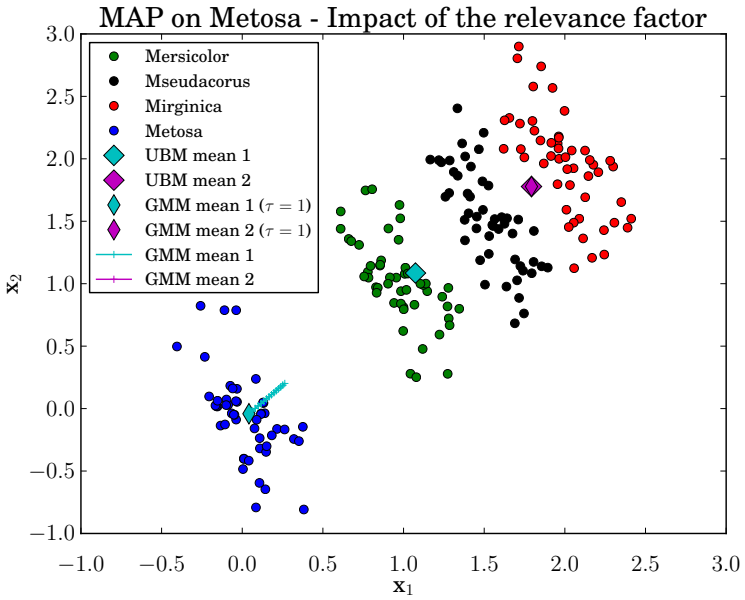
MAP adaptation allows to generate a class-specific model $\mathscr{S}_i$ with limited amounts of enrollment data. In practice it has been shown that mean-only adaptation, where only the means of the UBM are adapted, is effective for speaker [Reynolds et al., 2000] and face authentication [Lucey and Chen, 2004, Cardinaux et al., 2006]. We refer to this adaptation technique as *mean-only relevance MAP* in the following.

3.3.3 Supervector Notation

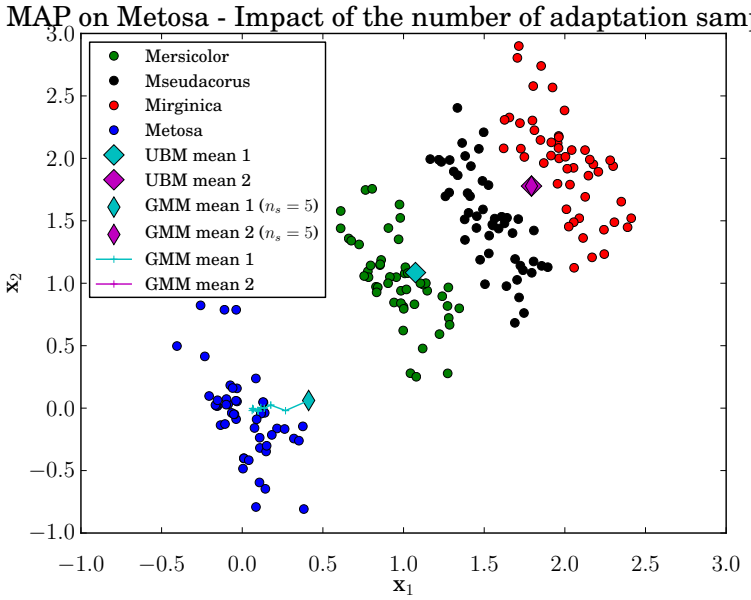
A compact way to write mean-only relevance MAP adaptation is by using GMM supervector notation. This notation also provides a compact representation to describe the session variability modeling techniques introduced in the remainder of this chapter. The GMM supervector notation consists of taking the parameters (weights, means and covariance matrices) of a GMM and creating a single vector or matrix to represent each of them.

An example of this would be that the means of the UBM can be concatenated to form a single mean supervector given by:

$$\boldsymbol{m} = \begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \\ \vdots \\ \boldsymbol{\mu}_C \end{bmatrix}. \tag{3.8}$$



(a) Impact of the relevance factor



(b) Impact of the number of enrollment samples

Figure 3.3 – MAXIMUM A POSTERIORI ESTIMATION FOR A GMM ON THE MIRIS DATASET. *This figure shows the adaptation of a UBM using maximum a posteriori. The UBM is adapted to the enrollment samples of the Metosa class. The impact of the relevance factor is shown on (a) (using all the* 50 *enrollment samples, while* τ *varies from* 1 *to* 16*). The impact of the number of enrollment samples (varying then number of enrollment samples from* 5 *to* 50 *for a fixed relevance factor* $\tau = 2$*) is shown on (b).*

Using this notation it is shown in [Vogt and Sridharan, 2008] that mean-only relevance MAP

adaptation equates to:

$$\boldsymbol{s}_i = \boldsymbol{m} + \boldsymbol{d}_i\,, \tag{3.9}$$

where the class-specific model is given by $\boldsymbol{s}_i$, which is a mean supervector consisting of two parts: (1) the prior world model, $\boldsymbol{m}$, and (2) a class-specific offset $\boldsymbol{d}_i$. When performing mean-only adaptation, the only GMM parameters that are different between the UBM and any class-specific model are the means. In the following, by abusing the notation, we, hence, refer to the UBM $\mathscr{M}$ by its mean supervector $\boldsymbol{m}$ and to the model $\mathscr{S}_i$ of class i by its mean supervector $\boldsymbol{s}_i$, respectively. In addition, the class-specific offset $\boldsymbol{d}_i$ is [Vogt and Sridharan, 2008]:

$$\boldsymbol{d}_i = \boldsymbol{D}\boldsymbol{z}_i\,, \tag{3.10}$$

where $\boldsymbol{D}$ is a diagonal matrix of size $(CD_{\boldsymbol{o}}, CD_{\boldsymbol{o}})$ [1] satisfying:

$$\boldsymbol{I} = \tau \boldsymbol{D}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{D}, \tag{3.11}$$

and $\boldsymbol{\Sigma}$ is a block diagonal matrix with block diagonal entries consisting of the covariance matrices $\boldsymbol{\Sigma}_c$ for each of the C components of the UBM:

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_2 & \mathbf{0} & \vdots \\ \vdots & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \boldsymbol{\Sigma}_C \end{bmatrix}. \tag{3.12}$$

The latent variable $\boldsymbol{z}_i$ is assumed to be normally distributed, $\boldsymbol{z}_i \sim \mathscr{N}(\mathbf{0}, \boldsymbol{I})$.

Creating a class-specific model is achieved by finding the MAP solution of $\boldsymbol{z}_i$ [Vogt and Sridharan, 2008]:

$$\boldsymbol{z}_i = (\tau \boldsymbol{I} + \boldsymbol{N}_i)^{-1} \boldsymbol{f}_{i|\boldsymbol{m}}. \tag{3.13}$$

The terms $\boldsymbol{N}_i$ and $\boldsymbol{f}_{i|\boldsymbol{m}}$ refer to the zeroth-order and mean centralized first-order statistics of the J_i enrollment samples of the i^{th} class. The mean centralized first-order statistic is:

$$\boldsymbol{f}_{i|\boldsymbol{m}} = \sum_{j=i}^{J_i} \boldsymbol{f}_{i,j|\boldsymbol{m}} \tag{3.14}$$

where the mean centralized first-order statistic for the j^{th} sample of class i is:

$$\boldsymbol{f}_{i,j|\boldsymbol{m}} = \boldsymbol{f}_{i,j} - \boldsymbol{N}_{i,j}\boldsymbol{m}, \tag{3.15}$$

1. We remind that C refers to the number of Gaussian components of the GMM, and $D_{\boldsymbol{o}}$ refers to the dimensionality of the feature vectors, and, therefore, $CD_{\boldsymbol{o}} = C \times D_{\boldsymbol{o}}$ refers to the dimensionality of a mean supervector.

$\boldsymbol{f}_{i,j} = \left[\boldsymbol{f}_{i,j;1}^\top, \boldsymbol{f}_{i,j;2}^\top, \ldots, \boldsymbol{f}_{i,j;C}^\top\right]^\top$, and $\boldsymbol{f}_{i,j;c}$ is the first-order statistic of component c for the j^{th} sample of class i given by eq. (3.3), when using the following notations:

$$n_{i,j;c} = n_c\left(\mathbb{O}_{i,j}\right), \quad \boldsymbol{f}_{i,j;c} = \boldsymbol{f}_c\left(\mathbb{O}_{i,j}\right) \quad \text{and} \quad \boldsymbol{S}_{i,j;c} = \boldsymbol{S}_c\left(\mathbb{O}_{i,j}\right). \tag{3.16}$$

The zeroth-order statistic of class i is:

$$\boldsymbol{N}_i = \sum_{j=1}^{J_i} \boldsymbol{N}_{i,j}, \tag{3.17}$$

where:

$$\boldsymbol{N}_{i,j} = \begin{bmatrix} \boldsymbol{N}_{i,j;1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{N}_{i,j;C} \end{bmatrix}, \text{ and } \boldsymbol{N}_{i,j;c} = \begin{bmatrix} n_{i,j;c} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & n_{i,j;c} \end{bmatrix}. \tag{3.18}$$

The term $n_{i,j;c}$ is the zeroth-order statistic of component c for the j^{th} sample of class i given by eq. (3.2) and $\boldsymbol{N}_{i,j;c}$ is of size $(D_{\boldsymbol{o}}, D_{\boldsymbol{o}})$.

3.3.4 Classification

At test time, given the model of a claimed class $\boldsymbol{s}_i$ and a set of feature vectors $\mathbb{O}_{\text{test}} = \{\boldsymbol{o}_1, \ldots, \boldsymbol{o}_{K_{\text{test}}}\}$ extracted from a test sample $\boldsymbol{\chi}_{\text{test}}$, the goal is to generate a score that quantifies whether or not the test features were generated by this model. This score might correspond to two different hypotheses, that are (H_0) "the features $\mathbb{O}_{\text{test}}$ were generated by the class-specific model $\boldsymbol{s}_i$" and (H_1) "the features were not generated by the class-specific model", respectively.

In a verification scenario (2-classes problem), a statistical test suitable to decide between these two hypotheses is a likelihood ratio test [Reynolds et al., 2000]:

$$\frac{P\left(\mathbb{O}_{\text{test}} \mid H_0\right)}{P\left(\mathbb{O}_{\text{test}} \mid H_1\right)} \begin{cases} \geq \theta & \text{accept } H_0 \\ < \theta & \text{reject } H_0 \end{cases} \tag{3.19}$$

where $P(\mathbb{O}_{\text{test}} \mid H_n), n = \{0, 1\}$ is the probability density function for the hypotheses H_n, considering the set of feature vectors $\mathbb{O}_{\text{test}}$ extracted from the test sample $\boldsymbol{\chi}_{\text{test}}$. The decision threshold θ is used to determine whether to accept or reject H_0.

The model associated with the hypothesis H_0 is well-defined and corresponds to the class-specific model $\boldsymbol{s}_i$, estimated with the enrollment samples of i. Therefore, the probability $P\left(\mathbb{O}_{\text{test}} \mid H_0\right)$ can be easily estimated as the likelihood of the test features $\mathbb{O}_{\text{test}}$ given the GMM class-specific model $\boldsymbol{s}_i$. In contrast, the model associated with the hypothesis H_1 is more ambiguous since it is supposed to describe all the possible alternatives to the hypothesized class i. A very popular choice is to use the UBM as the model associated with the hypothesis

H_1, and in this case, the probability density function value $P(\mathbb{O}_{\text{test}} \mid H_1)$ is estimated as the likelihood of the test features $\mathbb{O}_{\text{test}}$ given the UBM $\boldsymbol{m}$.

Log-likelihood ratio score

Even for a generic recognition scenario (possibly not a verification one), the likelihood ratio given by the left term in eq. (3.19) can be computed to obtain a biometric score. When using Gaussian distributions as in this work, it is often more convenient to consider the logarithm of this expression. This leads to a *log-likelihood ratio* (LLR) score:

$$h(\mathbb{O}_{\text{test}}, \boldsymbol{s}_i) = \sum_{k=1}^{K_{\text{test}}} \left(\ln\left(P(\boldsymbol{o}_k \mid \boldsymbol{s}_i)\right) - \ln\left(P(\boldsymbol{o}_k \mid \boldsymbol{m})\right)\right). \tag{3.20}$$

Linear scoring approximation

In [Glembek et al., 2009], a fast scoring technique known as *linear scoring* is proposed, that is used in this thesis. This is a first-order approximation of the log-likelihood ratio that is shown to be as accurate and up to two orders of magnitude more efficient to compute. Using the GMM supervector notation, linear scoring can be simply written:

$$h_{\text{linear}}(\mathbb{O}_{\text{test}}, \boldsymbol{s}_i) = (\boldsymbol{s}_i - \boldsymbol{m})^{\top} \boldsymbol{\Sigma}^{-1} \boldsymbol{f}_{\text{test}|\boldsymbol{m}}. \tag{3.21}$$

3.4 Inter-Session Variability Modeling and Joint Factor Analysis

Inter-session variability (ISV) modeling [Vogt and Sridharan, 2008] and *joint factor analysis* (JFA) [Kenny et al., 2007, Glembek, 2012] are two *session variability modeling* techniques that were successfully applied to speaker recognition.

ISV and JFA are applied in the context of a GMM-based system. In the case of mean-only relevance MAP adaptation, as described in sec. 3.3, there is no explicit modeling of session variability. The model consists of only two parts, the UBM $\boldsymbol{m}$ and the class-specific offset $\boldsymbol{d}_i$ as described by eq. (3.9). Ideally, the resulting class-specific model should be robust to any variations within the class's enrollment samples due to, for example, changes in illumination, expression or pose in the context of face recognition. However, this variation is not accounted for in eq. (3.9), and so this will likely lead to a suboptimal class-specific model, particularly in the case of limited enrollment data.

Session variability modeling proposes to model the variation between different sessions of the same class and compensate for this variation during enrollment as well as testing. This is achieved by excluding sources of session variation when generating a class-specific model as well as estimating and compensating for the different conditions (session variations) observed in test samples. This approach of session variability modeling is highly advantageous as it can

be used in conjunction with state-of-the-art normalization techniques to model the residual noise which will inevitably be left behind; no normalization technique is perfect and so there will always be some residual form of noise or session variation.

When we apply session variability modeling to face or speaker recognition, we consider that each sample corresponds to a different session. This seems intuitive because each sample can be captured under different conditions and/or different sensors. Following [Vogt and Sridharan, 2008], the particular conditions of a session are assumed to result in an additive offset to the class-specific model $\boldsymbol{s}_i$:

$$\boldsymbol{\mu}_{i,j} = \boldsymbol{s}_i + \boldsymbol{u}_{i,j}\,, \tag{3.22}$$

where $\boldsymbol{u}_{i,j}$ is the session-dependent offset for the j^{th} sample of class i, and $\boldsymbol{\mu}_{i,j}$ is the resulting mean supervector of the GMM that best represents the sample $\mathbb{O}_{i,j}$. The goal of enrollment using session variability modeling is to find the true session-independent class-specific model, $\boldsymbol{s}_i$, by jointly estimating it along with each $\boldsymbol{u}_{i,j}$.

Both ISV and JFA *explicitly model* session variability, however, JFA also explicitly models between-class variability. This difference, along with the algorithms used for estimation, training and classification, is discussed in more detail in the following sections.

3.4.1 Inter-Session Variability Modeling (ISV)

The ISV technique, proposed in [Vogt and Sridharan, 2008], assumes that within-class variation is contained in a linear subspace of the GMM mean supervector space. That is:

$$\boldsymbol{u}_{i,j} = \boldsymbol{U}\boldsymbol{x}_{i,j}\,, \tag{3.23}$$

where $\boldsymbol{U}$ is the low-dimensional subspace of size $(CD_{\boldsymbol{o}}, D_{\boldsymbol{U}})$ that contains within-class variation, and $\boldsymbol{x}_{i,j}$, of length $D_{\boldsymbol{U}}$, is the latent session variable, which is assumed to be normally distributed ($\boldsymbol{x}_{i,j} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$). As with relevance MAP adaptation, the class-dependent offset is set to $\boldsymbol{d}_i = \boldsymbol{D}\boldsymbol{z}_i$, as per eq. (3.10) and eq. (3.11).

To summarize, in this generative model each sample is assumed to have been generated by a GMM mean supervector:

$$\boldsymbol{\mu}_{i,j} = \boldsymbol{m} + \boldsymbol{U}\boldsymbol{x}_{i,j} + \boldsymbol{D}\boldsymbol{z}_i\,. \tag{3.24}$$

At enrollment time, the model for class i is obtained by estimating the latent variables, $\boldsymbol{z}_i$ and $\boldsymbol{x}_{i,j}$, using the procedure described in sec. 3.4.3. The estimated effect of session variability in each sample eq. (3.23) is then excluded from the class-specific model. This means that for ISV the resulting class-specific model is:

$$\boldsymbol{s}_i^{\text{ISV}} = \boldsymbol{m} + \boldsymbol{D}\boldsymbol{z}_i\,. \tag{3.25}$$

This should not be confused with relevance MAP adaptation eq. (3.9) because to obtain $\boldsymbol{s}_i^{\text{ISV}}$ the latent class variable $\boldsymbol{z}_i$ is estimated along with the latent session variable $\boldsymbol{x}_{i,j}$ in the generative framework defined by eq. (3.24), thus suppressing the effects of session variability. Therefore, the class-specific model for ISV, $\boldsymbol{s}_i^{\text{ISV}}$, is quite different to the one for relevance MAP adaptation, $\boldsymbol{s}_i$. Scoring for ISV is discussed in sec. 3.4.5.

3.4.2 Joint Factor Analysis (JFA)

JFA [Kenny et al., 2007] can be seen as an extension of ISV. Specifically, for JFA the class-dependent offset is defined as:

$$\boldsymbol{d}_i = \boldsymbol{V}\boldsymbol{y}_i + \hat{\boldsymbol{D}}\boldsymbol{z}_i\,, \tag{3.26}$$

in contrast to relevance MAP adaptation and ISV, where $\boldsymbol{d}_i = \boldsymbol{D}\boldsymbol{z}_i$. For JFA, $\boldsymbol{V}$ is a low rank rectangular matrix of size $(CD_{\boldsymbol{o}}, D_{\boldsymbol{V}})$, $\boldsymbol{y}_i$ is the latent class variable of size $D_{\boldsymbol{V}}$, which is assumed to be normally distributed ($\boldsymbol{y}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$), and $\boldsymbol{d}_i$ is, thus, distributed with covariance matrix $\hat{\boldsymbol{D}}^2 + \boldsymbol{V}\boldsymbol{V}^\top$. The assumption of this model is that most between-class variability is contained within a low-dimensional subspace $\boldsymbol{V}$, which is in fact the assumption of the well-known eigenvoice modeling technique [Thyes et al., 2000]. One of the motivations for using JFA is to improve enrollment of a class with limited data by allowing a class-specific model to be approximately represented by only the small number of factors in the latent class variable $\boldsymbol{y}_i$.

To summarize, in contrast to ISV eq. (3.24), for JFA each sample is modeled by:

$$\boldsymbol{\mu}_{i,j} = \boldsymbol{m} + \boldsymbol{V}\boldsymbol{y}_i + \boldsymbol{U}\boldsymbol{x}_{i,j} + \hat{\boldsymbol{D}}\boldsymbol{z}_i\,. \tag{3.27}$$

In this case, both $\boldsymbol{V}$ and $\hat{\boldsymbol{D}}$ are learned from training data, in addition to $\boldsymbol{U}$, using maximum likelihood (see sec. 3.4.4). At enrollment time, the model for class i is obtained by estimating the latent variables $\boldsymbol{x}_{i,j}$, $\boldsymbol{y}_i$ and $\boldsymbol{z}_i$ (see sec. 3.4.3). As with ISV, we then suppress the effects of session variability by removing the term $\boldsymbol{U}\boldsymbol{x}_{i,j}$. For JFA, the resulting class-specific model is:

$$\boldsymbol{s}_i^{\text{JFA}} = \boldsymbol{m} + \boldsymbol{V}\boldsymbol{y}_i + \hat{\boldsymbol{D}}\boldsymbol{z}_i\,. \tag{3.28}$$

Scoring is similar to ISV and is discussed in sec. 3.4.5.

In order to use the ISV and JFA frameworks described above we need to be able to: (1) estimate the latent variables, $\boldsymbol{x}_{i,j}$, $\boldsymbol{y}_i$ and $\boldsymbol{z}_i$, and (2) train the subspaces $\boldsymbol{U}$, $\boldsymbol{V}$ and $\hat{\boldsymbol{D}}$. As described below, to solve these two problems we follow the approach of [Vogt and Sridharan, 2008] for ISV, which is, in short, MAP estimation to solve problem (1) and maximum likelihood (ML) estimation to solve problem (2).

3.4.3 Estimation of Latent Variables

The approach to estimate the latent variables is identical for ISV and JFA. The aim is to jointly estimate the latent variables, $\boldsymbol{x}_{i,j}$, $\boldsymbol{y}_i$ and $\boldsymbol{z}_i$, using MAP estimation. In case of ISV, only $\boldsymbol{x}_{i,j}$ and $\boldsymbol{z}_i$ need to be estimated. These latent variables are of size $D_{\boldsymbol{U}}$, $D_{\boldsymbol{V}}$ and $CD_{\boldsymbol{o}}$ for $\boldsymbol{x}_{i,j}$, $\boldsymbol{y}_i$ and $\boldsymbol{z}_i$ respectively.

Central to this process is to note that the latent class variables ($\boldsymbol{z}_i$ for ISV and additionally $\boldsymbol{y}_i$ for JFA) are tied together for all of the J_i enrollment samples of class i. This means that all J_i enrollment samples share the same latent class variables but have different latent session variables $\left[\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2}, \ldots, \boldsymbol{x}_{i,J_i}\right]$. We can represent this in a convenient way by:

$$\begin{bmatrix} \boldsymbol{\mu}_{i,1} \\ \vdots \\ \boldsymbol{\mu}_{i,J_i} \end{bmatrix} = \begin{bmatrix} \boldsymbol{m} \\ \vdots \\ \boldsymbol{m} \end{bmatrix} + \tilde{\boldsymbol{A}}\tilde{\boldsymbol{\lambda}}_i, \tag{3.29}$$

where we have concatenated the latent variables of the class i to form, for JFA:

$$\tilde{\boldsymbol{\lambda}}_i = \left[\boldsymbol{z}_i^\top, \boldsymbol{y}_i^\top, \boldsymbol{x}_{i,1}^\top, \boldsymbol{x}_{i,2}^\top, \ldots, \boldsymbol{x}_{i,J_i}^\top\right]^\top, \tag{3.30}$$

and $\tilde{\boldsymbol{A}}$ is a composite matrix with J_i entries of $\boldsymbol{U}$, $\hat{\boldsymbol{D}}$ and $\boldsymbol{V}$, with $\boldsymbol{U}$ being repeated in a block diagonal fashion such that:

$$\tilde{\boldsymbol{A}} = \begin{bmatrix} \hat{\boldsymbol{D}} & \boldsymbol{V} & \boldsymbol{U} & \boldsymbol{0} & \boldsymbol{0} \\ \vdots & \vdots & \boldsymbol{0} & \ddots & \boldsymbol{0} \\ \hat{\boldsymbol{D}} & \boldsymbol{V} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{U} \end{bmatrix}, \tag{3.31}$$

the tilde symbol ˜ indicating that the dimension of a variable (or a matrix) depends on the number of samples J_i for the class i. We use a similar approach to represent ISV by simply removing the columns referring to $\boldsymbol{V}$ in eq. (3.31) and its associated latent variable $\boldsymbol{y}_i$ in eq. (3.30); also note that for ISV $\boldsymbol{D}$ is defined by eq. (3.11) rather than $\hat{\boldsymbol{D}}$, which is learned from data. For clarity we note that the dimensions of the matrices involved are the following: $\hat{\boldsymbol{D}}$ is assumed diagonal and is of dimension $(CD_{\boldsymbol{o}}, CD_{\boldsymbol{o}})$, $\boldsymbol{V}$ is rectangular of dimension $(CD_{\boldsymbol{o}}, D_{\boldsymbol{V}})$ and $\boldsymbol{U}$ is rectangular of dimension $(CD_{\boldsymbol{o}}, D_{\boldsymbol{U}})$. Thus the matrix $\tilde{\boldsymbol{A}}$ is of dimension $(J_i \times CD_{\boldsymbol{o}}, CD_{\boldsymbol{o}} + D_{\boldsymbol{V}} + J_i \times D_{\boldsymbol{U}})$ and $\tilde{\boldsymbol{\lambda}}_i$ is a vector of size $CD_{\boldsymbol{o}} + D_{\boldsymbol{V}} + J_i \times D_{\boldsymbol{U}}$.

Using the above formulation, class enrollment reduces to finding the MAP estimate of $\tilde{\boldsymbol{\lambda}}_i$:

$$\begin{aligned} \tilde{\boldsymbol{\lambda}}_i^* &= \underset{\tilde{\boldsymbol{\lambda}}_i}{\operatorname{argmax}}\ P(\tilde{\boldsymbol{\lambda}}_i \mid \mathbb{O}_{i,1}, \mathbb{O}_{i,2}, \ldots, \mathbb{O}_{i,J_i}), \\ &= \underset{\tilde{\boldsymbol{\lambda}}_i}{\operatorname{argmax}}\ P(\boldsymbol{z}_i) P(\boldsymbol{y}_i) \prod_{j=1}^{J_i} P(\mathbb{O}_{i,j} \mid \boldsymbol{x}_{i,j}, \boldsymbol{y}_i, \boldsymbol{z}_i) P(\boldsymbol{x}_{i,j}), \end{aligned} \tag{3.32}$$

where $\boldsymbol{y}_i$ is omitted in the case of ISV. Solving this leads to the solution [Kenny et al., 2008]:

$$\tilde{\boldsymbol{\lambda}}_i^* = \mathbb{E}\left[\tilde{\boldsymbol{\lambda}}_i\right] = \left(\boldsymbol{I} + \tilde{\boldsymbol{A}}^\top \tilde{\boldsymbol{\Sigma}}^{-1} \tilde{\boldsymbol{N}}_i \tilde{\boldsymbol{A}}\right)^{-1} \tilde{\boldsymbol{A}}^\top \tilde{\boldsymbol{\Sigma}}^{-1} \left(\sum_{i=1}^{J_i} \boldsymbol{f}_{i,j|\boldsymbol{m}}\right), \tag{3.33}$$

where:

$$\tilde{\boldsymbol{\Sigma}} = \begin{bmatrix} \boldsymbol{\Sigma} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \ddots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{\Sigma} \end{bmatrix} \quad \text{and} \quad \tilde{\boldsymbol{N}}_i = \begin{bmatrix} \boldsymbol{N}_{i,1} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \ddots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{N}_{i,J_i} \end{bmatrix}. \tag{3.34}$$

To solve eq. (3.33) we use a Gauss-Seidel approximation method inspired from [Vogt and Sridharan, 2008]. This approximation method was proposed for ISV and is necessary because $\tilde{\boldsymbol{A}}$ grows quadratically with respect to the number of samples J_i and so inverting the matrix $\left(\boldsymbol{I} + \tilde{\boldsymbol{A}}^\top \tilde{\boldsymbol{\Sigma}}^{-1} \tilde{\boldsymbol{N}}_i \tilde{\boldsymbol{A}}\right)$ quickly becomes intractable, even if we try to exploit its structure (see Section 3.3.2 of [Vogt and Sridharan, 2008] for more details).

Algorithm 3 Latent Variables Estimation of Identity/Class i for ISV/JFA

1: $\boldsymbol{y}_i = \boldsymbol{0}$, $\boldsymbol{z}_i = \boldsymbol{0}$ and $\boldsymbol{x}_{i,j} = \boldsymbol{0}$; $j = 1, \ldots, J_i$
2: Estimate $\boldsymbol{N}_{i,j}$ and $\boldsymbol{f}_{i,j}$; $j = 1, \ldots, J_i$ `# eq. (3.18) and eq. (3.3)`
3: $\boldsymbol{N}_i = \sum_{j=1}^{J_i} \boldsymbol{N}_{i,j}$ `# eq. (3.17)`
4: $\boldsymbol{f}_{i|\boldsymbol{m}} = \sum_{j=1}^{J_i} \boldsymbol{f}_{i,j|\boldsymbol{m}}$ `# eq. (3.15)`
5: **for** $i\,t = 1$ to Number of Gauss-Seidel iterations **do**
6: **Estimate** $\mathbb{E}\left[\boldsymbol{y}_i\right]$ `# eq. (3.36)`
7: **for** $j = 1$ to J_i **do**
8: **Estimate** $\mathbb{E}\left[\boldsymbol{x}_{i,j}\right]$ `# eq. (3.35)`
9: **end for**
10: **Estimate** $\mathbb{E}\left[\boldsymbol{z}_i\right]$ `# eq. (3.37)`
11: **end for**
12: **return** $\mathbb{E}\left[\boldsymbol{y}_i\right], \mathbb{E}\left[\boldsymbol{z}_i\right], \left[\mathbb{E}\left[\boldsymbol{x}_{i,1}\right], \ldots, \mathbb{E}\left[\boldsymbol{x}_{i,J_i}\right]\right]$

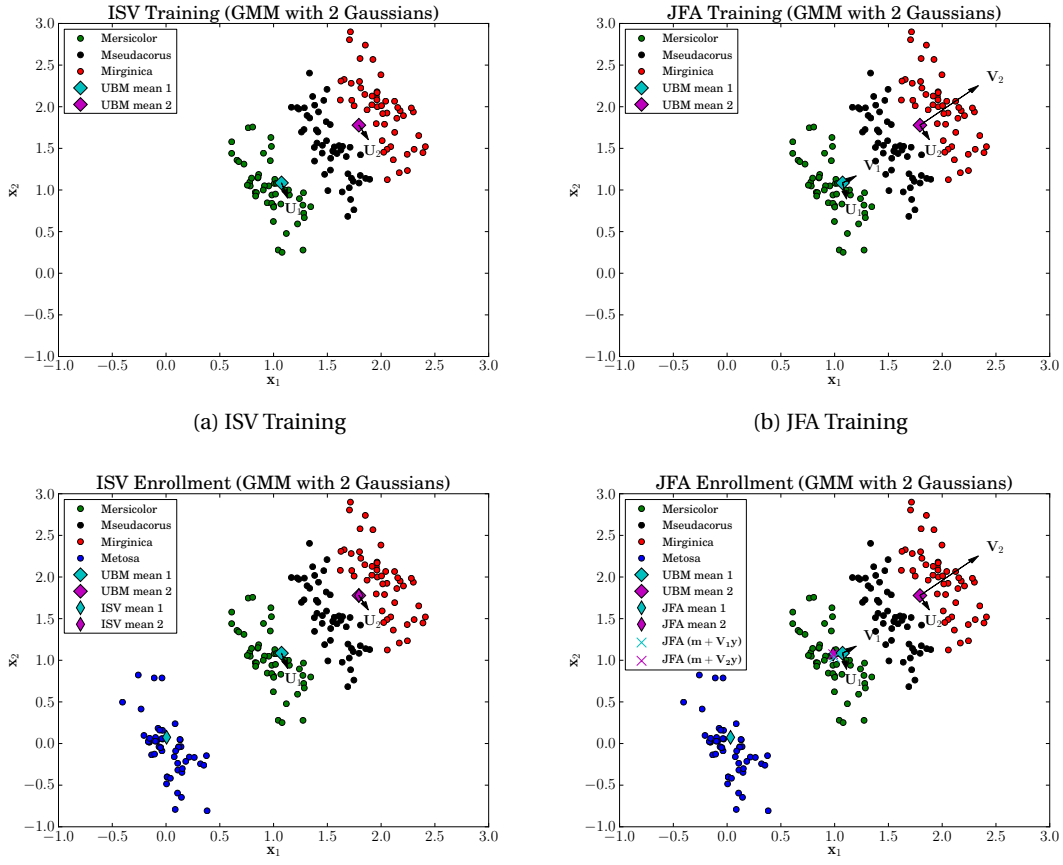
The Gauss-Seidel algorithm, alg. 3, iteratively estimates each latent variable. It does this by factorizing the concatenated latent variable $\tilde{\boldsymbol{\lambda}}_i$ into its respective latent variables $\boldsymbol{z}_i$, $\boldsymbol{y}_i$, $\boldsymbol{x}_{i,1}$ through to $\boldsymbol{x}_{i,J_i}$, this takes advantage of the known structure for these latent variables. Each factorized latent variable is then estimated using the most recent estimate of all of the other latent variables. In this case, we no longer jointly estimate each latent variable but estimate a latent variable by considering all of the others to be fixed (or known). This simplifies the estimation steps as we now only need to solve for one latent variable; a more detailed description and motivation for this approach is given in Section 3.4 of [Vogt and Sridharan, 2008]. We initialize this algorithm by setting all of the latent variables to $\boldsymbol{0}$, as they are assumed to be $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. For the case of ISV we omit the step of estimating $\mathbb{E}\left[\boldsymbol{y}_i\right]$ (line 6 of alg. 3) and $\boldsymbol{y}_i$ is effectively set to $\boldsymbol{0}$ in eq. (3.35) and eq. (3.37). The MAP estimation of each latent variable is:

$$\mathbb{E}\left[\boldsymbol{x}_{i,j}\right] = \left(\boldsymbol{I} + \boldsymbol{U}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{N}_{i,j} \boldsymbol{U}\right)^{-1} \boldsymbol{U}^\top \boldsymbol{\Sigma}^{-1} \left[\boldsymbol{f}_{i,j|m} - \boldsymbol{N}_{i,j}(\boldsymbol{V}\boldsymbol{y}_i + \boldsymbol{D}\boldsymbol{z}_i)\right], \tag{3.35}$$

$$\mathbb{E}\left[\boldsymbol{y}_i\right] = \left(\boldsymbol{I} + \boldsymbol{V}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{N}_i \boldsymbol{V}\right)^{-1} \boldsymbol{V}^\top \boldsymbol{\Sigma}^{-1} \left[\boldsymbol{f}_{i|m} - \boldsymbol{N}_i \boldsymbol{D} \boldsymbol{z}_i - \sum_{j=1}^{J_i} \boldsymbol{N}_{i,j} \boldsymbol{U} \boldsymbol{x}_{i,j}\right], \tag{3.36}$$

$$\mathbb{E}\left[\boldsymbol{z}_i\right] = \left(\boldsymbol{I} + \boldsymbol{D}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{N}_i \boldsymbol{D}\right)^{-1} \boldsymbol{D}^\top \boldsymbol{\Sigma}^{-1} \left[\boldsymbol{f}_{i|m} - \boldsymbol{N}_i \boldsymbol{V} \boldsymbol{y}_i - \sum_{j=1}^{J_i} \boldsymbol{N}_{i,j} \boldsymbol{U} \boldsymbol{x}_{i,j}\right]. \tag{3.37}$$

An illustration of the enrollment procedure applied to the Miris dataset is shown on fig. 3.4(c) and fig. 3.4(d) for ISV and JFA, respectively.



(a) ISV Training

(b) JFA Training

(c) ISV Enrollment

(d) JFA Enrollment

Figure 3.4 – TRAINING AND ENROLLMENT OF ISV AND JFA ON THE MIRIS DATASET. *This figure shows the application of the training ((a) and (b)) and enrollment ((c) and (d)) procedures of the ISV and JFA techniques on the Miris synthetic dataset. The ranks of the subspace $\boldsymbol{U}$ and $\boldsymbol{V}$ are set to* 1*, and these subspaces are shown using black arrows. For ISV, one of the GMM mean is not adapted (see (c) on the right)*

3.4.4 Estimation of Subspaces

To learn the subspaces $\hat{\boldsymbol{D}}$, $\boldsymbol{V}$ and $\boldsymbol{U}$ we use an expectation-maximization (EM) algorithm similar to that described in Section 5.2 of [Vogt and Sridharan, 2008] for ISV. This algorithm consists of an expectation step where MAP estimates of the latent variables are made (see sec. 3.4.3) and a maximization step where the parameters are updated using ML. For JFA we learn $\boldsymbol{V}$, $\boldsymbol{U}$ and $\hat{\boldsymbol{D}}$, while for ISV we only learn $\boldsymbol{U}$. It can be shown that the updates for the parameters ($\boldsymbol{V}$, $\boldsymbol{U}$ and $\hat{\boldsymbol{D}}$) are obtained by solving the following systems of equations:

$$\begin{aligned}&\boldsymbol{V}_c \left(\sum_{i=1}^{I} \boldsymbol{N}_{i;c} \mathbb{E}\left[\boldsymbol{y}_i \boldsymbol{y}_i^\top\right] \right) = \\ &\quad \sum_{i=1}^{I} \left(\sum_{j=1}^{J_i} \left(\boldsymbol{f}_{i,j;c|\boldsymbol{m}} - \boldsymbol{N}_{i,j;c} (\hat{\boldsymbol{D}}_c \boldsymbol{z}_i + \boldsymbol{U}_c \boldsymbol{x}_{i,j}) \right) \right) \mathbb{E}\left[\boldsymbol{y}_i\right]^\top ,\end{aligned} \tag{3.38}$$

$$\begin{aligned}&\boldsymbol{U}_c \left(\sum_{i=1}^{I} \sum_{j=1}^{J_i} \boldsymbol{N}_{i,j;c} \mathbb{E}\left[\boldsymbol{x}_{i,j} \boldsymbol{x}_{i,j}^\top\right] \right) = \\ &\quad \sum_{i=1}^{I} \left(\sum_{j=1}^{J_i} \left(\boldsymbol{f}_{i,j;c|\boldsymbol{m}} - \boldsymbol{N}_{i,j;c} (\boldsymbol{V}_c \boldsymbol{y}_i + \hat{\boldsymbol{D}}_c \boldsymbol{z}_i) \right) \mathbb{E}\left[\boldsymbol{x}_{i,j}\right]^\top \right) ,\end{aligned} \tag{3.39}$$

$$\begin{aligned}&\hat{\boldsymbol{D}}_c \left(\sum_{i=1}^{I} \boldsymbol{N}_{i;c} \mathbb{E}\left[\boldsymbol{z}_i \boldsymbol{z}_i^\top\right] \right) = \\ &\quad \sum_{i=1}^{I} \left(\sum_{j=1}^{J_i} \left(\boldsymbol{f}_{i,j;c|\boldsymbol{m}} - \boldsymbol{N}_{i,j;c} (\boldsymbol{V}_c \boldsymbol{y}_i + \boldsymbol{U}_c \boldsymbol{x}_{i,j}) \right) \right) \mathbb{E}\left[\boldsymbol{z}_i\right]^\top ,\end{aligned} \tag{3.40}$$

where $\boldsymbol{f}_{i,j;c|\boldsymbol{m}}$ is the mean normalized first-order statistics for component c similar to eq. (3.15):

$$\boldsymbol{V} = \begin{bmatrix} \boldsymbol{V}_1 \\ \vdots \\ \boldsymbol{V}_C \end{bmatrix}, \ \boldsymbol{U} = \begin{bmatrix} \boldsymbol{U}_1 \\ \vdots \\ \boldsymbol{U}_C \end{bmatrix}, \text{ and } \hat{\boldsymbol{D}} = \begin{bmatrix} \hat{\boldsymbol{D}}_1 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \ddots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \hat{\boldsymbol{D}}_C \end{bmatrix}. \tag{3.41}$$

In this formulation, the expected value of the square of the latent variables is given by:

$$\mathbb{E}\left[\boldsymbol{y}_i \boldsymbol{y}_i^\top\right] = \left(\boldsymbol{I} + \boldsymbol{V}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{N}_i \boldsymbol{V}\right)^{-1} + \mathbb{E}\left[\boldsymbol{y}_i\right] \mathbb{E}\left[\boldsymbol{y}_i\right]^\top , \tag{3.42}$$

$$\mathbb{E}\left[\boldsymbol{x}_{i,j} \boldsymbol{x}_{i,j}^\top\right] = \left(\boldsymbol{I} + \boldsymbol{U}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{N}_{i,j} \boldsymbol{U}\right)^{-1} + \mathbb{E}\left[\boldsymbol{x}_{i,j}\right] \mathbb{E}\left[\boldsymbol{x}_{i,j}\right]^\top , \tag{3.43}$$

$$\mathbb{E}\left[\boldsymbol{z}_i \boldsymbol{z}_i^\top\right] = \left(\boldsymbol{I} + \hat{\boldsymbol{D}}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{N}_i \hat{\boldsymbol{D}}\right)^{-1} + \mathbb{E}\left[\boldsymbol{z}_i\right] \mathbb{E}\left[\boldsymbol{z}_i\right]^\top . \tag{3.44}$$

Note that for ISV we need to substitute the learned matrix $\hat{\boldsymbol{D}}$ for the predefined matrix $\boldsymbol{D}$ in eq. (3.39), eq. (3.41) and eq. (3.44).

We use the training procedure described in [Burget et al., 2008], which is similar to the one proposed in [Kenny et al., 2008]. For JFA, this procedure first learns $\boldsymbol{V}$, then $\boldsymbol{U}$ and finally $\hat{\boldsymbol{D}}$. Each parameter is learned using an EM algorithm, where the E-Step is the same as the one described in sec. 3.4.3 and the M-Step is given by the equations above; the latent variables associated to those matrices which have yet to be learned are set to $\mathbf{0}$.

An illustration of this training procedure applied to the Miris dataset is shown on fig. 3.4(a) and fig. 3.4(b) for ISV and JFA, respectively.

Looking at eq. (3.39), eq. (3.38) and eq. (3.40), it can be noticed that only sums over all the classes or over all the samples of the training step are required for the M-step. These sums are computed during the E-step, and memory requirements can hence be reduced by storing only these sums instead of the expected values for each class or sample, separately.
Similar to the ML procedure for GMM, it is possible to split the computation of these sums when processing a large training set. This allows the E-step to be parallelized on several cores or nodes, each computation unit processing a subset of a partition of the training set.

Algorithm 4 Training Procedure for ISV using Expectation-Maximization

$\boldsymbol{z}_i = \mathbf{0}$ and $\boldsymbol{x}_{i,j} = \mathbf{0}$; $i = 1,\ldots,I,\ j = 1,\ldots,J_i$
Compute $\boldsymbol{N}_{i,j}$ and $\boldsymbol{f}_{i,j|\boldsymbol{m}}$; $i = 1,\ldots,I,\ j = 1,\ldots,J_i$ `# eq. (3.18) and eq. (3.15)`
$\boldsymbol{N}_i = \sum_{j=1}^{J_i} \boldsymbol{N}_{i,j}$ `# eq. (3.17)`
$\boldsymbol{f}_{i|\boldsymbol{m}} = \sum_{j=1}^{J_i} \boldsymbol{f}_{i,j|\boldsymbol{m}}$ `# eq. (3.15)`
Compute $\boldsymbol{D}$ `# eq. (3.11)`
Initialize $\boldsymbol{U}$ randomly
for it = 1 to maximum number of expectation-maximization iterations **do**
  E-step: ***Estimate*** $\mathbb{E}\left[\boldsymbol{x}_{i,j}\right]$ *# eq. (3.35)*
    Estimate $\mathbb{E}\left[\boldsymbol{z}_i\right]$ *# eq. (3.37)*
  M-step:
  for $c = 1$ to C **do**
    Update $\boldsymbol{U}_c$ *# eq. (3.39)*
  end for
end for
return ISV subspaces $[\boldsymbol{U}, \boldsymbol{D}]$

3.4.5 Classification

Classification for ISV and JFA relies on an LLR score similar to eq. (3.20). The key difference compared to the GMM baseline approach lies in the compensation for the unwanted session

Algorithm 5 Training Procedure for JFA using Expectation-Maximization

$\boldsymbol{y}_i = \mathbf{0}$, $\boldsymbol{z}_i = \mathbf{0}$ and $\boldsymbol{x}_{i,j} = \mathbf{0}$; $i = 1,\ldots,I$, $j = 1,\ldots,J_i$
Compute $\boldsymbol{N}_{i,j}$ and $\boldsymbol{f}_{i,j|\boldsymbol{m}}$; $i = 1,\ldots,I$, $j = 1,\ldots,J_i$ `# eq. (3.18) and eq. (3.15)`
$\boldsymbol{N}_i = \sum_{j=1}^{J_i} \boldsymbol{N}_{i,j}$ `# eq. (3.17)`
$\boldsymbol{f}_{i|\boldsymbol{m}} = \sum_{j=1}^{J_i} \boldsymbol{f}_{i,j|\boldsymbol{m}}$ `# eq. (3.15)`
Initialize $\boldsymbol{U}$, $\boldsymbol{V}$ and $\hat{\boldsymbol{D}}$ randomly
for it = 1 to maximum number of expectation-maximization iterations **do**
    E-step: ***Estimate*** $\mathbb{E}\left[\boldsymbol{y}_i\right]$ `# eq. (3.36)`
    M-step:
    for $c = 1$ to C **do**
        Update $\boldsymbol{V}_c$ `# eq. (3.38)`
    end for
end for
Estimate $\mathbb{E}\left[\boldsymbol{y}_i\right]$ `# eq. (3.36)`
for it = 1 to maximum number of expectation-maximization iterations **do**
    E-step: ***Estimate*** $\mathbb{E}\left[\boldsymbol{x}_{i,j}\right]$ `# eq. (3.35)`
    M-step:
    for $c = 1$ to C **do**
        Update $\boldsymbol{U}_c$ `# eq. (3.39)`
    end for
end for
Estimate $\mathbb{E}\left[\boldsymbol{x}_{i,j}\right]$ `# eq. (3.35)`
for it = 1 to maximum number of expectation-maximization iterations **do**
    E-step: ***Estimate*** $\mathbb{E}\left[\boldsymbol{z}_i\right]$ `# eq. (3.37)`
    M-step:
    for $c = 1$ to C **do**
        Update $\hat{\boldsymbol{D}}_c$ `# eq. (3.40)`
    end for
end for
return JFA subspaces $\left[\boldsymbol{U}, \boldsymbol{V}, \hat{\boldsymbol{D}}\right]$

variation in the test samples. The previous subsections describe how this unwanted session variation can be excluded during enrollment of the class-specific models for ISV and JFA. In this section, we discuss how to incorporate an estimate of the session variation in a test sample during LLR scoring.

A method to compensate for the effects of session variation in a set of observations $\mathbb{O}_{\text{test}} = \{\boldsymbol{o}_1,\ldots,\boldsymbol{o}_{K_{\text{test}}}\}$ extracted from a test sample $\boldsymbol{\chi}_{\text{test}}$ is proposed in [Vogt et al., 2005, Vogt and Sridharan, 2008]. Given a model for the class i without session variability effects ($\boldsymbol{s}_i^{\text{ISV}}$ and $\boldsymbol{s}_i^{\text{JFA}}$ for ISV and JFA, respectively), we estimate the latent session variable $\boldsymbol{x}_{i,\text{test}}$ for sample $\mathbb{O}_{\text{test}}$. Using this estimated latent session variable, we apply the corresponding offset to the i^{th} class-specific model (thus, $\boldsymbol{s}_i^{\text{ISV}} + \boldsymbol{U}\boldsymbol{x}_{i,\text{test}}$ and $\boldsymbol{s}_i^{\text{JFA}} + \boldsymbol{U}\boldsymbol{x}_{i,\text{test}}$ for ISV and JFA, respectively). This explicitly compensates for the estimated noise in $\mathbb{O}_{\text{test}}$ because the likelihood that the observed sample was produced by the claimed class i, *in the estimated intersession variability conditions*, is evaluated. Extending this to the case of the UBM, this formally results in the

following LLR:

$$h\left(\mathbb{O}_{\text{test}}, \boldsymbol{s}_i^*\right) = \sum_{k=1}^{K_{\text{test}}} \left(\ln\left(P\left(\boldsymbol{o}_k \mid \boldsymbol{s}_i^* + \boldsymbol{U}\boldsymbol{x}_{i,\text{test}}\right)\right) - \ln\left(P\left(\boldsymbol{o}_k \mid \boldsymbol{m} + \boldsymbol{U}\boldsymbol{x}_{\text{UBM,test}}\right)\right)\right), \tag{3.45}$$

where $\boldsymbol{s}_i^*$ is used to indicate either $\boldsymbol{s}_i^{\text{ISV}}$ or $\boldsymbol{s}_i^{\text{JFA}}$.

In [Vair et al., 2007], a simplification has been proposed, which assumes that the latent session variable $\boldsymbol{x}_{i,\text{test}}$ for sample $\mathbb{O}_{\text{test}}$ for each class i can be approximated using the latent session variable $\boldsymbol{x}_{\text{UBM,test}}$ estimated for the UBM. This assumption is referred to as the LPT (Loquendo Politechnico di Torino, [Vair et al., 2007]) assumption in [Glembek et al., 2009]. By doing this, for each test sample $\mathbb{O}_{\text{test}}$, the computation of only one latent session offset $\boldsymbol{U}\boldsymbol{x}_{\text{UBM,test}}$ is required, which is particularly useful when this sample is compared against several class-specific models. The LLR eq. (3.45) can hence be written:

$$h\left(\mathbb{O}_{\text{test}}, \boldsymbol{s}_i^*\right) = \sum_{k=1}^{K_{\text{test}}} \left(\ln\left(P\left(\boldsymbol{o}_k \mid \boldsymbol{s}_i^* + \boldsymbol{U}\boldsymbol{x}_{\text{UBM,test}}\right)\right) - \ln\left(P\left(\boldsymbol{o}_k \mid \boldsymbol{m} + \boldsymbol{U}\boldsymbol{x}_{\text{UBM,test}}\right)\right)\right). \tag{3.46}$$

If the linear scoring simplification is additionally used [Glembek et al., 2009], the final LLR is approximated by:

$$h_{\text{linear}}\left(\mathbb{O}_{\text{test}}, \boldsymbol{s}_i^*\right) = \left(\boldsymbol{s}_i^* - \boldsymbol{m}\right)^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{f}_{\text{test}\mid\boldsymbol{m}+\boldsymbol{U}\boldsymbol{x}_{\text{UBM,test}}}. \tag{3.47}$$

An illustration of the scoring for ISV and JFA is given in fig. 3.5(a) and fig. 3.5(b), respectively.

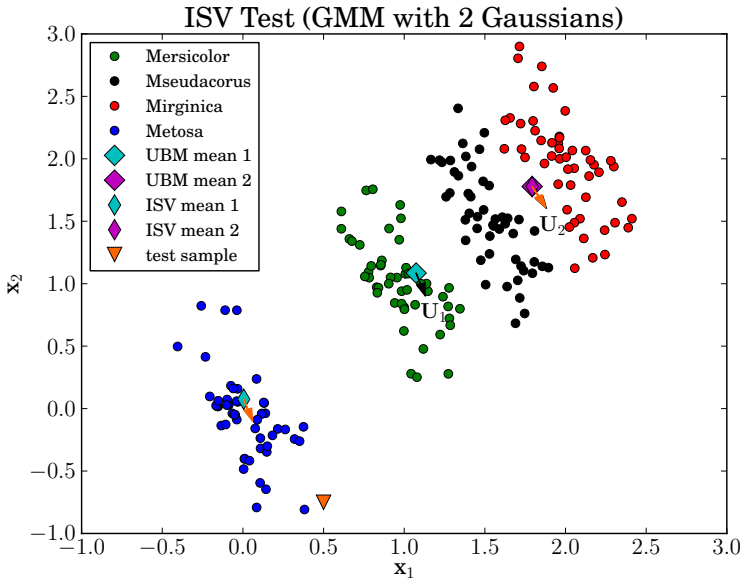
3.5 Total Variability Modeling (TV)

In [Dehak, 2009], it was shown that JFA can fail to separate between-class and within-class variations into two different subspaces. This is potentially caused by the high dimensionality of the GMM mean supervector space.

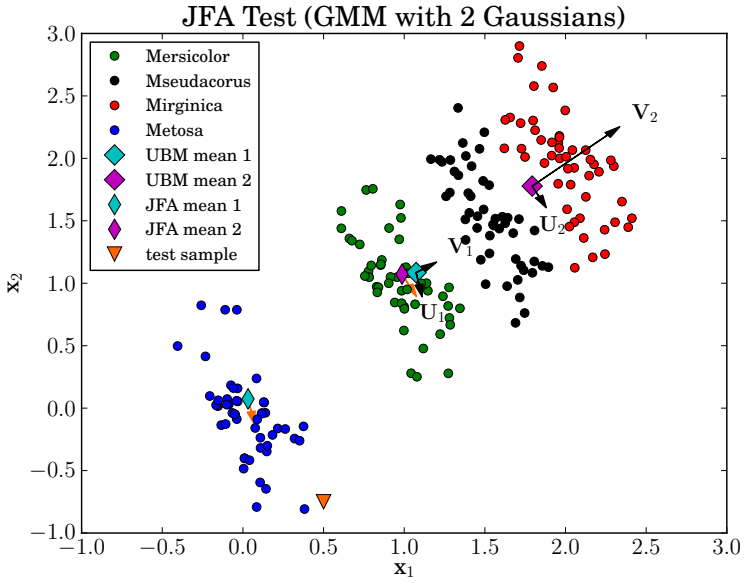
To address this issue, an alternative technique called *total variability* (TV) modeling was developed for speaker recognition [Dehak et al., 2009, Brümmer et al., 2010, Dehak et al., 2011] and later applied to face recognition [Wallace and McLaren, 2012]. This framework is built on the GMM approach and relies on the definition of a single subspace that contains both between-class and within-class variabilities. In particular, it aims to extract low-dimensional factors $\boldsymbol{v}$, so-called *i-vectors*, from biometric samples $\boldsymbol{\chi}$. More formally, the TV approach can be described in the GMM mean supervector space by:

$$\boldsymbol{\mu} = \boldsymbol{m} + \boldsymbol{T}\boldsymbol{v}, \tag{3.48}$$

where $\boldsymbol{T}$ is the low-dimensional total variability subspace and $\boldsymbol{v}$ the low-dimensional i-vector,



(a) ISV Test



(b) JFA Test

Figure 3.5 – TESTING PROCEDURE OF ISV AND JFA ON THE MIRIS DATASET. *This figure shows the testing procedures of the ISV and JFA techniques on the Miris synthetic dataset. The ranks of the subspace* $\boldsymbol{U}$ *and* $\boldsymbol{V}$ *are set to* 1*, and these subspaces are shown using black arrows. The orange arrows point to the compensated GMM client model against which the test samples (in orange) compute LLR scores.*

which is assumed to follow a normal distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$.

In contrast to ISV or JFA, TV does not explicitly perform session compensation. TV is just a

front-end that extracts a low dimensional i-vector $\boldsymbol{v}$ from each sample $\boldsymbol{\chi}$ based on the total variability of the training set. As such, it is likely to capture both class-specific and session-specific information. Hence, TV requires to use separate session compensation and scoring techniques after the extraction of i-vectors. However, this compensation is carried out in a low-dimensional space, the total variability space, instead of the high-dimensional GMM mean supervector space.

All these steps are described in the remainder of this chapter. Furthermore, the resulting i-vector toolchain is summarized in fig. 3.6.

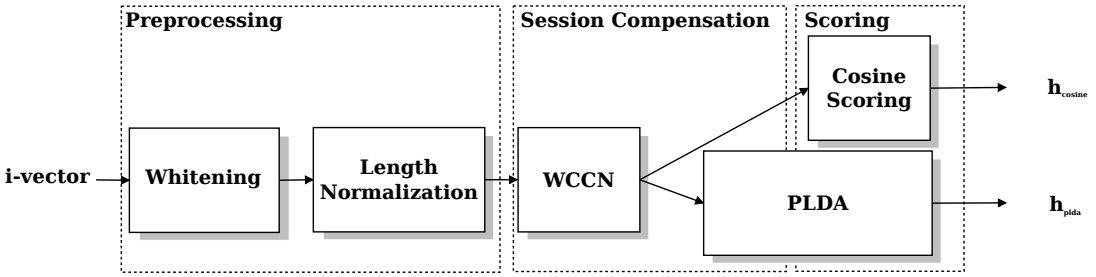


Figure 3.6 – I-VECTOR PROCESSING TOOLCHAIN. *This figure shows the steps of the i-vector processing toolchain employed in this thesis.*

3.5.1 Training

The TV subspace $\boldsymbol{T}$ is learned by maximizing the likelihood over a large training set, using the EM algorithm. This algorithm is similar to the one used to estimate the between-class subspace $\boldsymbol{V}$ in JFA (see alg. 5), with one major difference: while JFA jointly considers the samples coming from a given class, TV treats them as if they have been produced by different classes. This is an advantage, since large unlabeled training datasets can be used. In addition, TV relies on a covariance matrix $\boldsymbol{\Sigma}_{\text{TV}}$ that models the residual.

More formally, given a sample $\boldsymbol{\chi}_{i,j}$, the following equations hold for the latent variable $\boldsymbol{v}_{i,j}$:

$$\mathbb{E}\left[\boldsymbol{v}_{i,j}\right] = \left(\boldsymbol{I} + \boldsymbol{T}^{\top}\boldsymbol{\Sigma}_{\text{TV}}^{-1}\boldsymbol{N}_{i,j}\boldsymbol{T}\right)^{-1}\boldsymbol{T}^{\top}\boldsymbol{\Sigma}_{\text{TV}}^{-1}\boldsymbol{f}_{i,j|m}, \tag{3.49}$$

$$\mathbb{E}\left[\boldsymbol{v}_{i,j}\boldsymbol{v}_{i,j}^{\top}\right] = \left(\boldsymbol{I} + \boldsymbol{T}^{\top}\boldsymbol{\Sigma}_{\text{TV}}^{-1}\boldsymbol{N}_{i,j}\boldsymbol{T}\right)^{-1} + \mathbb{E}\left[\boldsymbol{v}_{i,j}\right]\mathbb{E}\left[\boldsymbol{v}_{i,j}\right]^{\top}. \tag{3.50}$$

They are estimated separately for each sample during the E-step.

Next, during the M-step, both the TV subspace $\boldsymbol{T}$ and the covariance matrix $\boldsymbol{\Sigma}_{\text{TV}}$ are updated. This consists of the update rules:

$$\boldsymbol{T}_c \left(\sum_{i=1}^{I} \sum_{j=1}^{J_i} \boldsymbol{N}_{i,j;c} \mathbb{E}\left[\boldsymbol{v}_{i,j} \boldsymbol{v}_{i,j}^{\top} \right] \right) = \sum_{i=1}^{I} \sum_{j=1}^{J_i} \boldsymbol{f}_{i,j;c|\boldsymbol{m}} \mathbb{E}\left[\boldsymbol{v}_{i,j} \right]^{\top}, \tag{3.51}$$

$$\boldsymbol{\Sigma}_{\mathrm{TV};c} = \frac{1}{n_c\left(\mathbb{O}_{\mathrm{train}}\right)} \sum_{i=1}^{I} \sum_{j=1}^{J_i} \left[\boldsymbol{S}_{i,j;c} - \frac{1}{2} \left(\boldsymbol{f}_{i,j;c} \mathbb{E}\left[\boldsymbol{v}_{i,j} \right]^{\top} \boldsymbol{T}_c^{\top} + \boldsymbol{T}_c \mathbb{E}\left[\boldsymbol{v}_{i,j} \right] \boldsymbol{f}_{i,j;c}^{\top} \right) \right]. \tag{3.52}$$

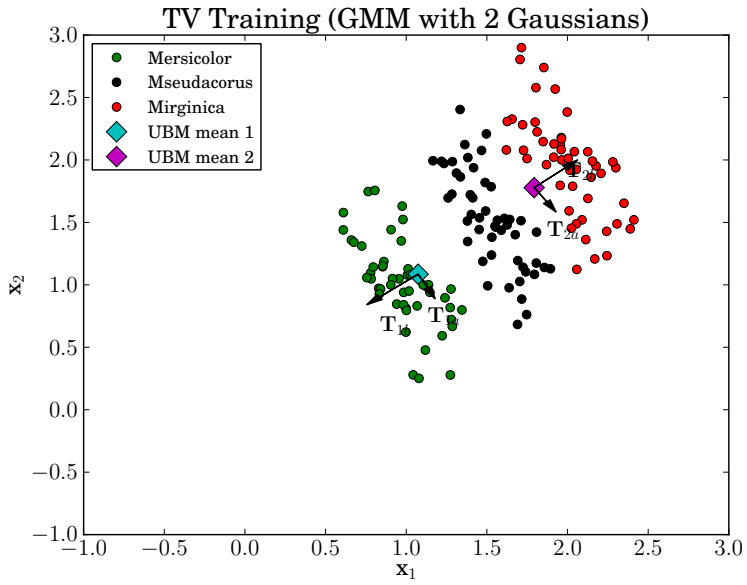


Figure 3.7 – TRAINING OF TV ON THE MIRIS DATASET. *This figure shows the application of the total variability modeling training procedure on the Miris synthetic dataset. The rank of* $\boldsymbol{T}$ *is set to* 2*. The subspace* $\boldsymbol{T}$ *is displayed with arrows, separately for each Gaussian component.*

In practice, $\boldsymbol{T}$ is initialized randomly and $\boldsymbol{\Sigma}_{\mathrm{TV}}$ with the covariance matrix of the UBM. In particular, we always consider a diagonal covariance matrix $\boldsymbol{\Sigma}_{\mathrm{TV}}$ in this thesis. The complete training procedure is summarized in alg. 6 and the application of this procedure to the Miris dataset is shown on fig. 3.7.

Looking at eq. (3.51) and eq. (3.52), it can be noticed that only sums over all the samples of the training step are required for the M-step. This observation can be applied to reduce the memory requirements within an implementation of this technique.
Furthermore, it is possible to split the computation of these sums when processing a large training set. Similarly to the ML training of the GMM, this allows to parallelize the E-step on several cores or nodes, each computation unit processing a subset of a partition of the training set.

Algorithm 6 Training Procedure for TV using Expectation-Maximization

$\boldsymbol{v}_{i,j} = \mathbf{0}$; $i = 1,\ldots,I$, $j = 1,\ldots,J_i$
Estimate $\boldsymbol{N}_{i,j}$ and $\boldsymbol{f}_{i,j|\boldsymbol{m}}$; $i = 1,\ldots,I$, $j = 1,\ldots,J_i$ `# eq. (3.18) and eq. (3.15)`
Initialize $\boldsymbol{T}$ randomly and $\boldsymbol{\Sigma}_{\text{TV}}$ with $\boldsymbol{\Sigma}$
for it = 1 to maximum number of expectation-maximization iterations **do**
    E-step: ***Estimate*** $\mathbb{E}\left[\boldsymbol{v}_{i,j}\right]$ `# eq. (3.49)`
        Estimate $\mathbb{E}\left[\boldsymbol{v}_i \boldsymbol{v}_{i,j}^\top\right]$ `# eq. (3.50)`
    M-step:
    for $c = 1$ to C **do**
        Update $\boldsymbol{T}_c$ `# eq. (3.51)`
        Update $\boldsymbol{\Sigma}_{\text{TV};c}$ `# eq. (3.52)`
    end for
end for
return TV subspace and covariance matrix $[\boldsymbol{T}, \boldsymbol{\Sigma}_{\text{TV}}]$

3.5.2 I-vector Extraction

Once TV training has been completed, the i-vector for a given sample $\boldsymbol{\chi}_{\text{test}}$ can be obtained using the following equation (which is similar to eq. (3.49)):

$$\boldsymbol{v}_{\text{test}} = \left(\boldsymbol{I} + \boldsymbol{T}^\top \boldsymbol{\Sigma}_{\text{TV}}^{-1} \boldsymbol{N}_{\text{test}} \boldsymbol{T}\right)^{-1} \boldsymbol{T}^\top \boldsymbol{\Sigma}_{\text{TV}}^{-1} \boldsymbol{f}_{\text{test}|\boldsymbol{m}}. \tag{3.53}$$

Since TV acts as a front-end feature extractor, any classification techniques can then be used. In practice, preprocessing algorithms, which map i-vectors into a more adequate space [Burget et al., 2011, Garcia-Romero and Espy-Wilson, 2011], as well as session compensation techniques are applied prior to classification.

3.5.3 I-vector Preprocessing

First, Cholesky *whitening* has been shown to boost classification performance [Burget et al., 2011, Wallace and McLaren, 2012]. Whitening consists of normalizing the i-vector space such that the covariance matrix of a training set of i-vectors is turned into the identity matrix. This is performed by applying:

$$\boldsymbol{v}^{(\text{whit})} = \boldsymbol{W}_{\text{WHIT}} \left(\boldsymbol{v} - \bar{\boldsymbol{v}}\right), \tag{3.54}$$

where $\bar{\boldsymbol{v}}$ is the mean of a training set of i-vectors, $\boldsymbol{v}^{(\text{whit})}$ the whitened i-vector, and $\boldsymbol{W}_{\text{WHIT}}$ the whitening transform. This transform $\boldsymbol{W}_{\text{WHIT}}$ is computed as the Cholesky decomposition of $\bar{\boldsymbol{\Sigma}}^{-1} = \boldsymbol{W}_{\text{WHIT}}^\top \left(\boldsymbol{W}_{\text{WHIT}}^\top\right)^\top$, where $\bar{\boldsymbol{\Sigma}}$ is the covariance matrix of a training set of i-vectors.

Another efficient preprocessing technique is i-vector *length normalization* [Garcia-Romero and Espy-Wilson, 2011, Wallace and McLaren, 2012], which aims at reducing the impact of a length mismatch between training and test i-vectors. It consists of mapping the i-vectors into

a unit hypersphere:

$$\boldsymbol{v}^{(\text{l-norm})} = \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|}, \tag{3.55}$$

which is very effective when using session compensation or scoring methods that assume Gaussian-like distributions.

3.5.4 Session Compensation using Within-Class Covariance Normalization

Within-class covariance normalization (WCCN) is a technique initially introduced for SVM-based speaker authentication [Hatch et al., 2006]. It aims to normalize the within-class covariance matrix of a training set of i-vectors. First the within-class scatter matrix is computed:

$$\boldsymbol{S}_W = \sum_{i=1}^{I} \left[\sum_{j=1}^{J_i} (\boldsymbol{v} - \bar{\boldsymbol{v}}_i)(\boldsymbol{v} - \bar{\boldsymbol{v}}_i)^\top \right], \tag{3.56}$$

where $\bar{\boldsymbol{v}}_i$ is the mean of the i-vectors of class i.

Given the number of classes I in the training set, the WCCN linear transform $\boldsymbol{W}_{\text{WCCN}}$ can be computed using the Cholesky decomposition of:

$$\left(\frac{1}{I}\boldsymbol{S}_W\right)^{-1} = \boldsymbol{W}_{\text{WCCN}}^\top \left(\boldsymbol{W}_{\text{WCCN}}^\top\right)^\top. \tag{3.57}$$

An i-vector $\boldsymbol{v}$ is projected into the corresponding WCCN space by:

$$\boldsymbol{v}^{(\text{WCCN})} = \boldsymbol{W}_{\text{WCCN}}\,\boldsymbol{v}. \tag{3.58}$$

3.5.5 Classification

Once session compensation has been performed, any scoring technique might be employed for classification purposes. In this work we consider two different strategies.

Cosine similarity scoring [Dehak et al., 2011] is a simple and efficient method used to estimate how close a (normalized) i-vector $\boldsymbol{v}_1$ extracted from a probe sample $\boldsymbol{\chi}_{\text{test}}$ is to the i-vector $\boldsymbol{v}_2$ representing a class i:

$$h_{\text{cosine}}(\boldsymbol{v}_1, \boldsymbol{v}_2) = \frac{\boldsymbol{v}_1 \cdot \boldsymbol{v}_2}{\|\boldsymbol{v}_1\|\,\|\boldsymbol{v}_2\|}. \tag{3.59}$$

Another technique commonly applied in the i-vector space is *probabilistic linear discriminant analysis* (PLDA) [Prince and Elder, 2007]. PLDA is a probabilistic framework that incorporates both between-class and within-class information and, therefore, performs session compensa-

tion.

3.6 Summary

This chapter introduced a set of generative probabilistic models. These techniques are built on top of the Gaussian mixture model (GMM) framework and aim at explicitly modeling session variability. In addition to the GMM framework, three approaches were described: inter-session variability (ISV) modeling, joint factor analysis (JFA) and total variability modeling (TV). ISV and JFA are supervised techniques that explicitly model and remove within-class variation using a low-dimensional subspace. In contrast, TV is an unsupervised learning technique, which projects samples in a low-dimensional subspace, where session compensation is carried out separately. In the next chapter, a description of the PLDA framework is presented, before proposing an exact and scalable formulation that drastically reduces the complexity of this approach.

4 Scalable Probabilistic Linear Discriminant Analysis

In this chapter, we describe *probabilistic linear discriminant analysis* (PLDA), a probabilistic and generative model that can be applied for various classification tasks. In particular, we show how to turn the original formulation [Prince and Elder, 2007] into a scalable formulation.

4.1 Original Formulation

Probabilistic linear discriminant analysis (PLDA) is a probabilistic generative model built on top of two latent variables that separately describes the between-class variations and the within-class variations.

4.1.1 Model Description

More formally, PLDA first assumes that the sample $\boldsymbol{o}_{i,j}$ (which is the j^{th} sample of the i^{th} class) can be described by the following generative process:

$$\boldsymbol{o}_{i,j} = f\left(\boldsymbol{h}_i, \boldsymbol{w}_{i,j}\right) + \boldsymbol{\epsilon}_{i,j}\,, \tag{4.1}$$

where:

1. $\boldsymbol{h}_i$ is a latent variable representing the between-class variations,
2. $\boldsymbol{w}_{i,j}$ is a latent variable representing the within-class variations,
3. $\boldsymbol{\epsilon}_{i,j}$ is a random variable describing the residual noise,
4. f is a function.

We use $D_{\boldsymbol{o}}$ to denote the dimensionality of the input sample $\boldsymbol{o}_{i,j}$. In addition, the PLDA model has a set of parameters, that we refer to as $\boldsymbol{\Theta}$ in the following.

The second assumption of the PLDA model is that the function f is bilinear, i.e., linear with respect to each of its latent variables $\boldsymbol{h}_i$ and $\boldsymbol{w}_{i,j}$. In practice, this means that eq. (4.1) can be

rewritten as follows:

$$\boldsymbol{o}_{i,j} = \boldsymbol{\mu} + \boldsymbol{F}\boldsymbol{h}_i + \boldsymbol{G}\boldsymbol{w}_{i,j} + \boldsymbol{\epsilon}_{i,j}\,, \tag{4.2}$$

where:

1. $\boldsymbol{\mu} + \boldsymbol{F}\boldsymbol{h}_i$ is a class-specific part,

2. $\boldsymbol{G}\boldsymbol{w}_{i,j} + \boldsymbol{\epsilon}_{i,j}$ constitutes the noise component.

The matrices $\boldsymbol{F}$ and $\boldsymbol{G}$ describe subspaces that contain the bases for the between-class variations and within-class variations, respectively. These subspaces are of dimensions $(D_{\boldsymbol{o}}, D_{\boldsymbol{F}})$ and $(D_{\boldsymbol{o}}, D_{\boldsymbol{G}})$, respectively. Furthermore, $\boldsymbol{F}$ and $\boldsymbol{G}$ are matrices of ranks $D_{\boldsymbol{F}} \leq D_{\boldsymbol{o}}$ and $D_{\boldsymbol{G}} \leq D_{\boldsymbol{o}}$. Correspondingly, $\boldsymbol{h}_i$ and $\boldsymbol{w}_{i,j}$ represent the position in these subspaces for $\boldsymbol{o}_{i,j}$ and are of dimensions $D_{\boldsymbol{F}}$ and $D_{\boldsymbol{G}}$.

The third assumption is the use of Gaussian priors for the variables $\boldsymbol{h}_i$, $\boldsymbol{w}_{i,j}$ and $\boldsymbol{\epsilon}_{i,j}$. Prior probabilities for the latent variables $\boldsymbol{h}_i$ and $\boldsymbol{w}_{i,j}$ are assumed to be Gaussian with zero mean and unit variance:

$$P(\boldsymbol{h}_i) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})\,, \tag{4.3}$$

$$P\left(\boldsymbol{w}_{i,j}\right) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})\,. \tag{4.4}$$

Finally, the residual $\boldsymbol{\epsilon}_{i,j}$, is defined to be Gaussian with zero mean and diagonal covariance $\boldsymbol{\Sigma}$.

$$P\left(\boldsymbol{\epsilon}_{i,j}\right) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})\,. \tag{4.5}$$

4.1.2 Inference

The process given by eq. (4.2) can be described in terms of a conditional probability:

$$P\left(\boldsymbol{o}_{i,j} \mid \boldsymbol{h}_i, \boldsymbol{w}_{i,j}, \boldsymbol{\Theta}\right) = \mathcal{N}\left(\boldsymbol{o}_{i,j} \mid \boldsymbol{\mu} + \boldsymbol{F}\boldsymbol{h}_i + \boldsymbol{G}\boldsymbol{w}_{i,j}, \boldsymbol{\Sigma}\right)\,, \tag{4.6}$$

where the parameters of the model are $\boldsymbol{\Theta} = \{\boldsymbol{\mu}, \boldsymbol{F}, \boldsymbol{G}, \boldsymbol{\Sigma}\}$. Eq. (4.3) and eq. (4.4) define the priors on the latent variables, $\boldsymbol{h}_i$ and $\boldsymbol{w}_{i,j}$, to be Gaussian. The equations above can be written in a more compact form by setting $\boldsymbol{A} = \begin{bmatrix} \boldsymbol{F} & \boldsymbol{G} \end{bmatrix}$ and

$$\boldsymbol{y}_{i,j} = \begin{bmatrix} \boldsymbol{h}_i \\ \boldsymbol{w}_{i,j} \end{bmatrix}\,. \tag{4.7}$$

This would result in:

$$\boldsymbol{o}_{i,j} = \boldsymbol{\mu} + \boldsymbol{A}\boldsymbol{y}_{i,j} + \boldsymbol{\epsilon}_{i,j}\,, \tag{4.8}$$

and:

$$P\left(\boldsymbol{o}_{i,j} \mid \boldsymbol{y}_{i,j}, \boldsymbol{\Theta}\right) = \mathcal{N}\left(\boldsymbol{o}_{i,j} \mid \boldsymbol{\mu} + \boldsymbol{A}\boldsymbol{y}_{i,j}, \boldsymbol{\Sigma}\right), \tag{4.9}$$

$$P\left(\boldsymbol{y}_{i,j}\right) = \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{I}\right). \tag{4.10}$$

The above formulation can be extended to handle multiple observations. For instance, given $J_i = 2$ observations for class i, we set:

$$\tilde{\boldsymbol{A}} = \begin{bmatrix} \boldsymbol{F} & \boldsymbol{G} & \boldsymbol{0} \\ \boldsymbol{F} & \boldsymbol{0} & \boldsymbol{G} \end{bmatrix}. \tag{4.11}$$

Consequently, we write that:

$$\tilde{\boldsymbol{x}}_i = \begin{bmatrix} \bar{\boldsymbol{x}}_{i,1} \\ \bar{\boldsymbol{x}}_{i,2} \end{bmatrix}, \quad \tilde{\boldsymbol{\epsilon}}_i = \begin{bmatrix} \boldsymbol{\epsilon}_{i,1} \\ \boldsymbol{\epsilon}_{i,2} \end{bmatrix}, \quad \tilde{\boldsymbol{w}}_i = \begin{bmatrix} \boldsymbol{w}_{i,1} \\ \boldsymbol{w}_{i,2} \end{bmatrix}, \tilde{\boldsymbol{\Sigma}} = \begin{bmatrix} \boldsymbol{\Sigma} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\Sigma} \end{bmatrix}, \tag{4.12}$$

and:

$$\tilde{\boldsymbol{y}}_i = \begin{bmatrix} \boldsymbol{h}_i \\ \boldsymbol{w}_{i,1} \\ \boldsymbol{w}_{i,2} \end{bmatrix}, \tag{4.13}$$

where:

$$\bar{\boldsymbol{x}}_{i,j} = \boldsymbol{o}_{i,j} - \boldsymbol{\mu}, \tag{4.14}$$

the tilde symbol ˜ indicating that the dimension of a variable (or a matrix) depends on the number of samples J_i for the class i.

This notation makes it explicit that we tie all of the J_i observations for class i to have the same latent class variable $\boldsymbol{h}_i$ but to have different latent session, or noise, variables $\boldsymbol{w}_{i,j}$; with such a formulation we drop the reference to $\boldsymbol{\mu}$, the mean of the data, in the Gaussian as it has already been subtracted from the samples in eq. (4.14).

For the general case of a class i with J_i samples, and keeping the same notation, the model could be rewritten:

$$\tilde{\boldsymbol{x}}_i = \tilde{\boldsymbol{A}}\tilde{\boldsymbol{y}}_i + \tilde{\boldsymbol{\epsilon}}_i. \tag{4.15}$$

This probabilistic model can be employed to address several recognition tasks. First, this requires to train the model, which is achieved using the expectation-maximization (EM) algorithm. Once the model has been trained, it is shown in [Prince and Elder, 2007] that central to the problem of recognition is the calculation of the likelihood that a set of observations, $\left[\boldsymbol{o}_{i,1}, \boldsymbol{o}_{i,2}, \ldots, \boldsymbol{o}_{i,J_i}\right]$, share the same latent class variable $\boldsymbol{h}_i$. Below we provide some more details

on each of these steps.

4.1.3 Training

To train the PLDA model an EM algorithm is used [Prince and Elder, 2007]. All of the M-steps are provided on a per sample basis, once the latent variables have been estimated. This estimation of the latent variables, corresponding to the E-step, presents the difficulties in making PLDA scalable. In the E-step, the first- and second-order moments of the latent variables need to be calculated:

$$\mathbb{E}\left[\tilde{\boldsymbol{y}}_i \mid \tilde{\boldsymbol{x}}_i, \boldsymbol{\Theta}\right] = \left(\tilde{\boldsymbol{I}} + \tilde{\boldsymbol{A}}^\top \tilde{\boldsymbol{\Sigma}}^{-1} \tilde{\boldsymbol{A}}\right)^{-1} \tilde{\boldsymbol{A}}^\top \tilde{\boldsymbol{\Sigma}}^{-1} \left(\tilde{\boldsymbol{x}}_i\right), \tag{4.16}$$

$$\mathbb{E}\left[\tilde{\boldsymbol{y}}_i \tilde{\boldsymbol{y}}_i^\top \mid \tilde{\boldsymbol{x}}_i, \boldsymbol{\Theta}\right] = \left(\tilde{\boldsymbol{I}} + \tilde{\boldsymbol{A}}^\top \tilde{\boldsymbol{\Sigma}}^{-1} \tilde{\boldsymbol{A}}\right)^{-1} + \mathbb{E}\left[\tilde{\boldsymbol{y}}_i \mid \tilde{\boldsymbol{x}}_i, \boldsymbol{\Theta}\right] \mathbb{E}\left[\tilde{\boldsymbol{y}}_i \mid \tilde{\boldsymbol{x}}_i, \boldsymbol{\Theta}\right]^\top. \tag{4.17}$$

From the above equations it is obvious that the problem for the E-step is to cope with the matrix $\left(\tilde{\boldsymbol{I}} + \tilde{\boldsymbol{A}}^\top \tilde{\boldsymbol{\Sigma}}^{-1} \tilde{\boldsymbol{A}}\right)^{-1}$ efficiently as it has to be recomputed for each iteration of EM. This matrix is, indeed, of dimension $(D_{\boldsymbol{F}} + J_i D_{\boldsymbol{G}}, D_{\boldsymbol{F}} + J_i D_{\boldsymbol{G}})$, and has to be used in calculations as well as stored. A solution to this problem is proposed in [Kenny, 2010] when applying PLDA to speaker recognition. Kenny's solution consists of applying a variational approximation to this inference problem; however, this approximation relies on a factorization, which assumes that the posterior variables are independent and whose quality with respect to the exact solution has not been demonstrated.

During the M-step, the values of the parameters $\boldsymbol{\Theta}$ are updated, according to the following rules, I being the number of classes:

$$\boldsymbol{\mu} = \frac{1}{I} \sum_{i=1}^{I} \left(\frac{1}{J_i} \sum_{j=1}^{J_i} \boldsymbol{o}_{i,j} \right), \tag{4.18}$$

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{F} & \boldsymbol{G} \end{bmatrix} = \left(\sum_{i=1}^{I} \sum_{j=1}^{J_i} \left(\boldsymbol{o}_{i,j} - \boldsymbol{\mu}\right) \mathbb{E}\left[\boldsymbol{y}_{i,j}^\top \mid \tilde{\boldsymbol{x}}_i, \boldsymbol{\Theta}\right] \right) \left(\mathbb{E}\left[\boldsymbol{y}_{i,j} \boldsymbol{y}_{i,j}^\top \mid \tilde{\boldsymbol{x}}_i, \boldsymbol{\Theta}\right] \right)^{-1}, \tag{4.19}$$

$$\boldsymbol{\Sigma} = \frac{1}{I} \sum_{i=1}^{I} \left(\frac{1}{J_i} \sum_{j=1}^{J_i} \operatorname{diag}\left[\left(\boldsymbol{o}_{i,j} - \boldsymbol{\mu}\right) \left(\boldsymbol{o}_{i,j} - \boldsymbol{\mu}\right)^\top - \boldsymbol{A} \mathbb{E}\left[\boldsymbol{y}_{i,j} \mid \tilde{\boldsymbol{x}}_i, \boldsymbol{\Theta}\right] \left(\boldsymbol{o}_{i,j} - \boldsymbol{\mu}\right)^\top \right] \right). \tag{4.20}$$

As shown in eq. (4.18), the expression of the mean $\boldsymbol{\mu}$ only depends on the samples $\boldsymbol{o}_{i,j}$ and, hence, only requires to be evaluated once, before the EM loop.

The training procedure is summarized in alg. 7 and depicted in fig. 4.1, considering the Miris synthetic dataset.

Algorithm 7 Training Procedure for PLDA using Expectation-Maximization

Training set: $\boldsymbol{o}_{i,j}$ with $i = 1,\ldots,I$, $j = 1,\ldots,J_i$
Compute $\boldsymbol{\mu}$ `# eq. (4.18)`
for it = 1 to maximum number of expectation-maximization iterations **do**
    E-step: *Evaluate the first- and second-order moments:*
    First-order moment

$$\mathbb{E}\left[\tilde{\boldsymbol{y}}_i \mid \tilde{\boldsymbol{x}}_i, \boldsymbol{\Theta}\right] \qquad \texttt{\# eq. (4.16)}$$

    Second-order moment

$$\mathbb{E}\left[\tilde{\boldsymbol{y}}_i \tilde{\boldsymbol{y}}_i^{\top} \mid \tilde{\boldsymbol{x}}_i, \boldsymbol{\Theta}\right] \qquad \texttt{\# eq. (4.17)}$$

    M-step: *Update the subspaces* $\boldsymbol{F}$ *and* $\boldsymbol{G}$ `# eq. (4.19)`
    Update the variance $\boldsymbol{\Sigma}$ `# eq. (4.20)`
end for
return PLDA model $\left[\boldsymbol{\mu}, \boldsymbol{F}, \boldsymbol{G}, \boldsymbol{\Sigma}\right]$

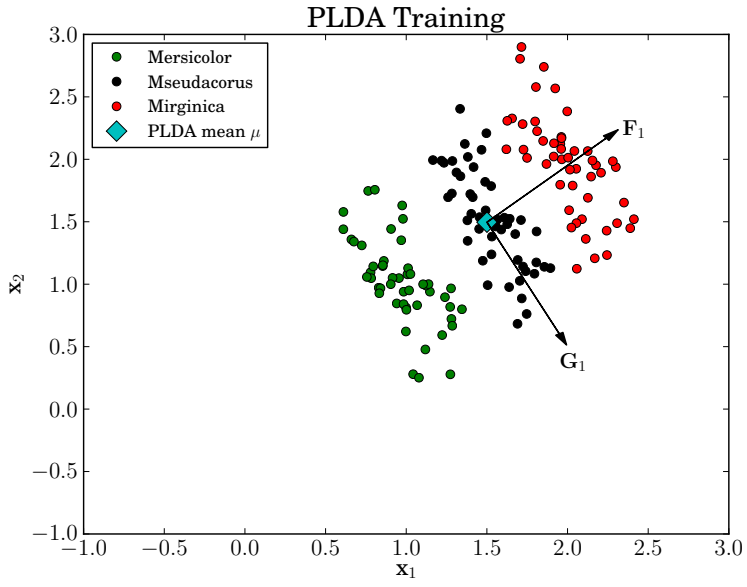


Figure 4.1 – TRAINING OF PLDA ON THE MIRIS DATASET. *This figure shows the subspaces learned, when applying the training procedure of PLDA on the Miris synthetic dataset. The ranks of the subspaces* $\boldsymbol{F}$ *and* $\boldsymbol{G}$ *are set to* 1. *These subspaces are shown with black arrows.*

4.1.4 Classification

Once the PLDA model has been trained, it can be used to perform various tasks, which all rely on likelihood calculations. Indeed, it is shown in [Li et al., 2012] that the model can be used to solve several problems ranging from identification and authentication through to registration. For this purpose, we aim to calculate the likelihood that a set of samples, $\tilde{\boldsymbol{x}}_i$, share the same latent class variable. This can be calculated in a probabilistic way by integrating out over all

of the latent variables. To do this, we tie together the latent class variable, $\boldsymbol{h}_i$, of the samples that we consider to have the same class and then consider each observation, $\boldsymbol{o}_{i,j}$, to have a separate latent session variable, $\boldsymbol{w}_{i,j}$. We then integrate over $\boldsymbol{h}_i$ and all of the individual $\boldsymbol{w}_{i,j}$. For the case of $J_i = 2$ this equates to:

$$P\left(\boldsymbol{o}_{i,1}, \boldsymbol{o}_{i,2}\right) = \int \left[\int P\left(\boldsymbol{o}_{i,1}, \mid \boldsymbol{h}_i, \boldsymbol{w}_{i,1}\right) P\left(\boldsymbol{w}_{i,1}\right) d\boldsymbol{w}_{i,1} \right. \\ \left. \int P\left(\boldsymbol{o}_{i,2}, \mid \boldsymbol{h}_i, \boldsymbol{w}_{i,2}\right) P\left(\boldsymbol{w}_{i,2}\right) d\boldsymbol{w}_{i,2} \right] P\left(\boldsymbol{h}_i\right) d\boldsymbol{h}_i \,, \tag{4.21}$$

where for brevity we have dropped the reference to $\boldsymbol{\Theta}$.

The above problem can be written as [Prince and Elder, 2007]: [1]

$$P\left(\tilde{\boldsymbol{x}}_i \mid \boldsymbol{\Theta}\right) = \mathcal{N}\left(\boldsymbol{0}, \tilde{\boldsymbol{\Sigma}} + \tilde{\boldsymbol{A}}\tilde{\boldsymbol{A}}^\top\right). \tag{4.22}$$

Equivalently, we can calculate the log-likelihood and write:

$$\ln\left[P\left(\tilde{\boldsymbol{x}}_i \mid \boldsymbol{\Theta}\right)\right] = -\frac{J_i D_{\boldsymbol{o}}}{2} \ln\left[2\pi\right] \\ -\frac{1}{2} \ln\left[\det\left(\tilde{\boldsymbol{\Sigma}} + \tilde{\boldsymbol{A}}\tilde{\boldsymbol{A}}^\top\right)\right] \\ -\frac{1}{2} \tilde{\boldsymbol{x}}_i^\top \left(\tilde{\boldsymbol{\Sigma}} + \tilde{\boldsymbol{A}}\tilde{\boldsymbol{A}}^\top\right)^{-1} \tilde{\boldsymbol{x}}_i \,. \tag{4.23}$$

Eq. (4.23) has three terms with various computational complexity, the first term being a simple constant factor depending upon the number of samples J_i. In contrast, the second term $\ln\left[\det\left(\tilde{\boldsymbol{\Sigma}} + \tilde{\boldsymbol{A}}\tilde{\boldsymbol{A}}^\top\right)\right]$ requires the computation of the determinant [2] of a matrix, which grows quadratically with the number of samples J_i. Finally, the third term $\tilde{\boldsymbol{x}}_i^\top \left(\tilde{\boldsymbol{\Sigma}} + \tilde{\boldsymbol{A}}\tilde{\boldsymbol{A}}^\top\right)^{-1} \tilde{\boldsymbol{x}}_i$ requires the inversion of the same large matrix and is also computationally demanding. An optimization to this quadratic term is suggested in [Li and Prince, 2010] (Section 3.2.3), by reformulating:

$$\tilde{\boldsymbol{x}}_i^\top \left(\tilde{\boldsymbol{\Sigma}} + \tilde{\boldsymbol{A}}\tilde{\boldsymbol{A}}^\top\right)^{-1} \tilde{\boldsymbol{x}}_i = \tilde{\boldsymbol{x}}_i^\top \tilde{\boldsymbol{\Sigma}}^{-1} \tilde{\boldsymbol{x}}_i - \tilde{\boldsymbol{x}}_i^\top \tilde{\boldsymbol{\Gamma}} \tilde{\boldsymbol{\Gamma}}^\top \tilde{\boldsymbol{x}}_i \,, \tag{4.24}$$

where $\tilde{\boldsymbol{\Gamma}} = \tilde{\boldsymbol{\Sigma}}^{-1} \tilde{\boldsymbol{A}} \left(\tilde{\boldsymbol{I}} + \tilde{\boldsymbol{A}}^\top \tilde{\boldsymbol{\Sigma}}^{-1} \tilde{\boldsymbol{A}}\right)^{-\frac{1}{2}}$. The first part of this new problem is easy to compute as $\tilde{\boldsymbol{\Sigma}}$ is diagonal. However the second term requires to calculate the square root of the large inverted matrix $\left(\tilde{\boldsymbol{I}} + \tilde{\boldsymbol{A}}^\top \tilde{\boldsymbol{\Sigma}}^{-1} \tilde{\boldsymbol{A}}\right)^{-1}$; although its computation might be performed offline, this matrix grows quadratically in size with the number of samples.

1. We have dropped the reference to $\boldsymbol{\mu}$ in this Gaussian model because we have subtracted it from the observations in (eq. (4.14)).

2. [Li and Prince, 2010, Li et al., 2012] implementation suggests that this can be decomposed into a series of simpler determinants by exploiting the structure of the PLDA model. In fact, we will show that it is possible to exploit this structure to reduce the complexity of most of the computationally demanding parts of both the training and the likelihood computation.

Two alternatives have been proposed to efficiently calculate the likelihood. [Li et al., 2012] suggested to use the predictive distribution rather than the joint distribution (as originally proposed in [Prince and Elder, 2007]), however, their solution does not explicitly solve the issue of multiple probe and enrollment samples. In [Kenny, 2010], another formulation for computing the likelihood is given where the lower bound of $\ln[P(\tilde{\boldsymbol{x}}_i \mid \boldsymbol{\Theta})]$ is used. In the next section, we show that, for the case of a Gaussian prior, a direct, exact and scalable solution can be derived, which obviates the need for storing the large inverted matrix, using predictive distributions or using the lower bound.

We note that central to providing a scalable solution for this problem, and the problem of training, is to find an efficient way to use the matrix $\left(\tilde{\boldsymbol{I}} + \tilde{\boldsymbol{A}}^\top \tilde{\boldsymbol{\Sigma}}^{-1} \tilde{\boldsymbol{A}}\right)^{-1}$. This matrix is of dimension $(D_{\boldsymbol{F}} + J_i D_{\boldsymbol{G}}, D_{\boldsymbol{F}} + J_i D_{\boldsymbol{G}})$ and grows quadratically with the number of samples thus limiting scalability. However, this matrix has a well defined structure, which can be exploited. In the next section, we show how an exact and scalable solution can be found for these problems.

4.2 Scalable Formulation

The overall idea of our approach consists of exploiting the structure of this probabilistic model by diagonalizing the model and then replacing full matrix inversions by a set of block matrix inversions, which are less computationally demanding. This scalable version of training is in contrast to previous solutions such as the nonscalable solution outlined in [Li et al., 2012] and the nonexact solution presented in [Kenny, 2010]. Also, the calculation of the likelihood is quite different to the solutions proposed in [Li et al., 2012] and [Kenny, 2010].

Considering the training procedure, the solution given in [Li et al., 2012] suggests that the computation of the moments of the tied latent variables $\tilde{\boldsymbol{y}}_i$ requires the matrix inversion $\left(\tilde{\boldsymbol{I}} + \tilde{\boldsymbol{A}}^\top \tilde{\boldsymbol{\Sigma}}^{-1} \tilde{\boldsymbol{A}}\right)^{-1}$. However, the following properties hold for the PLDA model and are described by the structure of the $\tilde{\boldsymbol{A}}$ matrix. First, the samples $\boldsymbol{o}_{i,j}$ of a given class i share a common term $\boldsymbol{F}\boldsymbol{h}_i$, but have separate confounding factors $\boldsymbol{G}\boldsymbol{w}_{i,j}$. Hence, the sum of the $\boldsymbol{o}_{i,j}$ for class i should intuitively be sufficient to estimate the latent variable $\boldsymbol{h}_i$. Second, each sample $\boldsymbol{o}_{i,j}$ is associated with a separate latent variable $\boldsymbol{w}_{i,j}$. In addition, the $\boldsymbol{w}_{i,j}$ are independent Gaussian samples. This implies that their independence is preserved if we consider any orthogonal mixtures of them.

In the following, we show that a simple change of variable allows us to diagonalize the PLDA model. In combination with simple linear algebra operations, this leads to a scalable formulation for both the training and the likelihood computation.

4.2.1 Change of Variable

Let $\tilde{\boldsymbol{U}}$ be any $J_i \times J_i$ orthogonal matrix whose first row $\tilde{\boldsymbol{u}}_0$ is $[1,\cdots,1]/\sqrt{J_i}$. The remaining rows can be anything with the constraint of $\tilde{\boldsymbol{U}}$ to be orthogonal. For instance, let the remaining rows $\tilde{\boldsymbol{u}}_j$ of $\tilde{\boldsymbol{U}}$ be (for $j \in \{1,\cdots,J_i-1\}$)

$$\tilde{\boldsymbol{u}}_j = \left[\underbrace{\frac{1}{\sqrt{j(j+1)}},\cdots,\frac{1}{\sqrt{j(j+1)}}}_{j \text{ identical positive terms}}, \frac{-j}{\sqrt{j(j+1)}}, 0, \cdots, 0 \right], \tag{4.25}$$

so that the row $\tilde{\boldsymbol{u}}_j$ has j identical positive terms followed by one negative term, sums to zero, and is of unit length.

Then, tensoring $\tilde{\boldsymbol{U}}$ with the identity matrix in the $\boldsymbol{o}_{i,j}$-space (also known as the Kronecker product) and multiplying by $\tilde{\boldsymbol{x}}_i$ leads to new variables $\mathring{\tilde{\boldsymbol{x}}}_i = \left(\tilde{\boldsymbol{U}} \otimes \boldsymbol{I}_{D_o}\right) \tilde{\boldsymbol{x}}_i$.[3] In particular, the upper part of $\mathring{\tilde{\boldsymbol{x}}}_i$ corresponds to a normalized sum of the $\bar{\boldsymbol{x}}_{i,j}$s, which is useful to estimate the class factor $\boldsymbol{h}_i$, as highlighted at the beginning of sec. 4.2. This transform could also be applied to the $\tilde{\boldsymbol{\epsilon}}_i$ variables leading to $\mathring{\tilde{\boldsymbol{\epsilon}}}_i$, and the following PLDA model:

$$\mathring{\tilde{\boldsymbol{x}}}_i = \left(\tilde{\boldsymbol{U}} \otimes \boldsymbol{I}_{D_o}\right) \tilde{\boldsymbol{A}} \tilde{\boldsymbol{y}}_i + \mathring{\tilde{\boldsymbol{\epsilon}}}_i \,. \tag{4.26}$$

In this case, the independence of the new variables $\mathring{\tilde{\boldsymbol{\epsilon}}}_i$ is preserved as the mixing matrix $\tilde{\boldsymbol{U}}$ is orthogonal.

Similarly, the change of variable could be applied to $\boldsymbol{w}_{i,j}$ by tensoring $\tilde{\boldsymbol{U}}$ with the identity matrix in the $\boldsymbol{w}_{i,j}$-space. Assuming that $\boldsymbol{h}_i$ is not updated by this change of variable, the transform leading to $\mathring{\tilde{\boldsymbol{y}}}_i$ would be as follows:

$$\mathring{\tilde{\boldsymbol{y}}}_i = \begin{bmatrix} \boldsymbol{I}_{D_F} & \boldsymbol{0} \\ \boldsymbol{0} & \tilde{\boldsymbol{U}} \otimes \boldsymbol{I}_{D_G} \end{bmatrix} \tilde{\boldsymbol{y}}_i = \tilde{\boldsymbol{V}} \tilde{\boldsymbol{y}}_i \,. \tag{4.27}$$

It is then straightforward to show that the PLDA model can be rewritten with the previously introduced variables as:

$$\mathring{\tilde{\boldsymbol{x}}}_i = \mathring{\tilde{\boldsymbol{A}}} \mathring{\tilde{\boldsymbol{y}}}_i + \mathring{\tilde{\boldsymbol{\epsilon}}}_i, \tag{4.28}$$

3. In the following, the circle symbol $\mathring{}$ is used for variables transformed by such a change of variable

with $\mathring{\tilde{\boldsymbol{A}}} = \left(\tilde{\boldsymbol{U}} \otimes \boldsymbol{I}_{D_o}\right) \tilde{\boldsymbol{A}} \tilde{\boldsymbol{V}}^{-1}$ and $\tilde{\boldsymbol{V}}$ being easy to invert as:

$$\tilde{\boldsymbol{V}}^{-1} = \begin{bmatrix} \boldsymbol{I}_{D_F} & \boldsymbol{0} \\ \boldsymbol{0} & \tilde{\boldsymbol{U}}^{\top} \otimes \boldsymbol{I}_{D_G} \end{bmatrix}. \tag{4.29}$$

Interestingly, $\mathring{\tilde{\boldsymbol{A}}}$ is more sparse than the original matrix $\tilde{\boldsymbol{A}}$ and is block diagonal, which justifies the choice of the previous change of variable:

$$\mathring{\tilde{\boldsymbol{A}}} = \begin{bmatrix} \sqrt{J_i}\boldsymbol{F} & \boldsymbol{G} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \ddots & \boldsymbol{0} \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{G} \end{bmatrix}. \tag{4.30}$$

In the following, we show that this indeed leads, for both the training and the likelihood computation, to matrix operations such as determinant computation and matrix inversion that can be performed very efficiently on a block basis, significantly reducing the complexity of this probabilistic approach.

Finally, the variables of our proposed solution have the following Gaussian distributions:

$$P\left(\mathring{\tilde{\boldsymbol{x}}}_i \mid \mathring{\tilde{\boldsymbol{y}}}_i, \boldsymbol{\Theta}\right) = \mathcal{N}\left[\mathring{\tilde{\boldsymbol{A}}}\mathring{\tilde{\boldsymbol{y}}}_i, \tilde{\boldsymbol{\Sigma}}\right], \tag{4.31}$$

$$P\left(\mathring{\tilde{\boldsymbol{y}}}_i\right) = \mathcal{N}\left[\boldsymbol{0}, \tilde{\boldsymbol{I}}\right] \text{ and } P\left(\mathring{\tilde{\boldsymbol{\epsilon}}}_i\right) = \mathcal{N}\left[\boldsymbol{0}, \tilde{\boldsymbol{\Sigma}}\right]. \tag{4.32}$$

4.2.2 Scalable Training

The goal is to be able to train the PLDA model so that it is scalable with respect to the number of training samples. For this purpose, an EM algorithm is used; however, the E-step of this algorithm has a bottleneck. With the original PLDA formulation, this leads to the computation of the matrix $\left(\tilde{\boldsymbol{I}} + \tilde{\boldsymbol{A}}^{\top}\tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\boldsymbol{A}}\right)^{-1}$, which grows quadratically with the number of samples, J_i. In contrast, the proposed diagonalization leads to a scalable E-step formulation. Using Bayes rule, the probability distribution of the latent variables of the transformed PLDA model can be written:

$$P\left(\mathring{\tilde{\boldsymbol{y}}}_i \mid \mathring{\tilde{\boldsymbol{x}}}_i, \boldsymbol{\Theta}\right) = \mathcal{N}\left[\mathring{\tilde{\boldsymbol{P}}}^{-1}\left(\mathring{\tilde{\boldsymbol{A}}}^{\top}\tilde{\boldsymbol{\Sigma}}^{-1}\mathring{\tilde{\boldsymbol{x}}}_i\right), \mathring{\tilde{\boldsymbol{P}}}^{-1}\right], \tag{4.33}$$

with $\mathring{\tilde{\boldsymbol{P}}} = \tilde{\boldsymbol{I}} + \mathring{\tilde{\boldsymbol{A}}}^{\top}\tilde{\boldsymbol{\Sigma}}^{-1}\mathring{\tilde{\boldsymbol{A}}}$ being block diagonal. The upper left block is:

$$\boldsymbol{P}_0 = \begin{bmatrix} \boldsymbol{I}_{D_F} + J_i\boldsymbol{F}^{\top}\boldsymbol{\Sigma}^{-1}\boldsymbol{F} & \sqrt{J_i}\boldsymbol{F}^{\top}\boldsymbol{\Sigma}^{-1}\boldsymbol{G} \\ \sqrt{J_i}\boldsymbol{G}^{\top}\boldsymbol{\Sigma}^{-1}\boldsymbol{F} & \boldsymbol{I}_{D_G} + \boldsymbol{G}^{\top}\boldsymbol{\Sigma}^{-1}\boldsymbol{G} \end{bmatrix}, \tag{4.34}$$

and the remaining $J_i - 2$ blocks are equal to:

$$\boldsymbol{P}_1 = \mathfrak{G}^{-1}, \tag{4.35}$$

with the (symmetric) matrix $\mathfrak{G}$ being defined by:

$$\mathfrak{G} = \left(\boldsymbol{I}_{D_G} + \boldsymbol{G}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{G}\right)^{-1}. \tag{4.36}$$

The inversion of $\boldsymbol{P}_0$ can be further optimized using a formulation of the inverse of a block matrix that uses the Schur complement. The corresponding identity can be found in [Ouellette, 1981] (Equations 1.11 and 1.10):

$$\begin{bmatrix} \boldsymbol{L} & \boldsymbol{M} \\ \boldsymbol{N} & \boldsymbol{O} \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{R}, & -\boldsymbol{R}\boldsymbol{M}\boldsymbol{O}^{-1} \\ -\boldsymbol{O}^{-1}\boldsymbol{N}\boldsymbol{R}, & \boldsymbol{O}^{-1} + \boldsymbol{O}^{-1}\boldsymbol{N}\boldsymbol{R}\boldsymbol{M}\boldsymbol{O}^{-1} \end{bmatrix}, \tag{4.37}$$

where we have substituted $\boldsymbol{R} = \left(\boldsymbol{L} - \boldsymbol{M}\boldsymbol{O}^{-1}\boldsymbol{N}\right)^{-1}$. Another related expression is the Woodbury matrix identity (see equation $C.7$ of [Bishop, 2007]), which states that:

$$(\boldsymbol{L} + \boldsymbol{M}\boldsymbol{O}\boldsymbol{N})^{-1} = \boldsymbol{L}^{-1} - \boldsymbol{L}^{-1}\boldsymbol{M}\left(\boldsymbol{O}^{-1} + \boldsymbol{N}\boldsymbol{L}^{-1}\boldsymbol{M}\right)^{-1}\boldsymbol{N}\boldsymbol{L}^{-1}. \tag{4.38}$$

Employing the last two identities leads to:

$$\boldsymbol{P}_0^{-1} = \begin{bmatrix} \mathfrak{F}_{J_i} & \sqrt{J_i}\mathfrak{H}^T \\ \sqrt{J_i}\mathfrak{H} & \left(\boldsymbol{I}_{D_G} - J_i \mathfrak{H} \boldsymbol{F}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{G}\right)\mathfrak{G} \end{bmatrix}, \tag{4.39}$$

where for legibility the following matrices have been defined:

$$\boldsymbol{Q} = \left(\boldsymbol{\Sigma} + \boldsymbol{G}\boldsymbol{G}^\top\right)^{-1} = \boldsymbol{\Sigma}^{-1} - \boldsymbol{\Sigma}^{-1}\boldsymbol{G}\mathfrak{G}\boldsymbol{G}^\top\boldsymbol{\Sigma}^{-1}, \tag{4.40}$$

$$\mathfrak{F}_{J_i} = \left(\boldsymbol{I}_{D_F} + J_i \boldsymbol{F}^\top \boldsymbol{Q}\boldsymbol{F}\right)^{-1}, \tag{4.41}$$

$$\mathfrak{H} = -\mathfrak{G}\boldsymbol{G}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{F}\mathfrak{F}_{J_i}. \tag{4.42}$$

Estimating the First-order Moment of the Latent Variables

This is given by the mean of the Gaussian distribution in eq. (4.33), which is $\mathring{\tilde{\boldsymbol{P}}}^{-1}\left(\mathring{\tilde{\boldsymbol{A}}}^\top \tilde{\boldsymbol{\Sigma}}^{-1} \mathring{\tilde{\boldsymbol{x}}}_i\right)$. $\mathring{\tilde{\boldsymbol{P}}}$ is diagonal by blocks and can be efficiently inverted (eq. (4.36) and eq. (4.39)). Then, the computation of $\mathring{\tilde{\boldsymbol{P}}}^{-1}\mathring{\tilde{\boldsymbol{A}}}^T\tilde{\boldsymbol{\Sigma}}^{-1}$ gives a block diagonal matrix, the first block being:

$$\begin{bmatrix} \sqrt{J_i}\mathfrak{F}_{J_i}\boldsymbol{F}^T\boldsymbol{Q} \\ \mathfrak{G}\boldsymbol{G}^T\boldsymbol{\Sigma}^{-1}\left(\boldsymbol{I}_{D_o} - J_i\boldsymbol{F}\mathfrak{F}_{J_i}\boldsymbol{F}^T\boldsymbol{Q}\right) \end{bmatrix}, \tag{4.43}$$

and the other ones being equal to $\mathfrak{G}\boldsymbol{G}^T\boldsymbol{\Sigma}^{-1}$.

As $\boldsymbol{h}_i$ is not affected by the change of variable (eq. (4.27)), it corresponds to the upper subvector of $\mathring{\tilde{\boldsymbol{y}}}_i$. Therefore, the first-order moment of $\boldsymbol{h}_i$ is directly obtained by multiplying the first block-rows of the matrix $\mathring{\tilde{\boldsymbol{P}}}^{-1}\mathring{\tilde{\boldsymbol{A}}}^T\mathring{\tilde{\boldsymbol{\Sigma}}}^{-1}$ with $\mathring{\tilde{\boldsymbol{x}}}_i$, which gives:

$$\mathbb{E}\left[\boldsymbol{h}_i \mid \tilde{\boldsymbol{x}}_i, \boldsymbol{\Theta}\right] = \tilde{\mathfrak{F}}_{J_i} \sum_j \boldsymbol{F}^\top \boldsymbol{Q} \bar{\boldsymbol{x}}_{i,j}\,. \tag{4.44}$$

Considering only the $\mathring{\tilde{\boldsymbol{w}}}_i$ (lower) subvector of $\mathring{\tilde{\boldsymbol{y}}}_i$, the corresponding (lower) part $\mathring{\tilde{\boldsymbol{B}}}$ of the matrix $\mathring{\tilde{\boldsymbol{P}}}^{-1}\mathring{\tilde{\boldsymbol{A}}}^T\mathring{\tilde{\boldsymbol{\Sigma}}}^{-1}$ can be decomposed into a sum of two matrices, the first one being sparse with a single non-zero block (upper left) equal to $\boldsymbol{B}_0 = -J_i \tilde{\mathfrak{G}} \boldsymbol{G}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{F} \tilde{\mathfrak{F}}_{J_i} \boldsymbol{F}^T \boldsymbol{Q}$, and the second one being diagonal by blocks with identical blocks $\boldsymbol{B}_1 = \tilde{\mathfrak{G}} \boldsymbol{G}^T \boldsymbol{\Sigma}^{-1}$:

$$\mathring{\tilde{\boldsymbol{B}}} = \begin{bmatrix} \boldsymbol{B}_0 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} + \begin{bmatrix} \boldsymbol{B}_1 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \ddots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{B}_1 \end{bmatrix}. \tag{4.45}$$

Furthermore, the first-order moment of the variables $\tilde{\boldsymbol{w}}_i$ is given by

$$\begin{aligned} \mathbb{E}\left[\tilde{\boldsymbol{w}}_i \mid \tilde{\boldsymbol{x}}_i, \boldsymbol{\Theta}\right] = &\left(\tilde{\boldsymbol{U}}^T \otimes \boldsymbol{I}_{D_G}\right) \begin{bmatrix} \boldsymbol{B}_0 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \mathring{\tilde{\boldsymbol{x}}}_i \\ &+ \left(\tilde{\boldsymbol{U}}^T \otimes \boldsymbol{I}_{D_G}\right) \begin{bmatrix} \boldsymbol{B}_1 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \ddots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{B}_1 \end{bmatrix} \left(\tilde{\boldsymbol{U}} \otimes \boldsymbol{I}_{D_o}\right) \tilde{\boldsymbol{x}}_i. \end{aligned} \tag{4.46}$$

The previous decomposition greatly simplifies the computation, and leads to the following expression for each $\boldsymbol{w}_{i,j}$

$$\mathbb{E}\left[\boldsymbol{w}_{i,j} \mid \tilde{\boldsymbol{x}}_i, \boldsymbol{\Theta}\right] = \tilde{\mathfrak{G}} \boldsymbol{G}^T \boldsymbol{\Sigma}^{-1} \bar{\boldsymbol{x}}_{i,j} - \tilde{\mathfrak{G}} \boldsymbol{G}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{F} \tilde{\mathfrak{F}}_{J_i} \boldsymbol{F}^T \boldsymbol{Q} \sum_j \bar{\boldsymbol{x}}_{i,j} \tag{4.47}$$

which, after grouping the common factors, finally provides:

$$\mathbb{E}\left[\boldsymbol{w}_{i,j} \mid \tilde{\boldsymbol{x}}_i, \boldsymbol{\Theta}\right] = \tilde{\mathfrak{G}} \boldsymbol{G}^\top \boldsymbol{\Sigma}^{-1} \left(\bar{\boldsymbol{x}}_{i,j} - \boldsymbol{F}\mathbb{E}\left[\boldsymbol{h}_i \mid \tilde{\boldsymbol{x}}_i, \boldsymbol{\Theta}\right]\right). \tag{4.48}$$

Estimating the Second-order Moment of the Latent Variables

Calculating the expected value of the second-order moment of the latent variables, eq. (4.17), in a scalable manner is more difficult.

This second-order moment can be expressed as the sum of both the square of the mean and the variance. The mean has already been expressed in eq. (4.44) and eq. (4.48). The computation

of the variance is performed using the expression of $\mathring{\tilde{\boldsymbol{P}}}^{-1}$ previously evaluated, and reverting the change of variable. This leads to:

$$\tilde{\mathfrak{P}}^{-1} = \tilde{\boldsymbol{V}}^{-1}\mathring{\tilde{\boldsymbol{P}}}^{-1}\tilde{\boldsymbol{V}}, \tag{4.49}$$

where $\tilde{\boldsymbol{V}}^{-1}$ is efficiently computed as:

$$\tilde{\boldsymbol{V}}^{-1} = \begin{bmatrix} \boldsymbol{I}_{D_F} & \boldsymbol{0} \\ \boldsymbol{0} & \tilde{\boldsymbol{U}}^T \otimes \boldsymbol{I}_{D_G} \end{bmatrix}. \tag{4.50}$$

In addition, the variance of the latent variables is only ever used on a per sample basis, which implies that only the elements of the first row, first column and diagonal of $\tilde{\mathfrak{P}}^{-1}$ are necessary. A direct computation provides a matrix which is as follows:

$$\begin{bmatrix} \tilde{\mathfrak{F}}_{J_i} & \mathfrak{H}^T & \cdots & \mathfrak{H}^T \\ \mathfrak{H} & \mathfrak{D} & & \\ \vdots & & \ddots & \\ \mathfrak{H} & & & \mathfrak{D} \end{bmatrix},$$

with $\mathfrak{D} = \left(\boldsymbol{I}_{D_G} - \mathfrak{H}\boldsymbol{F}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{G}\right)\mathfrak{G}$. Exploiting the structure of this matrix leads to:

$$\mathrm{Var}\left[\boldsymbol{y}_{i,j} \mid \tilde{\boldsymbol{x}}_i, \boldsymbol{\Theta}\right] = \begin{bmatrix} \tilde{\mathfrak{F}}_{J_i} & \mathfrak{H}^\top \\ \mathfrak{H} & \left(\boldsymbol{I}_{D_G} - \mathfrak{H}\boldsymbol{F}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{G}\right)\mathfrak{G} \end{bmatrix}. \tag{4.51}$$

Finally the corresponding term involved in the E-Step update rules is given by:

$$\mathbb{E}\left[\tilde{\boldsymbol{y}}_i\tilde{\boldsymbol{y}}_i^T \mid \tilde{\boldsymbol{x}}_i, \boldsymbol{\Theta}\right] = \mathrm{Var}\left[\tilde{\boldsymbol{y}}_i \mid \tilde{\boldsymbol{x}}_i, \boldsymbol{\Theta}\right] + \mathbb{E}\left[\tilde{\boldsymbol{y}}_i \mid \tilde{\boldsymbol{x}}_i, \boldsymbol{\Theta}\right]\mathbb{E}\left[\tilde{\boldsymbol{y}}_i \mid \tilde{\boldsymbol{x}}_i, \boldsymbol{\Theta}\right]^T. \tag{4.52}$$

The resulting exact and scalable training procedure is provided in alg. 8.

4.2.3 Scalable Likelihood

Similarly to eq. (4.22), the likelihood of the proposed solution is obtained by integrating out the latent variable $\mathring{\tilde{\boldsymbol{y}}}_i$:

$$P\left(\mathring{\tilde{\boldsymbol{x}}}_i \mid \boldsymbol{\Theta}\right) = \mathcal{N}\left[\boldsymbol{0}, \tilde{\boldsymbol{\Sigma}} + \mathring{\tilde{\boldsymbol{A}}}\mathring{\tilde{\boldsymbol{A}}}^\top\right], \tag{4.53}$$

which can be split into three terms (like in eq. (4.23)), the last two of them being difficult to evaluate, as they require to compute, respectively, the determinant and the inverse of the

Algorithm 8 Scalable Training Procedure for PLDA using Expectation-Maximization

Training set: $\boldsymbol{o}_{i,j}$ with $i = 1,\dots,I$, $j = 1,\dots,J_i$
Compute $\boldsymbol{\mu}$ `# eq. (4.18)`
for it = 1 to maximum number of expectation-maximization iterations **do**
  E-step: *Evaluate the first- and second-order moments:*
  Precompute $\mathfrak{G}$, $\boldsymbol{Q}$, $\mathfrak{F}_{J_i}$ *and* $\mathfrak{H}$ `# eq. (4.36), eq. (4.40), eq. (4.41) and eq. (4.42)`
  First-order moment

$$\mathbb{E}\left[\tilde{\boldsymbol{y}}_i \mid \tilde{\boldsymbol{x}}_i, \boldsymbol{\Theta}\right] \qquad \text{\# eq. (4.44) and eq. (4.48)}$$

  Second-order moment

$$\mathbb{E}\left[\tilde{\boldsymbol{y}}_i \tilde{\boldsymbol{y}}_i^\top \mid \tilde{\boldsymbol{x}}_i, \boldsymbol{\Theta}\right] \qquad \text{\# eq. (4.52)}$$

  M-step: *Update the subspaces* $\boldsymbol{F}$ *and* $\boldsymbol{G}$ `# eq. (4.19)`
  Update the variance $\boldsymbol{\Sigma}$ `# eq. (4.20)`
end for
return PLDA model $\left[\boldsymbol{\mu}, \boldsymbol{F}, \boldsymbol{G}, \boldsymbol{\Sigma}\right]$

following large matrix:

$$\left(\tilde{\boldsymbol{\Sigma}} + \mathring{\tilde{\boldsymbol{A}}}\mathring{\tilde{\boldsymbol{A}}}^\top\right). \tag{4.54}$$

However, with our proposed solution, $\tilde{\boldsymbol{\Sigma}} + \mathring{\tilde{\boldsymbol{A}}}\mathring{\tilde{\boldsymbol{A}}}^\top$ is a block diagonal matrix, the upper left block being $\boldsymbol{\Sigma} + J_i\boldsymbol{F}\boldsymbol{F}^\top + \boldsymbol{G}\boldsymbol{G}^\top$ and the $J_i - 1$ other ones being equal to $\boldsymbol{\Sigma} + \boldsymbol{G}\boldsymbol{G}^\top$. Therefore, some simplifications are possible to improve the efficiency of the approach.

The second term of the likelihood in eq. (4.23), which involves the computation of the determinant of this large matrix, can be efficiently computed using the following decomposition:

$$\det\left(\tilde{\boldsymbol{\Sigma}} + \mathring{\tilde{\boldsymbol{A}}}\mathring{\tilde{\boldsymbol{A}}}^\top\right) = \det\left(\tilde{\boldsymbol{\Sigma}}\right)\det\left(\tilde{\boldsymbol{I}} + \mathring{\tilde{\boldsymbol{A}}}^\top\tilde{\boldsymbol{\Sigma}}^{-1}\mathring{\tilde{\boldsymbol{A}}}\right), \tag{4.55}$$

and then using the block decomposition of $\tilde{\boldsymbol{I}} + \mathring{\tilde{\boldsymbol{A}}}^\top\tilde{\boldsymbol{\Sigma}}^{-1}\mathring{\tilde{\boldsymbol{A}}}$ given by eq. (4.34) and eq. (4.35) leading to:

$$\det\left(\tilde{\boldsymbol{\Sigma}} + \mathring{\tilde{\boldsymbol{A}}}\mathring{\tilde{\boldsymbol{A}}}^\top\right) = \det\left(\boldsymbol{\Sigma}\right)^{J_i}\det\left(\mathfrak{G}^{-1}\right)^{J_i}\det\left(\mathfrak{F}_{J_i}^{-1}\right). \tag{4.56}$$

The third term involved in the computation of the likelihood relies on the inversion of the matrix $\tilde{\boldsymbol{\Sigma}} + \mathring{\tilde{\boldsymbol{A}}}\mathring{\tilde{\boldsymbol{A}}}^T$. This matrix is block diagonal, the upper left block being $\mathfrak{T}_0 = \boldsymbol{\Sigma} + J_i\boldsymbol{F}\boldsymbol{F}^T + \boldsymbol{G}\boldsymbol{G}^T$ and the $J_i - 1$ other ones being equal to $\mathfrak{T}_1 = \boldsymbol{\Sigma} + \boldsymbol{G}\boldsymbol{G}^T$. Therefore, this matrix inversion can be efficiently performed on a block basis.

In addition, $\boldsymbol{G}$ is of dimension $(D_{\boldsymbol{o}}, D_{\boldsymbol{G}})$, usually with $D_{\boldsymbol{G}} \ll D_{\boldsymbol{o}}$ and hence, $\boldsymbol{G}\boldsymbol{G}^T$ is a potentially

low rank matrix. Furthermore, it is indeed possible to further reduce the complexity of each block inversion $\mathfrak{T}_0^{-1}$ and $\mathfrak{T}_1^{-1}$, using the Woodbury matrix identity eq. (4.38). Applying this identity to compute the inverse of the block $\mathfrak{T}_1$ leads to:

$$\mathfrak{T}_1^{-1} = \boldsymbol{\Sigma}^{-1} - \boldsymbol{\Sigma}^{-1}\boldsymbol{G}\mathfrak{G}\boldsymbol{G}^T\boldsymbol{\Sigma}^{-1} = \boldsymbol{Q}. \tag{4.57}$$

Similarly, this identity could consecutively be applied twice to compute the inverse of the other block $\mathfrak{T}_0$, which gives:

$$\mathfrak{T}_0^{-1} = \boldsymbol{Q} - J_i\boldsymbol{Q}^T\boldsymbol{F}\mathfrak{F}_{J_i}\boldsymbol{F}^T\boldsymbol{Q}, \tag{4.58}$$

Finally, the likelihood is expressed as a function of the transformed input $\mathring{\bar{\boldsymbol{x}}}_i$. However, it is easy to notice that the change of variable could be easily reverted without increasing the complexity of the likelihood computation. $\left(\tilde{\boldsymbol{\Sigma}} + \mathring{\tilde{\boldsymbol{A}}}\mathring{\tilde{\boldsymbol{A}}}^T\right)^{-1}$ can indeed be expressed as a sum of two matrices, the first one being diagonal by blocks with identical blocks $\boldsymbol{Q}$, and the second one being sparse with a single non-zero block (upper left) equal to $J_i\boldsymbol{Q}^T\boldsymbol{F}\mathfrak{F}_{J_i}\boldsymbol{F}^T\boldsymbol{Q}$. This provides a formulation similar to the right hand side of eq. (4.45). Next, to revert the change of variable, the left hand side of these matrices are multiplied by $\left(\tilde{\boldsymbol{U}} \otimes \boldsymbol{I}_{D_o}\right)^T$ while the right hand side is multiplied by $\left(\tilde{\boldsymbol{U}} \otimes \boldsymbol{I}_{D_o}\right)$. For the first matrix, all the blocks are the same, and hence the mixing matrices have no influence. The second matrix uses the sum of the input samples for the class. This finally gives:

$$\begin{aligned}\mathring{\bar{\boldsymbol{x}}}_i^\top \left(\tilde{\boldsymbol{\Sigma}} + \mathring{\tilde{\boldsymbol{A}}}\mathring{\tilde{\boldsymbol{A}}}^\top\right)^{-1} \mathring{\bar{\boldsymbol{x}}}_i = &\sum_{j=1}^{J_i} \bar{\boldsymbol{x}}_{i,j}^\top \boldsymbol{\Sigma}^{-1} \bar{\boldsymbol{x}}_{i,j} \\ &- \left(\sum_{j=1}^{J_i} \bar{\boldsymbol{x}}_{i,j}^\top \boldsymbol{Q}^\top \boldsymbol{F}\right) \mathfrak{F}_{J_i} \left(\sum_{j=1}^{J_i} \boldsymbol{F}^\top \boldsymbol{Q} \bar{\boldsymbol{x}}_{i,j}\right) \\ &- \sum_{j=1}^{J_i} \bar{\boldsymbol{x}}_{i,j}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{G}\mathfrak{G}\boldsymbol{G}^\top \boldsymbol{\Sigma}^{-1} \bar{\boldsymbol{x}}_{i,j}.\end{aligned} \tag{4.59}$$

If the likelihood computation involves the same number of samples J_i several times, the above expression might be further optimized using the same trick as proposed in [Li et al., 2012]. The square root of the matrices $\mathfrak{F}_{J_i}$ and $\mathfrak{G}$ could be precomputed and then used to calculate one half of each the last two terms before taking their magnitude. It is important to notice that the change of variable does not affect the PLDA model, and hence, the above formulae remain valid to compute the likelihood given by eq. (4.23).

Finally, the above equations show that storing the information for a model (enrollment of a given class) reduces to a single low-dimensional feature vector such as $\left(\sum_{j=1}^{J_i} \boldsymbol{F}^\top \boldsymbol{Q} \bar{\boldsymbol{x}}_{i,j}\right)$ and a scalar (corresponding to the other terms involved in the likelihood computation), which further emphasizes the efficiency and scalability of the proposed approach.

Complexity		[Li et al., 2012]	Our approach
Likelihood computation	Memory	$\mathcal{O}\left(J_i^2\right)$	$\mathcal{O}\left(1\right)$
	Time	$\mathcal{O}\left(J_i^2\right)$	$\mathcal{O}\left(J_i\right)$
Training	Memory	$\mathcal{O}\left(J_i^2\right)$	$\mathcal{O}\left(1\right)$
	Time	$\mathcal{O}\left(J_i^3\right)$	$\mathcal{O}\left(J_i\right)$

Table 4.1 – COMPLEXITY OF THE PLDA MODEL. *This table shows the complexity of the PLDA model with respect to the number of samples J_i for the class, for both the likelihood computation and the training, assuming that matrices have been precomputed whenever possible. The likelihood complexity of [Li et al., 2012] is the one for the joint distribution approach (Section* 3.2 *in [Li et al., 2012]).*

4.3 Complexity

In this section we analyze the complexity of our proposed solution against the solution proposed in [Li et al., 2012]. We start by first examining the complexity of training the PLDA model and then finish with some analysis and comments on the complexity of computing the likelihood. We refer to time and memory complexity to express respectively the time and the memory required to run the algorithm. A quick summary of the below analysis is provided in Table 4.1.

4.3.1 Training

As mentioned previously, one of the most computationally demanding parts of the PLDA approach occurs during the E-step of the training algorithm. The E-step update rules given in [Li et al., 2012] rely on the usage of the matrix $\left(\tilde{\boldsymbol{I}} + \tilde{\boldsymbol{A}}^\top \tilde{\boldsymbol{\Sigma}}^{-1} \tilde{\boldsymbol{A}}\right)$, which grows quadratically with the number of samples. This suggests that the training procedure is demanding as for each iteration of EM, this matrix has to be inverted and multiplied with the similarly large matrix $\tilde{\boldsymbol{A}}$. The memory complexity is given by $\mathcal{O}\left(J_i^2\right)$, whereas the time complexity is $\mathcal{O}\left(J_i^3\right)$. [4] In contrast, the update rules of our proposed approach provided in sec. 4.2.2 show that exploiting the structure of the PLDA model leads to a much more efficient training procedure.

Considering the computation of the first-order moment of the latent variables, this is in contrast to [Li et al., 2012] performed on a per sample basis. Therefore, the time complexity is linear with the number of samples J_i as opposed to cubic (inversion of a large matrix, which grows quadratically). In addition, the matrices involved in eq. (4.44) and eq. (4.48)) are common to all the training samples for the class. Therefore, they can be precomputed and the memory requirement is constant, irrespective of the number of samples. This leads to a memory complexity of $\mathcal{O}\left(1\right)$. And, as for the likelihood computation, if the number of training samples for the class J_i has changed, only the matrix $\mathfrak{F}_{J_i}$ needs to be recomputed.

4. For readability, we consider that the time complexity of inverting a square matrix of size (N, N) is given by $\mathcal{O}\left(N^3\right)$. Using a non-naive approach such as the popular Strassen algorithm, it can be improved to $\mathcal{O}\left(N^\alpha\right)$ with $\alpha \approx 2.81$.

Finally, the second-order moment of the latent variables is only used on a per sample basis, as already depicted in previous work [Prince and Elder, 2007, Li et al., 2012]. Furthermore, this has linear time complexity with the number of samples J_i, and once again, many matrices can be precomputed by examining eq. (4.51) to further optimize the training procedure. In addition, only the sum of the second-order moments is required for the maximization step, which does not affect the memory complexity for training, which is $\mathcal{O}(J_i)$.

4.3.2 Likelihood Computation

Comparing the naive way to compute the likelihood given by eq. (4.23) with our proposed solution (given by eq. (4.56) and eq. (4.59)), several conclusions can be drawn.

Considering memory usage, the matrix $\left(\tilde{\boldsymbol{I}} + \tilde{\boldsymbol{A}}^\top \tilde{\boldsymbol{\Sigma}}^{-1} \tilde{\boldsymbol{A}}\right)$ involved in the approach of [Li et al., 2012] is of dimension $(D_{\boldsymbol{F}} + J_i D_{\boldsymbol{G}}, D_{\boldsymbol{F}} + J_i D_{\boldsymbol{G}})$, and, hence, the memory complexity is $\mathcal{O}\left(J_i^2\right)$. In contrast, the proposed solution exploits the structure of the problem by using several smaller matrices, such as $\mathfrak{G}$, $\boldsymbol{Q}$ or $\mathfrak{F}_{J_i}$, which are of constant dimension, irrespective of the number of samples. This implies that the memory complexity is $\mathcal{O}(1)$.

In addition, many of these matrices can be precomputed. For instance, $\mathfrak{G}$ or $\boldsymbol{Q}$ are required to compute the likelihood of any set of samples. Besides, the inverted matrix $\mathfrak{F}_{J_i}$ can be precomputed and used to compute the likelihood of any set of J_i samples. If the number of samples, J_i, involved in the likelihood computation has changed, the only new and required matrix inversion is for the $\mathfrak{F}_{J_i}$ matrix which is of dimension $(D_{\boldsymbol{F}}, D_{\boldsymbol{F}})$. In contrast, with the naive solution, the inverted matrix $\left(\tilde{\boldsymbol{\Sigma}} + \tilde{\boldsymbol{A}}\tilde{\boldsymbol{A}}^\top\right)^{-1}$ has to be recomputed. Furthermore the time complexity considering matrix inversion and with respect to the number of samples J_i has in this case become $\mathcal{O}(1)$ instead of $\mathcal{O}\left(J_i^3\right)$.

Finally, assuming that all the previous matrices involved in the likelihood computation have been precomputed, the likelihood of J_i samples given the PLDA model requires the calculation of products $\tilde{\boldsymbol{\Gamma}}^\top \mathring{\boldsymbol{x}}_i$. The time complexity with respect to the number of samples for the class J_i is then quadratic, which is $\mathcal{O}\left(J_i^2\right)$. With the proposed solution and using the square root of the matrices $\mathfrak{G}$ and $\mathfrak{F}_{J_i}$, this involves several matrix-vector multiplications such as $\left(\mathfrak{G}^{\frac{1}{2}} \boldsymbol{G}^\top \boldsymbol{\Sigma}^{-1}\right) \bar{\boldsymbol{x}}_{i,j}$, and the time complexity is given by $\mathcal{O}(J_i)$.

4.4 Summary

This chapter introduced probabilistic linear discriminant analysis (PLDA), a generative probabilistic framework that can be used for various classification tasks. In particular, we presented an exact and scalable formulation of this model that significantly reduces the complexity of the model, both at training and test time. In the following chapters, we apply this model, as well as the techniques described in chapter 3, to the tasks of face, speaker and bimodal recognition.

5 Application to Face Recognition

In this chapter we apply the proposed modeling techniques to the task of automatic face recognition. We begin by discussing related work, before describing several face recognition systems based on the techniques presented in chapter 3 and chapter 4. Experiments are conducted on several publicly available databases using well defined evaluation protocols. In particular, we analyze the impact of challenging recording conditions on the performance of the proposed systems.

5.1 Background

Early work on face recognition was conducted in the sixties by Bledsoe, who proposed a *semi-automatic* system [Bledsoe, 1966]. Funded by an unnamed intelligence agency, the idea of his work consists of using computational power to speed up and to improve person identification. Given a large database of photographs of faces (each of them labeled with an identity) and a new photograph of an unknown person, the task is to select a small subset of photographs from the database such that one of them matches the sample of the unknown person. For this purpose, a semi-automatic system was proposed that operates as follows. First, human operators extract a set of features from photographs, such as the coordinates of the center of the pupils, the extremities of the mouth or the corners of the eyes. Next, from these coordinates, a set of twenty distances is computed such as width of mouth distance, pupil to pupil distance, which define a *feature vector*. When building the database, this feature extraction step is performed on each photograph, and features vectors (as well as a label of the identity) are stored in the computer. During the recognition phase, the feature vector from the new photograph (test sample) is extracted and compared to the feature vector of each photograph in the database, yielding a distance for each comparison. Photographs of the database corresponding to minimal distances are returned.

Bledsoe already noticed challenges due to the high variability of faces in head rotation and tilt, lighting intensity and angle, facial expression, aging, etc. In addition, this initial work highlights important steps for face recognition: (1) localizing points of interest, (2) extracting

interesting features and (3) performing classification based on this information. This is directly related to the key components of a classification task, as depicted in fig. 2.1. More specifically, automatic face recognition typically consists of the steps depicted in fig. 5.1.

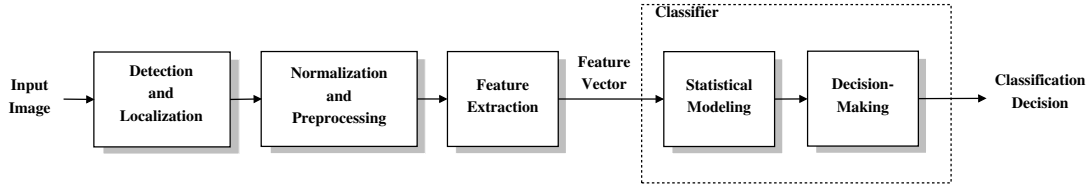


Figure 5.1 – SIMPLIFIED STRUCTURE OF A TYPICAL FACE RECOGNITION SYSTEM.

To our knowledge, the first attempt at building a *fully automatic face recognition system* was performed in the seventies by Kanade [Kanade, 1973, 1977]. At the end of the eighties, [Sirovich and Kirby, 1987, Kirby and Sirovich, 1990] introduced an efficient way to represent pictures of faces using *principal component analysis*. This has led to the popular *Eigenface* representation for recognizing faces proposed in [Turk and Pentland, 1991a,b]. Since then, face recognition has become a mature field and many different techniques have been proposed. Furthermore, researchers typically consider a single step (or a subset of them) of the face recognition toolchain in their work. In the following, we review existing work for each step of the processing chain.

5.1.1 Image Acquisition

The acquisition of images is the first step of a face recognition system. The most common way to acquire such information is by using a camera and recording two-dimensional still images.

Nevertheless, other approaches were also proposed in the literature. Stereo vision cameras can be used to acquire images of a face under different angles. This allows to build a three-dimensional model of a face and to perform matching of three-dimensional structures [Bronstein et al., 2005]. Another possibility is to rely on videos, which provide multiple frames of the person to recognize [Lee et al., 2003].

This work focuses on the classical face recognition task using two-dimensional still images.

5.1.2 Face Detection and Localization

Once the image has been acquired, the coordinates of all the faces, if any, in the image are determined. This process is known as *face detection*. This task is often seen as a binary classification relying on two components. First, a search algorithm extracts subwindows at different locations and scales. Second, these subwindows are fed to a binary classifier that determines whether they contain a face or not. Typically, a clustering algorithm is applied on the positive detections to merge multiple detections.

Several systems have been proposed for detecting faces in images, relying on various classification techniques. For instance, approaches based on density estimation [Moghaddam and Pentland, 1997, Sung and Poggio, 1998], neural networks [Rowley et al., 1996] or a sparse network of linear functions [Yang et al., 1999] were proposed. In particular, boosted classifiers [Freund and Schapire, 1997] became very popular and showed state-of-the-art performance [Viola and Jones, 2001, Fröba and Ernst, 2004, Zhang et al., 2007, Atanasoaei, 2012]. Existing face detection systems are very accurate when dealing with well-separated frontal faces in images with simple backgrounds. But similarly to face recognition, the performance of state-of-the-art algorithms is affected by challenging conditions such as head pose variations or cluttered backgrounds.

Another related task is *face localization*, which assumes that the input contains exactly one face. It aims at refining the detection and/or at accurately locating facial features such as eye centers, the nose tip or mouth corners [Cootes et al., 1995, 2001, Atanasoaei, 2012].

5.1.3 Normalization and Preprocessing

Once a face on an image has been localized, it is commonly converted to grayscale and aligned. This is usually performed by geometrically normalizing the image $\mathfrak{I}$ so that the left and right [1] eyes $\boldsymbol{a}_l^*$ and $\boldsymbol{a}_r^*$ are located at certain positions in the aligned image $\mathfrak{I}^*$:

$$\mathfrak{I}^*(\boldsymbol{p}) = \mathfrak{I}\left(\frac{1}{s}\boldsymbol{Q}_{-\alpha}(\boldsymbol{p}-\boldsymbol{o}^*)+\boldsymbol{o}\right), \tag{5.1}$$

where the scale s and the angle α are computed as:

$$s = \frac{\left\|\boldsymbol{a}_r^*-\boldsymbol{a}_l^*\right\|}{\|\boldsymbol{a}_r-\boldsymbol{a}_l\|}, \qquad \alpha = \arctan\left(\frac{a_{r,y}-a_{l,y}}{a_{r,x}-a_{l,x}}\right), \qquad \boldsymbol{Q}_\alpha = \begin{pmatrix}\cos\alpha & -\sin\alpha\\ \sin\alpha & \cos\alpha\end{pmatrix}, \tag{5.2}$$

with $\boldsymbol{a}_l$ and $\boldsymbol{a}_r$ being the 2D coordinates of the left and right eye in the original image, $\boldsymbol{o}$ and $\boldsymbol{o}^*$ the transformation offsets in the original and the aligned image (usually, the center between the eyes is used in both cases) and $\boldsymbol{Q}_\alpha$ the rotation matrix. After aligning the image to the eye positions, the image is cut to a specific image resolution $\boldsymbol{r} = \left(r_x, r_y\right)^\top$.

Normalized images are often subject to illumination variations. Different illumination conditions can produce very different images of the same subject. In fact, [Adini et al., 1997] showed that image variations due to lighting changes are more significant than those due to different personal identities. One efficient way to reduce the impact of lighting variations is by applying a preprocessing algorithm. Image processing techniques such as histogram equalization [Ramírez-Gutiérrez et al., 2010] and filtering [Shashua and Riklin-Raviv, 2001, Wang et al., 2004, Jobson et al., 1997a,b] are commonly employed. Another strategy is to formalize the problem as an optimization task [Gross and Brajovic, 2003, Chen et al., 2006]. In practice, multistage algorithms offer a good trade-off between performance and accuracy [Tan and

1. Left and right are referred to from the perspective of the subject that is shown in the image.

Triggs, 2010].

5.1.4 Feature Extraction

After image normalization and preprocessing, the concatenation of all pixels of the resulting image could be used to feed a classifier. In practice, computer vision tasks usually extract visual features that contain more robust and relevant information. These features can be extracted at specific points (e.g., the keypoints returned by a facial feature localizer or a regular grid of points), at the block level or densely at the pixel level.

Most of the popular feature extractors have been employed for face recognition in various ways, such as *local binary patterns* (LBP) [Ahonen et al., 2004, 2006] as well as its extension that considers multiple scales [Wang et al., 2009], *Gabor filters* [Wiskott et al., 1997, Zhang et al., 2005, Pinto et al., 2009, Günther et al., 2012a], the *scale-invariant feature transform* (SIFT) [Lowe, 2004, Prince and Elder, 2007], or the *discrete cosine transform* (DCT) [Sanderson and Paliwal, 2003].

Many of these techniques are hand-crafted and researchers tend to combine several visual descriptors to boost the performance of modern face recognition systems. Furthermore, there is an emerging interest in using *deep learning* to obtain a suitable face representation [Chopra et al., 2005].

5.1.5 Modeling and Classification

Different face recognition systems have been engineered during the last decades, using both *discriminative* or *generative* machine learning techniques. Besides, the classifiers are fed either by the whole face region or by local features.

When using the whole face region as a raw input to a classifier, the method is said to be *holistic*. Example of such discriminative approaches are *Eigenfaces* [Turk and Pentland, 1991a,b], *Fisherfaces* [Belhumeur et al., 1997], the *Bayesian intrapersonal/extrapersonal classifier* (BIC) [Moghaddam et al., 1998] and the *support vector machines* (SVM)-based system proposed in [Phillips, 1999].

But local features-based techniques are also popular. They may rely on a simple similarity measure for classification, such as in [Wiskott et al., 1997], which considers Gabor-based features extracted on an elastic graph, or in [Ahonen et al., 2006], which extracts histograms of LBPs. Alternatively, popular classifiers such as multi-layer perceptrons (MLP) with DCT-based features or SVM with Gabor-based features are proposed in [Cardinaux et al., 2003] and [Pinto et al., 2009], respectively.

On the other hand, generative face recognition systems have been conducted using Gaussian mixture models (GMM) [Cardinaux et al., 2003, Lucey and Chen, 2004], Hidden Markov models

[Cardinaux et al., 2004, 2006] and Bayesian networks [Heusch and Marcel, 2007]. More recently, a probabilistic framework relying on subspaces to model variations has been proposed [Prince and Elder, 2007, Li et al., 2012]. In this chapter, we investigate the use of probabilistic models, such as inter-session variability (ISV) modeling, joint factor analysis (JFA), total variability (TV or i-vectors) modeling (see chapter 3) and our scalable formulation of probabilistic linear discriminant analysis (PLDA) (see chapter 4) for the task of face recognition.

5.2 Databases

There are several publicly available databases of facial images to evaluate face recognition systems. The numbers of identities and images in these databases vary from 165 images of 15 persons in the *Yale Face Database*[2] to the extremes of over 750,000 images of 337 identities in the *Multi-PIE* database [Gross et al., 2008] or more than 13,000 images of 5,749 people in *Labeled Faces in the Wild* (LFW) database [Huang et al., 2007b]. Commonly, there is only one face present in each image of the database, and often additional information about the images are provided, like the identity and the gender of the person, the facial expression or the environment conditions the image was taken in. Furthermore, in nearly every database at least the locations of the left and right eye are annotated by hand and sometimes, more annotations like mouth and nose are provided.

Depending on the intended task, the databases contain images captured under different environment conditions. Most databases include images that are taken in strictly frontal pose, so that the effects of facial expressions, strong illuminations, partial occlusions, or human aging processes can be studied, i.e., variations that occur in an access control scenario. Other databases like LFW provide images in a completely unrestricted environment. Therefore, one possible separation between image databases is the way, image variations like illumination, facial expression, occlusion and pose are handled. In *controlled* databases some or all image variations are enforced, while *uncontrolled* databases include images as they would occur in every day life conditions.

In addition, to ensure a fair comparison of face recognition algorithms, image databases are often accompanied with evaluation protocols (see sec. 2.6). All the evaluation protocols employed in this work were made publicly available through the *Python Package Index* (PyPI) [3].

5.2.1 Controlled Databases

Three of the databases, where all image variations are controlled, are Multi-PIE, CAS-PEAL and AR face. Two other databases that we consider to be in the group of controlled image databases are FRGC and GBU. Though the images of the latter databases have (partially) been taken in environments with unrestricted illumination conditions and with some facial expressions, all

2. http://vision.ucsd.edu/content/yale-face-database
3. https://pypi.python.org

faces in the images are not occluded and perfectly frontal, i.e., show no out-of-plane rotation.

CMU Multi-PIE

The *CMU Multi-PIE* database [4] [Gross et al., 2008] consists of 755,370 images shot in four different sessions from 337 subjects. Some samples of this database are shown in fig. 5.2. The Multi-PIE database itself does not provide evaluation protocols, but we generated and published several face verification protocols ourselves. All protocols are split up into a training, a development and an evaluation set, where the identities between the sets are disjoint. The training set is composed of all 208 individuals that did not participate in all four sessions, while the size of development set (64 identities) and evaluation set (65 identities) is almost equal. In this work we use protocols for controlled illumination, expression and pose.

Figure 5.2 – SAMPLES FROM THE MULTI-PIE DATABASE.

Three illumination protocols are defined to analyze the impact of illumination at both enrollment and test time. In the *M* (matched) protocol, model enrollment is performed using images with neutral illumination, and the probe set also consists of images with neutral illumination. The *U* (unmatched) protocol relies on the same enrollment data, but probe images have various illumination conditions. Finally, in the *G* protocol, both enrollment and probe images have various illumination conditions. The same training set is shared by these three protocols and contains images from several sessions with various illumination conditions.

In the expression and pose protocols, model enrollment is performed using images with neutral illumination and expression and frontal pose. The probe sets contain images with either facial expressions (*E*) or non-frontal pose (*P*), similarly to the training sets. When assessing the performance of the systems, results are split by expression or pose.

CAS-PEAL

The *CAS-PEAL* database [5] [Gao et al., 2008] includes 9,031 frontal images [6] (and several non-frontal images, which we do not use due to lack of protocol) from 1,040 Chinese persons. Some samples of this database are shown in fig. 5.3. Using these images six identification protocols

4. http://www.multipie.org
5. http://www.jdl.ac.cn/peal/
6. unlike the number 9,032 incorrectly reported in [Gao et al., 2008]

Protocol	Group	Number of identities	Number of samples (train or test)	Number of trials
P	Train	208	7,775	n.a.
	DEV	64	3,328	212,992
	EVAL	65	3,380	219,700
M	Train	208	9,785	n.a.
	DEV	64	256	16,384
	EVAL	65	260	16,900
U	Train	208	9,785	n.a.
	DEV	64	4,864	311,296
	EVAL	65	4,940	321,100
G	Train	208	9,785	n.a.
	DEV	64	4,864	311,296
	EVAL	65	4,940	321,100
E	Train	208	1,095	n.a.
	DEV	64	576	36,864
	EVAL	65	585	38,025

Table 5.1 – MULTI-PIE EVALUATION PROTOCOLS. *This table depicts the number of identities, samples and trials in Multi-PIE (training set* Train*, development set* DEV *and evaluation set* EVAL*).*

are provided with the database. Each of the protocols tests a different image variation: facial *expression*, non-frontal *lighting*, *accessory*, different *background*, subject-camera *distance* and *aging*.

Figure 5.3 – SAMPLES FROM THE CAS-PEAL DATABASE.

Unconventionally, the training set defined by the CAS-PEAL database consists of 1,200 images that are a random subset of the images of the evaluation set. In each of the protocols all 1,040 neutral and frontally illuminated images serve as model images; models are enrolled from one image per person only. For the probe sets the numbers of images and subjects differ between protocols, a complete list is given in [Gao et al., 2008].

Protocol	Group	Number of identities	Number of samples (train or test)	Number of trials
accessory	Train	300	1,200	n.a.
	DEV	1,040	2,285	2,376,400
aging	Train	300	1,200	n.a.
	DEV	1,040	66	68,640
background	Train	300	1,200	n.a.
	DEV	1,040	553	575,120
distance	Train	300	1,200	n.a.
	DEV	1,040	275	286,000
expression	Train	300	1,200	n.a.
	DEV	1,040	1,570	1,632,800
lighting	Train	300	1,200	n.a.
	DEV	1,040	2,243	2,332,720

Table 5.2 – CAS-PEAL EVALUATION PROTOCOLS. *This table depicts the number of identities, samples and trials in CAS-PEAL (training set* Train *and development set* DEV*).*

AR face Database

The *AR face database*[7] [Martínez and Benavente, 1998] contains 3,312 images[8] from 76 male and 60 female clients taken in two sessions. Facial images in this database include three variations: facial expressions, strong controlled illumination and occlusions with sunglasses and scarfs. Some samples of this database are shown in fig. 5.4.

Figure 5.4 – SAMPLES FROM THE AR FACE DATABASE.

Several verification protocols were defined for this database, splitting up the identities into 50 training subjects (28 men and 22 women) and each 43 clients (24 male and 19 female) in the development and evaluation set. For model enrollment we use those two images per client that have neutral illumination, neutral expression and no occlusion. The protocols *illumination, occlusion, occlusion_and_illumination* test the specific image variations that are defined in the database, i.e., probe images have either non-frontal illumination, partially occluded faces, or both occlusion and illumination.

7. http://www2.ece.ohio-state.edu/~aleix/ARdatabase.html
8. The official website reports more than 4,000 images, but we could not reach the controller of the database to clarify the difference.

Protocol	Group	Number of identities	Number of samples (train or test)	Number of trials
illumination	Train	50	1,076	n.a.
	DEV	43	258	11,094
	EVAL	43	258	11,094
occlusion	Train	50	1,076	n.a.
	DEV	43	172	7,396
	EVAL	43	172	7,396
occlusion_and_illumination	Train	50	1,076	n.a.
	DEV	43	344	14,792
	EVAL	43	344	14,792

Table 5.3 – AR FACE EVALUATION PROTOCOLS. *This table depicts the number of identities, samples and trials in AR face (training set* Train*, development set* DEV *and evaluation set* EVAL*).*

Face Recognition Grand Challenge (FRGC)

The *Face Recognition Grand Challenge*[9] (FRGC) database in its version ver2.0 contains 36,818 high resolution images of 466 clients that were collected in various sessions during four years. Additionally, the database also includes 3D image data, but these are not used in this work. Some samples of this database are shown in fig. 5.5. The database provides several biased protocols [Phillips et al., 2005] (named *experiments* by the authors), three of which utilize 2D image data only: experiments *2.0.1*, *2.0.2* and *2.0.4*.

Figure 5.5 – SAMPLES FROM THE FRGC DATABASE.

Experiments *2.0.1* and *2.0.2* compare only controlled images that were taken in a studio environment. While experiment *2.0.1* provides one image to enroll a client model, experiment *2.0.2* enrolls a client model from four images. Similarly, in experiment *2.0.1* a score is computed by comparing a client model with a single probe, whereas for experiment *2.0.2* four probe images per person are integrated to build a single score. Finally, experiment *2.0.4* uses the same client models as experiment *2.0.1*, but probe images that were taken in a corridor or outdoors with unconstrained illumination conditions.

For each experiment the protocols define different *masks* (sub-protocols), which specify pairs {model/probe} that should be taken to evaluate. In our experiments we use the most difficult *mask III*, throughout. The training set, which is identical for the three experiments, contains

9. http://face.nist.gov/frgc/

12,776 studio and corridor images from 266 clients. The clients of the training set form a subset of the test clients, making these protocols biased.

Protocol	Group	Number of identities	Number of samples (train or test)	Number of trials
2.0.1	Train	222	12,776	n.a.
	DEV	466	8,456	64,028,832
2.0.2	Train	222	12,776	n.a.
	DEV	466	8,456	4,001,802
2.0.4	Train	222	12,776	n.a.
	DEV	466	4,228	32,014,416

Table 5.4 – FRGC EVALUATION PROTOCOLS. *This table depicts the number of identities, samples and trials in FRGC (training set* Train *and development set* DEV*).*

The Good, the Bad & the Ugly (GBU)

The Good, the Bad and the Ugly[10] (GBU) database [Phillips et al., 2011] is built from 8,638 high resolution frontal outdoor images of 782 clients. Some samples of this database are shown in fig. 5.6. It defines the three protocols *Good*, *Bad* and *Ugly*, which specify image pairs that should be compared. Each protocol includes 1,085 different images that are used to enroll client models — each model is enrolled from a single image and there exists several client models per identity. Likely, 1,085 probe images are defined by each protocol and all models are compared with all probes to compute the final ROC curves. Additionally, four different training sets are present; we take the largest set *x8* in all our experiments on the GBU database, unlike [Phillips et al., 2011], who used the *x2* training set to train their baseline algorithm.

Figure 5.6 – SAMPLES FROM THE GBU DATABASE.

5.2.2 Uncontrolled Databases

Since we do not want to restrict the applications to use controlled image data, we also investigate two challenging databases that contain images captured under completely uncontrolled conditions as they would appear in several real world scenarios.

10. http://www.nist.gov/itl/iad/ig/focs.cfm

Protocol	Group	Number of identities	Number of samples (train or test)	Number of trials
Good	Train (*x8*)	345	1,766	n.a.
	DEV	437	1,085	1,177,225
Bad	Train (*x8*)	345	1,776	n.a.
	DEV	437	1,085	1,177,225
Ugly	Train (*x8*)	345	1,776	n.a.
	DEV	437	1,085	1,177,225

Table 5.5 – GBU EVALUATION PROTOCOLS. *This table depicts the number of identities, samples and trials in GBU (training set* Train *and development set* DEV*).*

BANCA

The first uncontrolled database we explore is *BANCA*[11] [Bailly-Baillière et al., 2003]. Originally, it captures video and audio recordings of 208 persons that utter prompted sequences in one among four European languages. Recordings were taken in twelve different sessions, where in each session every subject generated two videos, one true claimant access and one informed impostor access. From each of these videos, five images and one audio signal was extracted. Some samples of this database are shown in fig. 5.7. However, only the English language was made available [Bailly-Baillière et al., 2003], which consists of 52 persons. Therefore, in this work, we use only the images of the BANCA English database.

Figure 5.7 – SAMPLES FROM THE BANCA DATABASE.

Several *open set* verification protocols are proposed with the database [Bailly-Baillière et al., 2003]. We here take only one of the most challenging protocols *P*, which enrolls client models on five *controlled* images, but probes the system with *controlled*, *degraded* and *adverse* images (for details see [Bailly-Baillière et al., 2003]). Two particularities of this database are that it is small, e.g., the training set consists of only 300 images and that the numbers of 2,340 client and 3,120 impostor scores are almost balanced.

11. http://www.ee.surrey.ac.uk/CVSSP/banca/

Protocol	Group	Number of identities	Number of samples (train or test)	Number of trials
P	Train	30	300	n.a.
	DEV	26	2,730	2,730
	EVAL	26	2,730	2,730

Table 5.6 – BANCA EVALUATION PROTOCOLS. *This table depicts the number of identities, samples and trials in BANCA (training set* Train*, development set* DEV *and evaluation set* EVAL*).*

Labeled Faces in the Wild (LFW)

One of the most popular image databases is the *Labeled Faces in the Wild*[12] (LFW) database [Huang et al., 2007b]. This database contains 13,233 face images from 5,749 celebrities, which were downloaded from the internet, labeled with the name of the celebrity that is shown in the image and cropped by a face detector. In this work we use the images aligned by the funneling algorithm [Huang et al., 2007a]. Some samples of this database are shown in fig. 5.8. The database itself does not provide the eye locations for the images, but we rely on the publicly available[13] annotations of [Guillaumin et al., 2009]. They consist of nine facial feature points (mouth corners, eyes corners and nose) obtained by a facial feature detector [Everingham et al., 2006] after alignment with the funneling algorithm [Huang et al., 2007a]. The locations of the eye centers are estimated by computing the midpoint of the eye corners.

Figure 5.8 – SAMPLES FROM THE LFW DATABASE. *This figure shows samples from the LFW database, after alignment with the funneling algorithm [Huang et al., 2007a].*

The particularity of the LFW database is that it specifies pairs of images, for which a score should be computed, equally distributed over client and impostor pairs. In our case we always choose the first image of the pair for model enrollment and the second image as probe. For the training sets LFW permits two alternatives: *image-restricted* defining specific image pairs that might be used for training, and *unrestricted* using all images of the training subjects. Here, we chose the *unrestricted* setup since several algorithms need to know the identity information of the training images, which is forbidden to be used in the *image-restricted* training setup.

The LFW database is split into two so-called *views*. Since *view 1* is considered to optimize algorithm configurations we only use *view 2* to report the final results. In *view 2* the subjects

12. http://vis-www.cs.umass.edu/lfw/
13. http://lear.inrialpes.fr/people/guillaumin/data.php

are split into ten different *folds*. Each fold contains 300 intrapersonal (same subject) and 300 extrapersonal (different subjects) image pairs, for which a *verification rate* is computed, which is equal to $1 - \frac{|\text{FA}|+|\text{FR}|}{|\text{trials}|}$. In our implementation of the *view 2* protocol, for each of the ten experiments seven folds are used for training, while two folds build the *development set*, from which the threshold θ is estimated and the last fold is employed to compute the classification rate of this fold. Finally, the mean and the standard deviation of the classification successes over all ten experiments is reported [Huang et al., 2007b].

In addition, we define an identification protocol (*P0*) on this database. Identities containing only one sample are first removed. Next, three splits for training, development and evaluation are defined. The training set consists of 280 identities, which all have at least 10 samples, to be able to accurately model within-class variations. The remaining identities were split as follows: 400 for the development set and 1,000 for the evaluation set. Model enrollment is performed using the first image labeled as 0001 of each client. This finally results in a challenging identification protocol (1,000-class problem) using images of faces *in the wild*.

Protocol	**Group**	**Number of identities**	**Number of samples** (train or test)	**Number of trials**
Verification (per fold)	Train	~ 4,000	~ 9,000	n.a.
	DEV	~ 1,100	~ 1,500	1,200
	EVAL	~ 600	~ 800	600
Identification	Train	280	4,613	n.a.
	DEV	400	854	341,600
	EVAL	1,000	2,297	2,297,000

Table 5.7 – LFW EVALUATION PROTOCOLS. *This table depicts the number of identities, samples and trials in LFW (training set* Train*, development set* DEV *and evaluation set* EVAL*).*

5.3 Systems Description

Overall we evaluate nine face recognition systems. These systems are described in the remainder of this section, several of them sharing the same preprocessing and/or feature extraction step. In particular, the cropping of faces is performed using the annotations of the eye centers, which are provided by most of the databases. A summary of these systems is provided in tab. 5.8 and tab. 5.9.

5.3.1 Baseline Systems

We consider a set of three baseline systems for comparison purposes.

One of them is the popular *Eigenfaces* method [Turk and Pentland, 1991a,b], that we refer to as **PCA** in the following. After geometrically normalizing images to a resolution $\boldsymbol{r} = (64, 80)^\top$ (see sec. 5.1.3), the multistage preprocessing algorithm introduced in [Tan and Triggs, 2010] is applied. Next, images are linearized into large vectors, and PCA is applied using the singular

	PCA	LRPCA	LDA-IR
Normalization	$\boldsymbol{r} = (64, 80)^\top$	$\boldsymbol{r} = (80, 80)^\top$	$\boldsymbol{r} = (65, 75)^\top$
Preprocessing	***Multistage algorithm*** [Tan and Triggs, 2010]	***Self-quotient image*** [Wang et al., 2004]	2 color channels
Feature Extraction	Raw pixels (Vectorization)		
Modeling	***PCA*** • $D_{\mathbf{PCA}} = 5120$ $d_{\cos}$	***PCA*** • for 14 regions) • $D_{\mathbf{PCA}} = 14 \times 250 = 3,500$ $d_{\text{Euclidean}}$	***LDA*** (for each color channel) • $D_{\mathbf{PCA}} = 150$ each sum of $d_{\text{Euclidean}}$

Table 5.8 – DESCRIPTION OF THE FACE RECOGNITION SYSTEMS (PART 1).

value decomposition approach to learn a projection matrix $\boldsymbol{W}_{\mathbf{PCA}}$ (see sec. 2.1.1). The projection matrix is then applied to the vectors, retaining 100% of the variance (5,120 coefficients), and compared using a distance measure. We choose the cosine similarity measure h_{cosine} (see eq. (3.59)), since our preliminary work suggests that it performs better than the Euclidean distance.

The two other baseline systems are taken from the *CSU Face Recognition Resources*, [14] which provide the baseline algorithms for *the Good, the Bad & the Ugly* (GBU) challenge [Phillips et al., 2011, Lui et al., 2012]. We rely on the source code provided by the Colorado State University (CSU), and use the parameterizations of the algorithms that were optimized for the GBU database.

The first CSU baseline system is *local region PCA* (**LRPCA**), which computes PCA subspaces for several local regions of the face such as the eyes, the nose and the mouth [Phillips et al., 2011]. After a geometric normalization to a resolution $\boldsymbol{r} = (80, 80)^\top$, thirteen partially overlapping local regions as well as the complete face chip are extracted. Next, all these regions are preprocessed using the self-quotient image algorithm [Wang et al., 2004], before being normalized to zero mean and unit variance. At training time, PCA is applied to each region, before retaining the 3$^{\text{rd}}$ through 252$^{\text{th}}$ eigenvectors. Besides, a normalization and weighting schemes are applied to each dimension. A face is finally encoded by concatenating the 250 projected coefficients for each of the fourteen regions into a new vector of dimensionality 3,500. The comparison relies on the Pearson's correlation coefficient between pairs of images.

The second CSU baseline system **LDA-IR** has the particularity of exploiting color information, in contrast to all the other systems evaluated that solely rely on grayscale information [Lui et al., 2012]. Images are first geometrically normalized to a resolution $\boldsymbol{r} = (75, 65)^\top$ before extracting and processing information from two color channels. The red channel from the RGB color space is extracted, since it brings robustness to mild lighting variations, while the I chrominance from the YIQ color space is employed to compensate for more severe illumi-

14. http://www.cs.colostate.edu/facerec/algorithms/baselines2011.php

<table>
<tr><th></th><th>GMM</th><th>ISV</th><th>JFA</th></tr>
<tr><td>Normalization</td><td colspan="3">$\boldsymbol{r} = (64, 80)^\top$</td></tr>
<tr><td>Preprocessing</td><td colspan="3">***Multistage algorithm***
[Tan and Triggs, 2010]</td></tr>
<tr><td>Feature Extraction</td><td colspan="3">***DCT features***
• by block
• $(B_x \times B_y) = (12 \times 12)$ pixels
• Full overlap of 11 pixels)</td></tr>
<tr><td rowspan="3">Modeling</td><td colspan="3">***UBM GMM***
• 512 components with diagonal covariance
• 50 k-means iterations
• 50 EM iterations</td></tr>
<tr><td>***GMM MAP***</td><td>***ISV***
• Relevance factor $\tau = 4$
•$D_{\boldsymbol{U}} \in \{2, 5, 10, 20, 50, 100, 200\}$
•10 EM iterations</td><td>***JFA***
•$D_{\boldsymbol{U}} = D_{\boldsymbol{V}} \in \{2, 5, 10, 20, 50, 100\}$
•10 EM iterations</td></tr>
<tr><td colspan="3"></td></tr>
<tr><th></th><th>TV-Cosine</th><th>TV-PLDA</th><th>SIFT-PLDA</th></tr>
<tr><td>Normalization</td><td colspan="2">$\boldsymbol{r} = (64, 80)^\top$</td><td>9 keypoints
(IPD=50 pixels)</td></tr>
<tr><td>Preprocessing</td><td colspan="2">***Multistage algorithm***
[Tan and Triggs, 2010]</td><td>None</td></tr>
<tr><td>Feature Extraction</td><td colspan="2">***DCT features***
• by block
• $(B_x \times B_y) = (12 \times 12)$ pixels
• Full overlap of 11 pixels)</td><td>***SIFT***
• 3 scales</td></tr>
<tr><td rowspan="3">Modeling</td><td colspan="2">***UBM GMM***
• 512 components with diagonal covariance
• 50 k-means iterations
• 50 EM iterations</td><td rowspan="2">***PCA***
•$D_{\mathbf{PCA}} = 200$</td></tr>
<tr><td colspan="2">***Total Variability***
• $D_{\boldsymbol{T}} = 400$
• 25 EM iterations
• Whitening
• WCCN</td></tr>
<tr><td>$d_{\cos}$</td><td colspan="2">***PLDA***
• $D_{\boldsymbol{F}} = D_{\boldsymbol{G}} \in \{2, 5, 10, 20, 30, 40, 50, 60\}$
• 200 EM iterations</td></tr>
</table>

Table 5.9 – DESCRIPTION OF THE FACE RECOGNITION SYSTEMS (PART 2).

nation variations. Besides, a normalization to zero mean and unit variance (using statistics computed on the training set) is performed on both channels, separately. PCA is then applied to dimensionally reduce the feature space, retaining 98% of the energy. Next, LDA is employed such that the dimension of the resulting feature vectors is the minimum of 128 and (the number of classes in the training set minus one), again, separately to each color channel. Finally, feature vector comparisons are performed based on the Euclidean distance, summing the distances obtained in each color channel.

5.3.2 SIFT-PLDA System

A PLDA system fed by SIFT descriptors (**SIFT-PLDA**) is designed, similarly to [Li et al., 2012]. In contrast to [Li et al., 2012] that relies on SIFT descriptors for the LFW database distributed [15] by [Guillaumin et al., 2009], we define our own feature extraction scheme to be able to evaluate the system on any database. To avoid the use of a facial feature localizer, we extract SIFT descriptors on a fixed position grid of nine keypoints that is positioned according to the eye center coordinates. This grid of keypoints was defined based on statistical measurements computed on the automatic annotations of facial fiducial points distributed by [Guillaumin et al., 2009]. These statistics were estimated on the training set of the *View1* protocol of LFW so that the resulting grid consists of nine keypoints roughly located on eye corners, mouth corners and nose tip. Prior to the keypoint extraction, the image is normalized so that the interpupillary distance (IPD) remains constant at 50 pixels. SIFT descriptors [Lowe, 2004], each of dimensionality 128, are then extracted on these nine keypoints at three different scales, resulting in a feature vector consisting of $3,456$ coefficients.

Next, we reduce the dimensionality of these feature vectors using PCA and retain the 200 dimensions with the highest eigenvalues. PLDA is finally applied as introduced in chapter 4. At training time, the subspaces $\boldsymbol{F}$ and $\boldsymbol{G}$ are initialized based on the result of singular value decomposition on the between-class and within-class scatter of the training data, respectively. Each basis vector is normalized by the eigenvalue to ensure that the latent variables have unit variance. We initialize the covariance matrix $\boldsymbol{\Sigma}$ to be the covariance of the training data. We always perform 200 rounds of EM training as suggested by our preliminary work. The dimensionality of the subspaces $\boldsymbol{F}$ and $\boldsymbol{G}$ is optimized on the development set (if any), using a grid search approach. To restrict the search space, we consider $D_{\boldsymbol{F}} = D_{\boldsymbol{G}} = \nu$, with $\nu \in \{2,5,10,20,30,40,50,60\}$ under the constraint $\nu <$ |training classes|.

5.3.3 GMM-based Systems

Five systems are GMM-based and rely on the techniques described in chapter 3 and chapter 4, respectively **GMM**, **ISV**, **JFA**, **TV-Cosine** and **TV-PLDA**.

These systems all rely on the same Universal Background Model and they hence employ the same normalization, preprocessing and feature extraction techniques.

First, they share the same normalization and preprocessing step as **PCA**, geometrically normalizing images to a resolution $\boldsymbol{r} = (64,80)^\top$ before applying the multistage preprocessing algorithm introduced in [Tan and Triggs, 2010].

Next, the feature extraction is performed as follows. The preprocessed image is decomposed into a set of K overlapping blocks, taken on a regular grid of blocks of size $B_x \times B_y$. In practice, we use a block size $B_x = B_y = 12$ pixels with a full overlap, leading to $K = 3,657$ blocks per

15. http://lear.inrialpes.fr/people/guillaumin/data.php

image.

A feature vector is extracted from each block. These K feature vectors are considered as observations of the same signal (the same face). This spatial decomposition of the preprocessed image is a key aspect to be able to apply the modeling techniques described in this chapter since they aim at modeling the resulting observations in a generative way. Theoretically, any visual descriptor could be used as a representation of a block. In practice, the most common approach relies on the *2D discrete cosine transform* (2D-DCT) and consists of extracting the lowest frequency 2D-DCT coefficients [Sanderson and Paliwal, 2003, Lucey and Chen, 2004, Cardinaux et al., 2006, Wallace et al., 2011] since they are less susceptible to noise. This leads to a lower $D_{\boldsymbol{o}}$-dimensional representation of a block ($D_{\boldsymbol{o}} < B_x B_y$), in a similar way as the compression performed by the JPEG file format [ITU, 1993]. $D_{\boldsymbol{o}}$ is set equal to 44 in the following experiments. An overview of this feature extraction process, commonly called the parts-based approach, is shown in fig. 5.9.

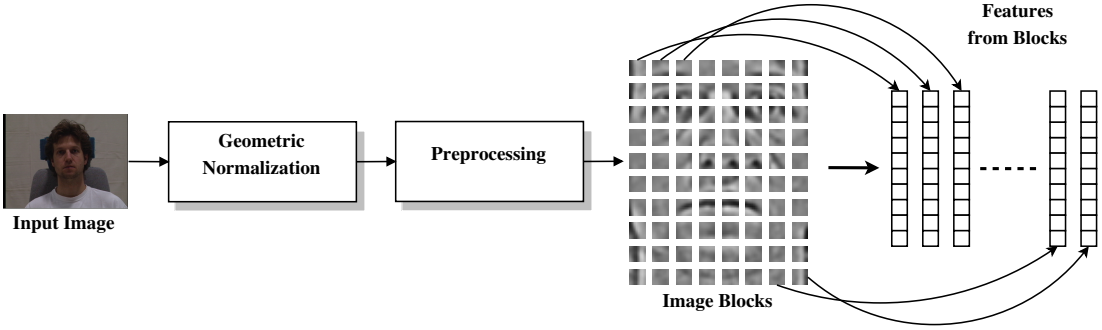


Figure 5.9 – PARTS-BASED FEATURE EXTRACTION. *This figure shows the parts-based feature extraction approach, which decomposes the image of a face into a set of overlapping blocks, before extracting a feature vector from each block.*

In addition, pre- and post-normalization [Wallace et al., 2012] are commonly applied to improve the robustness to session variability:

– Prior to the feature extraction, the pixel values in each block are normalized to zero mean and unit variance. In this case, the zeroth-order DCT coefficient is discarded, as the previous normalization makes it vanish.
– After feature extraction, the resulting 2D-DCT feature vectors are normalized to zero mean and unit variance in each dimension with respect to the other feature vectors of the image.

Hence, the first pre-normalization is performed on a per-block basis, whereas the post-normalization is achieved on a per-sample basis.

Finally, if an image is decomposed into a set of $K = 3,657$ blocks, its final representation is a set of $K = 3,657$ feature vectors, $\mathbb{O} = \{\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_K\}$, each of dimensionality $D_{\boldsymbol{o}} = 44$.

At training time, we first derive the UBM, before training the subspaces $\boldsymbol{U}$, $\boldsymbol{V}$ or $\boldsymbol{T}$ of the **ISV**, **JFA** or **TV** systems. UBMs with 512 components and diagonal covariance matrices are trained using 50 iterations of k-means followed by 50 EM iterations.

Next, subspaces are learned using the EM algorithm. For **ISV**, the $\boldsymbol{U}$ subspace is trained using 10 iterations and its dimensionality is optimized on the development set (if any), considering $D_{\boldsymbol{U}} \in \{2,5,10,20,50,100,200\}$. For **JFA**, the $\boldsymbol{U}$, $\boldsymbol{V}$ and $\boldsymbol{D}$ subspaces are trained using 10 iterations, and the dimensionality of $\boldsymbol{U}$ and $\boldsymbol{V}$ is optimized in the same way, considering $D_{\boldsymbol{U}} = D_{\boldsymbol{V}} = \nu$, with $\nu \in \{2,5,10,20,50,100\}$ under the constraint $\nu <$ |training classes|. For the two **TV** systems, the total variability space $\boldsymbol{T}$ is first trained using 25 iterations. Based on prior work, we set the dimensionality of this subspace equal to 400, except for the BANCA database, on which we use 200 since its training set consists of only 300 samples. For **TV-PLDA**, PLDA is applied in the same way as **SIFT-PLDA**, using 200 rounds of EM training (see sec. 5.3.2).

For the **GMM** and **ISV** systems, a relevance factor $\tau = 4$ is used for client model adaptation.

5.4 Experimental Results

In the following, we evaluate the face recognition systems introduced in the previous section on several databases. The techniques employed by these recognition systems were implemented in Bob [Anjos et al., 2012], an open source toolkit for machine learning and signal processing developed during my thesis (see appendix A). In addition, all results and plots reported in this section can be easily regenerated using the satellite package [16] that accompanies this dissertation (see sec. A.3.2).

5.4.1 Face Variations

First, we employ controlled databases to separately estimate the impact of pose, illumination and expression (PIE) variations as well as occlusions on face recognition systems. The dimensionality of the subspaces is tuned on the development set, before reporting results on the evaluation set.

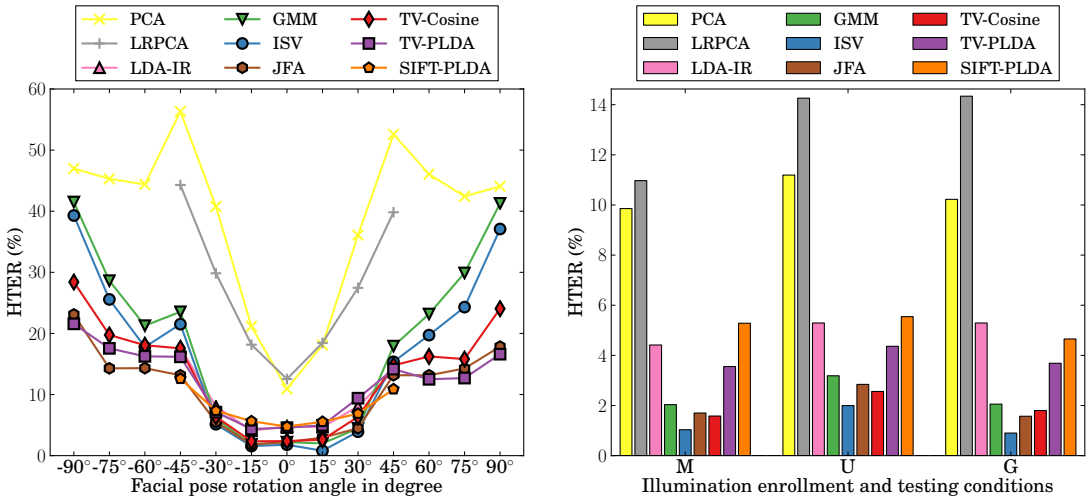
Face Poses

We start by addressing the issue of non-frontal pose, which is of particular interest in surveillance applications. To test how the systems perform on non-frontal images, we execute them on the protocol *P* of the Multi-PIE database. Model enrollment is performed using only frontal images, while probe samples are taken from left profile to right profile in steps of 15°.

By default, the image alignment step uses the eye positions as long as both eyes are visible in the image, i.e., for images with a rotation less or equal to ±45°. In the profile and near-profile cases, we adopt a different cropping strategy, aligning images according to the eye and mouth positions. The final positions are $\boldsymbol{a}_e^* = (25,16)^\top$ and $\boldsymbol{a}_m^* = (25,52)^\top$ for eye and mouth in the left profile images, and $\boldsymbol{a}_e^* = (38,16)^\top$ and $\boldsymbol{a}_m^* = (38,52)^\top$ in the right profile. We choose these positions to assure the face including the nose tip to be inside the image, while keeping most

16. https://pypi.python.org/pypi/xbob.thesis.elshafey2014

of the background outside.



(a) Effect of pose

(b) Effect of illumination

Figure 5.10 – PERFORMANCE OF THE SYSTEMS FOR VARIOUS POSES AND ILLUMINATION CONDITIONS ON MULTI-PIE. *This figure shows the* HTER *on the evaluation set of Multi-PIE, on protocol* P *for several poses (a) and on the three illumination protocols* M, U *and* G *(b).*

In fig. 5.10(a), the HTER on the evaluation set is plotted for each of the tested poses, independently. It can be observed that close-to-frontal poses up to ±15° can be handled by most systems. On the other hand, starting from ±45°, the systems are impacted by the pose.

Comparing the systems, two baselines **PCA** and **LRPCA** are significantly worse than the other systems, and their performance is comparable to chance starting from ±45°. For close-to-frontal poses, **ISV** slightly outperforms the other systems, whereas for profile and near-profile poses, **JFA** and **TV-PLDA** are the most accurate systems. In particular, **TV-PLDA** leads to a relative improvement in HTER of 54% on profile faces when compared to **GMM**, which is performing well on frontal faces.

As shown in fig. 5.10(a), we did not run the evaluation on profile and near-profile cases for the CSU baselines **LDA-IR** and **LRPCA** since their implementation does not allow to crop faces without providing the locations of the two eyes. This is also the case for **SIFT-PLDA** as the keypoints would not be correctly located on the face in this situation.

Illumination

Another issue in automatic face recognition is uncontrolled illumination.

First, we evaluate the systems on the three illumination protocols of Multi-PIE, which all share the same training set containing images with various illumination conditions. The HTER on

the evaluation set is plotted for each of the systems on fig. 5.10(b).

Comparing the protocols *M* and *U*, which rely on the same enrollment data, provides a way to measure the impact of illumination (see sec. 5.2.1). The only difference is the use of additional probe samples with non-frontal illumination in the *U* protocol. As expected, this negatively impacts all the systems, the error rates increasing up to a factor of two.

Looking at the protocol *G*, which defines the same probe samples as *U*, we can observe the impact of using more enrollment samples with non-frontal illumination. Interestingly, several algorithms such as **ISV**, **JFA** and **SIFT-PLDA** obtain results that are comparable to the ones of the *M* protocol, sometimes even better. This suggests that one possible solution to deal with uncontrolled illumination is to require enrollment samples taken under various illumination conditions.

Comparing the systems, **ISV** is the best performing on all theses protocols, whereas **PCA** is severely impacted by illumination conditions. **JFA**, **TV-Cosine** and **GMM** also offer reasonably good performances. Comparing **ISV** to **GMM**, we observe a relative improvement in HTER of 50% and 56% on protocol *M* and *G*, respectively.

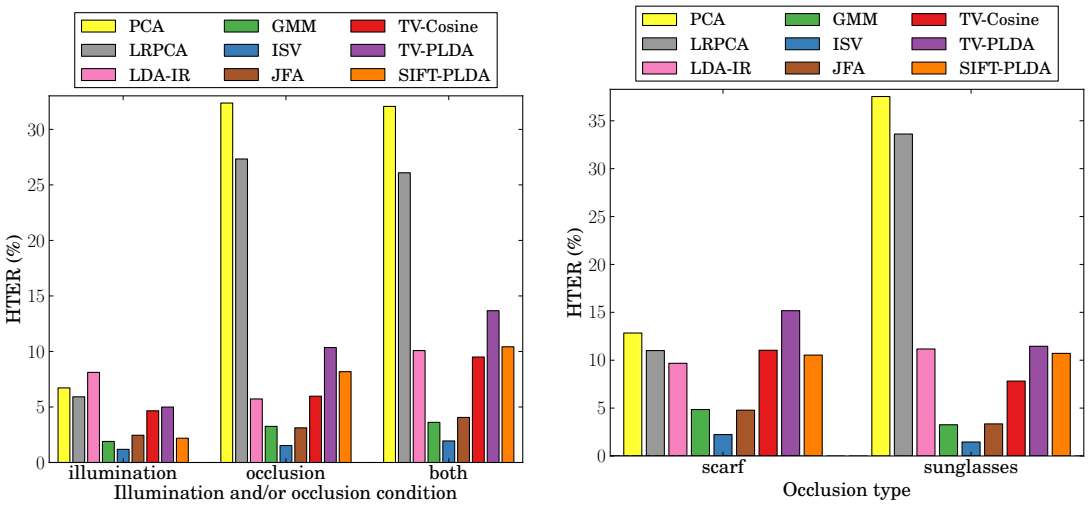
Next, we consider the AR face database to evaluate again the impact of illumination. Results (HTER) on the evaluation set are depicted on the left side of fig. 5.11(a). Similar trends as on Multi-PIE can be observed, **ISV** being again the best performing system, followed by **GMM**, **JFA** and **SIFT-PLDA**. However, **LDA-IR** as well as the **TV**-based systems are less accurate on this database. In the case of the **TV**-based systems, a possible explanation for this degradation is the smaller size of the training set (1,076 images versus 9,785 for Multi-PIE).

Facial Expressions

Another aspect an automatic face recognition system must deal with is facial expression. To test the algorithms against various expressions we rely on the protocol *E* of the Multi-PIE database.

The results of the expression experiment are shown in fig. 5.12. While neutral faces are recognized quite well by all algorithms, other expressions influence most of the algorithms, sometimes severely. A notable exception is **SIFT-PLDA**, whose performance does not vary much according to the different expressions. However, this is not the best performing system.

Overall, the lowest error rate is obtained with **ISV**, except for the *disgust* and *surprise* expression, where **JFA** and **TV-Cosine** are slightly better, respectively. In contrast, the **PCA** and **LRPCA** baselines, which are pixel-based, are unable to cope with extreme expressions like *disgust* and *scream*. Besides, **TV**-based systems do not perform well on this protocol. Once more, this might be explained by the relatively small size of the training set (1,095 images with various expressions).



(a) Effect of illumination and occlusion

(b) Effect of different occlusion types

Figure 5.11 – PERFORMANCE OF THE SYSTEMS ON AR FACE. *This figure shows the* HTER *on the evaluation set of AR face considering both illumination conditions and occlusions (a) and occlusions only (b).*

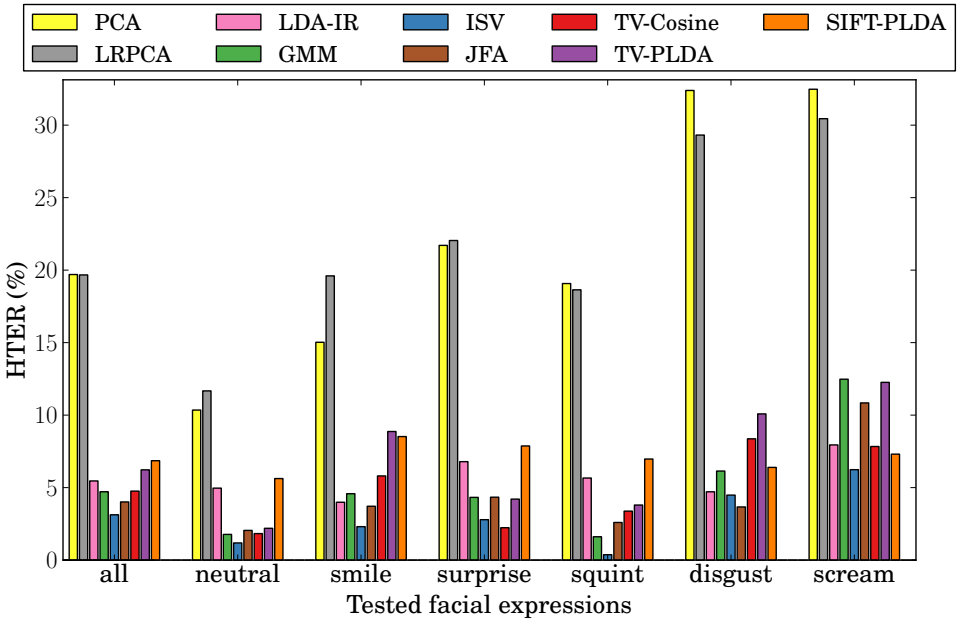


Figure 5.12 – PERFORMANCE OF THE SYSTEMS FOR VARIOUS EXPRESSIONS ON MULTI-PIE. *This figure shows the* HTER *on the evaluation set of Multi-PIE, on protocol* E *(expressions).*

Occlusions

In several real world scenarios, the faces captured by cameras are partially occluded, which typically affects the performance of face recognition systems. Two prominent occlusions are sunglasses, which hide the key identity information located in the eye region [Sinha et al., 2006], and scarfs covering the lower part of the face during winter. One database that tests exactly these two types of occlusions is the AR face database with its protocol *occlusion*.

Fig. 5.11(a) contains the results of the occlusion experiments. As a baseline for this database we selected the protocol *illumination*, on which all algorithms perform reasonably well. When occlusions come into play, all algorithms suffer a drop in performance, independent of whether there is additional non-frontal illumination. The most robust system is **ISV**, while **PCA** and **LRPCA** perform again poorly.

Having a closer look by separating between the two occlusion types (cf. fig. 5.11(b)), scarfs and sunglasses seem to have different impacts on the systems. Raw pixels algorithms like **PCA** and **LRPCA** are unable to handle sunglasses. In contrast, the other systems perform comparatively well on both types of occlusions.

Conclusions

In these experiments, we evaluated the impact of face variations. Considering the best systems, the HTER is typically in the range [1%, 10%] in presence of occlusions, expressions or illumination variations, compared to around 20% for profile poses. This suggests that pose variation remains one of the most challenging problems in automatic face recognition.

5.4.2 Experiments on other Databases

After separately analyzing the impact of several face variations, we now execute the face recognition systems on other publicly available face databases. The evaluation is performed following the evaluation protocols and measures shipped with the databases, if any. Specifically, we here include those databases that do not provide separate development and evaluation sets. When a development set is available, the dimensionality of the subspaces is tuned on this set before reporting results on the evaluation set. Otherwise, we employ reasonable values chosen a priori, based on the training set size.

BANCA

Considering the protocol *P* of BANCA English, the HTER on the evaluation set is reported on fig. 5.13. BANCA is a fairly old database, and one of its main limitations is the rather small training set (300 images from 30 subjects). The best performing system is **ISV**, followed by **GMM** and **JFA**. In contrast **TV**-based systems and **SIFT-PLDA** are not accurate on this database.

This suggests once more that a larger training set is required for these approaches.

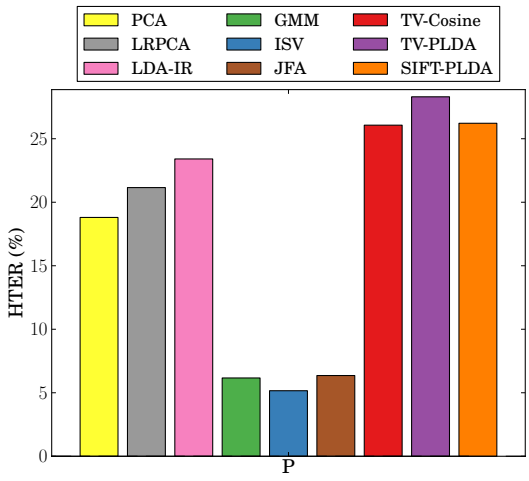


Figure 5.13 – PERFORMANCE OF THE SYSTEMS ON BANCA ENGLISH. *This figure shows the* HTER *on the evaluation set of the protocol* P *of BANCA English.*

CAS-PEAL

On the CAS-PEAL database, identification protocols have been initially defined, and results are reported in terms of recognition rates. The results of this evaluation are given in fig. 5.14, where we also include results of the *Gabor feature-based PCA+LDA* (GPCA+LDA) [17] algorithm [Gao et al., 2004]. Since the images provided by the CAS-PEAL database are grayscale only, the **LDA-IR** system cannot be run.

When comparing the systems, the trend is similar on all the protocols. For the simple variations of *background* and *distance*, several systems work close to perfect, except for **PCA**, **LRPCA**, and **TV-PLDA**. It was indeed noticed previously that **TV-PLDA** requires a training set large enough, whereas it only consists of 1,200 images in the CAS-PEAL database. Considering *Aging*, facial *expressions* and *accessories*, nearly all systems drop performance, but **ISV** and **TV-Cosine** are still stable against these variations.

The most severe problem on this database is the change of illumination. In protocol *lighting* of the CAS-PEAL database not only the directions of light sources are varied, but also the light type changes from ambient light in enrollment images to fluorescent or incandescent light in probe images. This explains the dramatic drop of performance of all systems. Still, ISV outperforms the GPCA+LDA baseline.

17. In [Gao et al., 2008] the CAS-PEAL database organizers propose to use LGBPHS [Zhang et al., 2005], which seems to work better than GPCA+LDA, but they do not provide exact numbers for the experimental results.

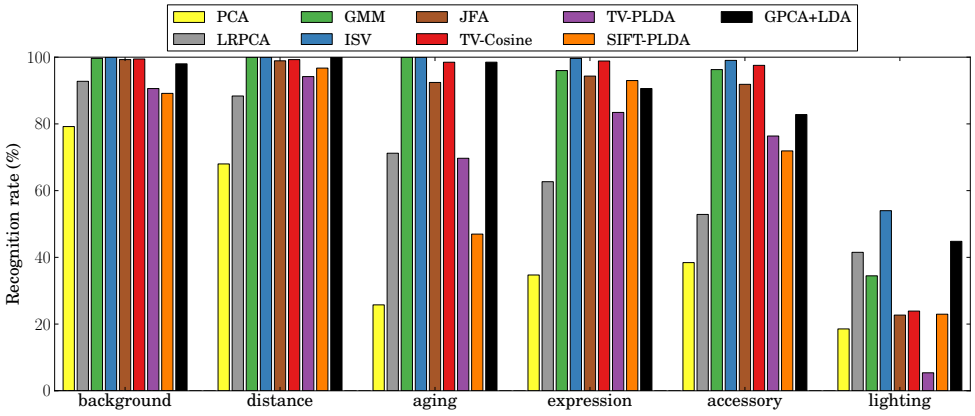


Figure 5.14 – PERFORMANCE OF THE SYSTEMS ON CAS-PEAL. *This figure shows the recognition rate on the CAS-PEAL protocols for both the tested face recognition systems and the GPCA+LDA baseline of [Gao et al., 2004].*

Face Recognition Grand Challenge (FRGC)

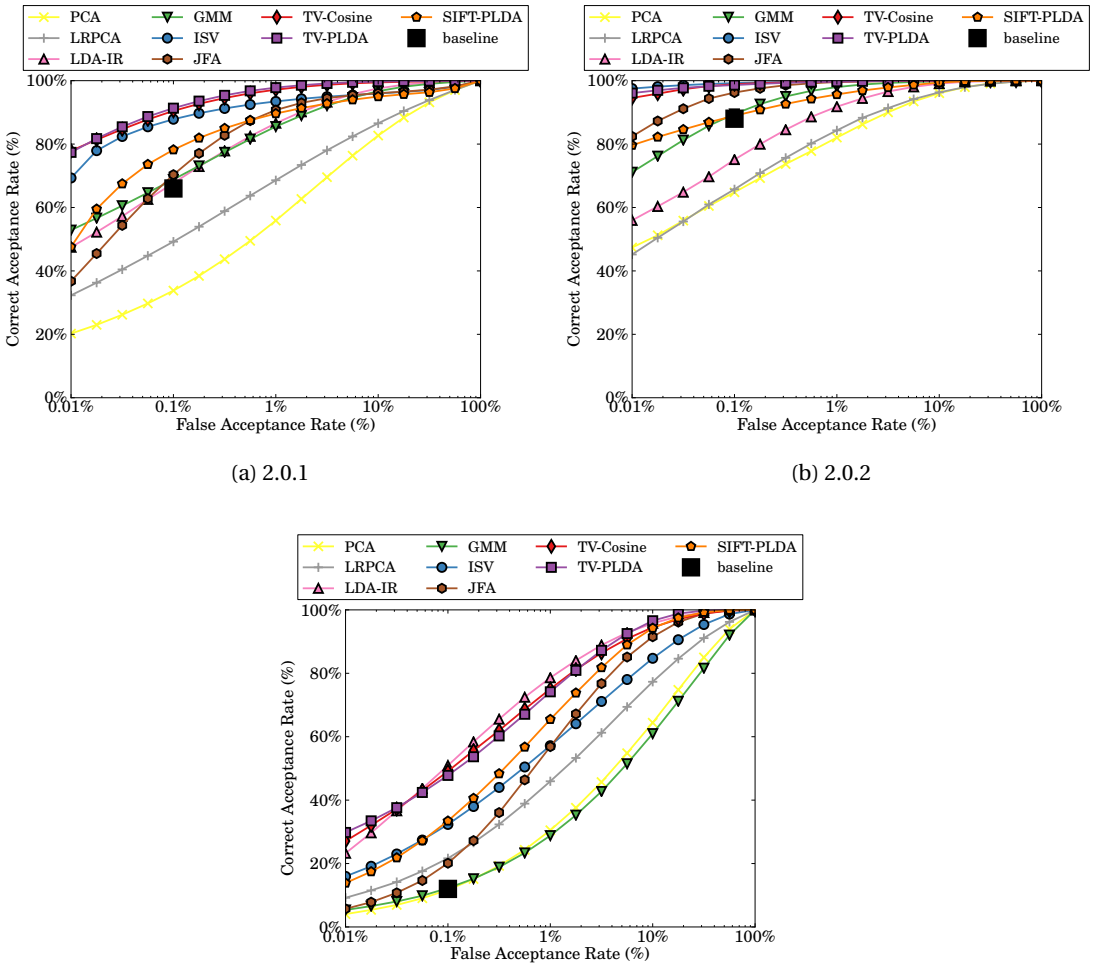
The FRGC database comes with 3 different biased verification protocols. The ROC curves for the algorithms executed on these experiments are presented in fig. 5.15. The baseline results reported by [Phillips et al., 2006] are also present in the plot as a single marker at 0.1% FAR.

Experiment *2.0.1* compares controlled studio portrait images with each other, using one image for model enrollment and one image for probing. In this scenario, the **TV**-based systems outrival the other algorithms, followed by **ISV**. Interestingly, all the **GMM**-based systems outperform the **GMM**-baseline, which confirms the applicability of session variability modeling techniques in face recognition. In contrast, **LDA-IR** does not perform very well in this experiment. This suggests that, extracting color information does not seem to be beneficial when images are taken in controlled environments.

Experiment *2.0.2* tests how well multiple images per person can improve verification performance. The results in fig. 5.15(b) show that most of the proposed systems gain a lot in performance, while the CSU baselines **LRPCA** and **LDA-IR** cannot exploit multiple enrollment images that well.

Finally, experiment *2.0.4* uses probe images with uncontrolled illumination. Fig. 5.15(c) illustrates that the **TV**-based systems and **LDA-IR** work nicely on this experiment, followed by **ISV** and **SIFT-PLDA**, while **PCA** and **GMM** perform poorly. Again, **GMM**-based session variability modeling techniques bring significant improvements, when compared to the **GMM** baseline. For instance, **TV-PLDA** brings a relative improvement in the CAR at FAR = 0.1% of 33%, 10% and 289% for experiment *2.0.1, 2.0.2* and *2.0.4*, respectively, when compared to **GMM**.

On all experiments, the best proposed systems outperform the FRGC baseline.



(a) 2.0.1

(b) 2.0.2

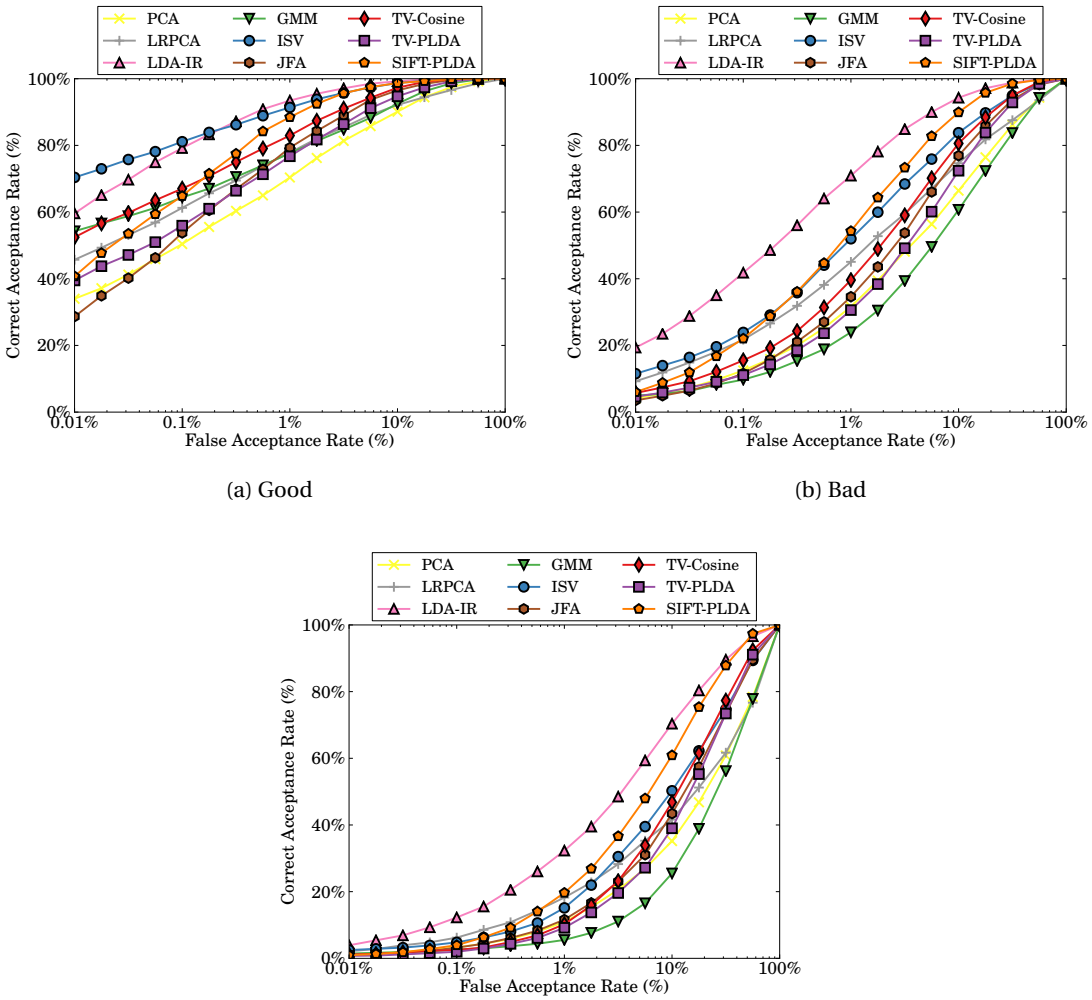
(c) 2.0.4

Figure 5.15 – PERFORMANCE OF THE SYSTEMS ON FRGC. *This figure shows ROC curves for the different protocols of FRGC. The FRGC baseline performance of* 66%, 88% *and* 12% CAR *at* FAR = 0.1%, *respectively, is marked.*

The Good, the Bad and the Ugly (GBU)

The GBU database provides verification protocols with increasing difficulty: *Good*, *Bad* and *Ugly*. The ROC curves for all tested algorithms are given in fig. 5.16. Compared to FRGC, there is a difference in the training set, which is about six times smaller (2,076 images).

On protocol *Good*, several systems perform reasonably well, the best two being **ISV** and **LDA-IR**. **SIFT-PLDA** is also fairly accurate for operating points, where the FAR is comparable or greater than the FRR. For protocols *Bad* and *Ugly*, **LDA-IR** outnumbers all other algorithms. This might be explained by the additional use of color information and by the fact that this algorithm was designed and tuned for this database. In particular, the **GMM** baseline performs poorly on these protocols. But the derived session variability modeling techniques allow a



(a) Good

(b) Bad

(c) Ugly

Figure 5.16 – PERFORMANCE OF THE SYSTEMS ON GBU. *This figure shows ROC curves with a logarithmic* FAR *axis for the different protocols of GBU.*

significant boost in performance. In addition, **SIFT-PLDA** is fairly accurate on these protocols as well.

Labeled Faces in the Wild (LFW)

The last database, on which we test our algorithms, is the *Labeled Faces in the Wild* (LFW) database. Fig. 5.17 displays the average classification rates as well as the standard deviations over the 10 different folds of *view 2* as required by the LFW protocol [Huang et al., 2007b].

With 83.5% [18] average classification accuracy, **SIFT-PLDA** performs best on this database, followed by **TV-PLDA**, **TV-Cosine**, **JFA** and **ISV**, while human performance is about 97.5%. These results suggest that **PLDA** modeling is suitable to deal with images of faces in the wild, by accurately modeling the within-class variations. In contrast, the three baseline systems, **PCA**, **LRPCA** and **LDA-IR** perform relatively poorly in this scenario. This might be explained by the use of pixel-based representations of the face, instead of more robust local features.

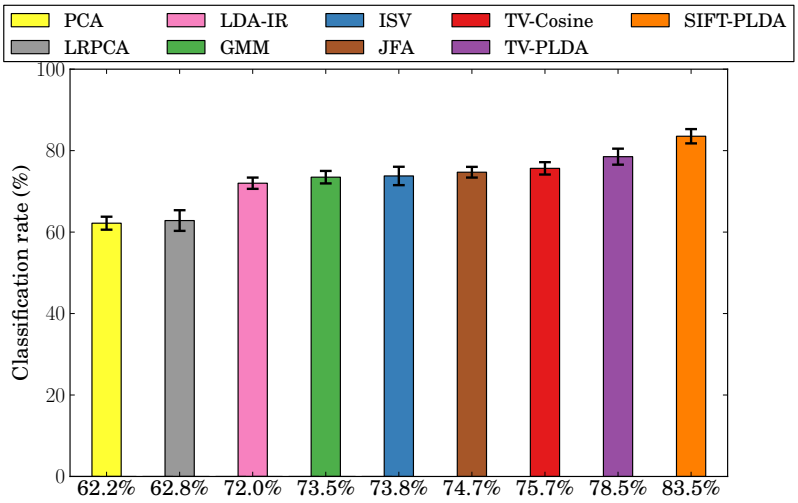


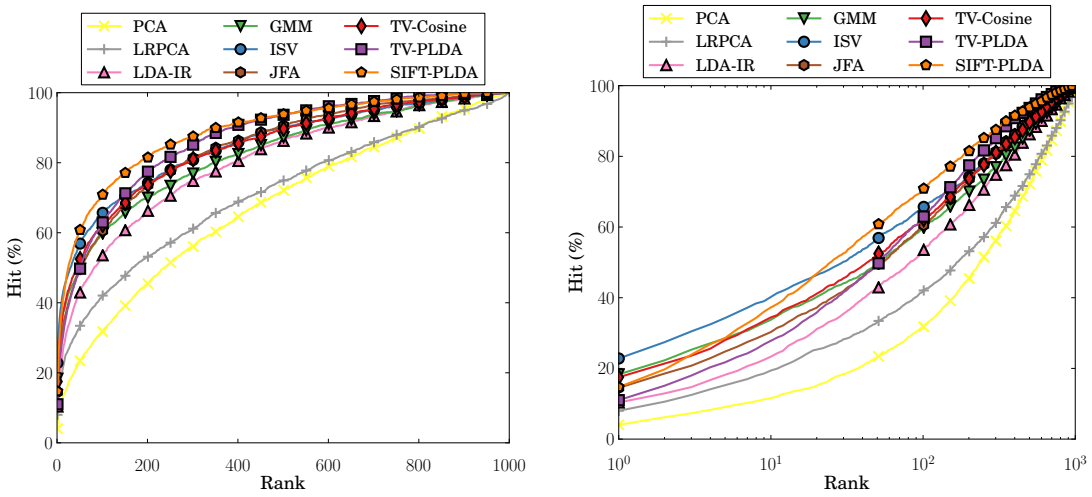
Figure 5.17 – VERIFICATION PERFORMANCE OF THE SYSTEMS ON LFW. *This figure shows the classification rate on LFW* view 2.

In addition to the previous and standard evaluation on LFW, we additionally run the face recognition systems on the previously defined *P0* protocol (see sec. 5.2.2), which aims at addressing the problem of large-scale identification. Results on the evaluation set are reported on fig. 5.18, which displays CMC curves, using both a linear (fig. 5.18(a)) and logarithmic (fig. 5.18(b)) x-axis to improve legibility.

Interestingly, the best performing system depends on the operating point on the system. Considering an identification rank in the range [1,20], **ISV** outperforms other systems, the recognition rate at rank 1 being slightly above 20%. If the system can afford to return a larger output list of potential matches (more than 20), **SIFT-PLDA** is then the most accurate system, and for even larger list, **TV-PLDA** becomes a good choice as well. These results suggest that these techniques are able to cope with the session variability issue up to a certain level.

In addition, some errors returned by one of the well performing systems, **SIFT-PLDA**, are given on fig. 5.19. Enrollment being performed with a single image, pair of samples (one for enrollment and one for test) of the same subject with the lowest scores are shown on fig. 5.19(a). Similarly, pair of samples of different subjects leading to the highest scores are shown on fig. 5.19(b).

18. In [El Shafey et al., 2013c], we reported a classification rate of 86.3% for **SIFT-PLDA**, but we used the visual features made publicly available by [Guillaumin et al., 2009] that rely on a facial feature detectors instead of a fixed grid of keypoints.



(a) Identification on LFW (rank on a linear scale)

(b) Identification on LFW (rank on a logarithm scale)

Figure 5.18 – IDENTIFICATION PERFORMANCE OF THE SYSTEMS ON LFW. *This figure shows the identification performance on the protocol* P0 *of the LFW database.*

Overall, similar conclusions as in sec. 5.4.1 can be drawn: pose and occlusions (sunglasses) remain major challenges in face recognition.

(a) Samples of the same subject with lowest scores

(b) Samples of different subjects with highest scores

Figure 5.19 – LARGEST ERRORS OF SIFT-PLDA ON THE IDENTIFICATION PROTOCOL OF LFW. *This figure shows the positive trials with the lowest scores (a) and the negative trials with the highest scores (b) for SIFT-PLDA on the identification protocol of LFW. Trials are organized by column (enrollment sample is on the top, test sample on the bottom).*

Finally, we can observe that face recognition techniques still require significant improvements in terms of accuracy to make possible identification on large datasets. If the returned list consists of 10% of the total number of identities (1,000 on the evaluation set), the identification rate of the best system, **SIFT-PLDA**, is only around 70%.

5.5 Summary and Concluding Remarks

In this chapter we applied the considered probabilistic models to the task of face recognition. In particular, we separately examined the impact of face variations such as pose, illumination or expression on the performance of the systems, before conducting experiments on uncontrolled databases. Overall, the proposed modeling techniques bring significant improvements in uncontrolled recording conditions, **TV-PLDA** and **ISV** being very competitive.

The experimental findings can be summarized as:

1. GMM-based session variability modeling techniques bring significant improvements, when compared to the **GMM** baseline. While **TV**-based systems, and especially **TV-PLDA** require a sufficiently large training set to be efficient, **ISV** and **JFA** perform reasonably throughout all the conducted experiments.
2. Pose remains a major problem for automatic face recognition, when there is a mismatch between enrollment and test samples. In this scenario, the best two systems are **TV-PLDA** and **JFA**, which provide a relative improvement in HTER of 50% when compared to **GMM**.
3. The proposed systems are able to cope reasonably well with non-frontal illumination and facial expressions. In particular, results suggest that adding enrollment samples taken under various conditions improves the accuracy of face recognition systems.
4. Severe occlusions caused by sunglasses affect the performance of all the systems. This can be explained by the fact that the region around the eyes contains important cues for face recognition
5. The results of the **LDA-IR** baseline show that color information is a useful cue to boost the performance in uncontrolled environments. Therefore, these information could be integrated in the proposed approaches to improve their accuracy in real world applications.
6. In general, raw pixel-based approaches such as **PCA** and **LRPCA** do not perform well on challenging recording conditions, in contrast to local feature-based techniques.
7. **ISV** and the **TV**-based systems are the best performing systems on the large FRGC database. For instance, a relative improvement in the CAR at FAR = 0.1% of 33% is observed with **TV-PLDA** on experiment *2.0.1* when compared to **GMM**.
8. Finally, the identification experiments conducted on LFW suggest that face recognition systems should still be significantly improved to make possible face identification in uncontrolled environments on a large dataset.

In the next chapter, we apply the same modeling techniques to the task of speaker recognition.

6 Application to Speaker Recognition

In this chapter we apply the proposed modeling techniques to the task of automatic speaker recognition. In particular, this chapter is mainly an extension of the Idiap Research Institute submission to the 2012 NIST Speaker Recognition Evaluation (NIST SRE12), where an inter-session variability-based system was submitted.

We begin by discussing related work in the field of speaker recognition. Next, we introduce the NIST SRE12 corpus and we describe the speaker recognition systems considered. Finally, large-scale experiments are reported and discussed.

6.1 Background

Soon after the development of digital computers, there has been a considerable amount of activity in laboratories and universities to build automatic speaker recognition systems [Reynolds, 1995b]. Semi-automatic systems have originally been proposed based on a visual comparison of speech spectrograms, which are representations of the spectrum of frequencies in a signal [Bolt et al., 1970]. But quickly, fully automatic systems were developed using simple template matching techniques (for instance, at the Bell Labs [Atal, 1974]). In addition, early work showed that automatic systems could outperform human listeners [Rosenberg, 1973].

Several applications of speaker recognition technology are possible ranging from forensics to automatic indexing of audio content. In particular, telephone-based scenarios are very common, such as identification of criminals or verifying a person's identity before a banking transaction. Depending on the level of cooperation of the person to recognize, these applications rely on either *text-dependent* or *text-independent* speech. In a text-dependent application, the recognition system has a priori knowledge of the text spoken by the subject. On the other hand, text-independent recognition is more challenging, but also more flexible.

Numerous approaches for speaker recognition have been proposed during the past four decades and we, hence, do not aim at providing a thorough review of the field. Compared to early template-based systems, current state-of-the-art techniques benefit from recent ad-

vances in statistical machine learning [Kinnunen and Li, 2010]. However, phonetic variability, changes in the environment (e.g., acoustic channel, background noise) and within-speaker variation (e.g., emotional state, aging) remain major challenges in speaker recognition.

Refining fig. 2.1, a recent automatic speaker recognition system typically consists of the steps depicted in fig. 6.1. In the following, we review existing work for each step of the processing chain, considering text-independent speaker recognition.

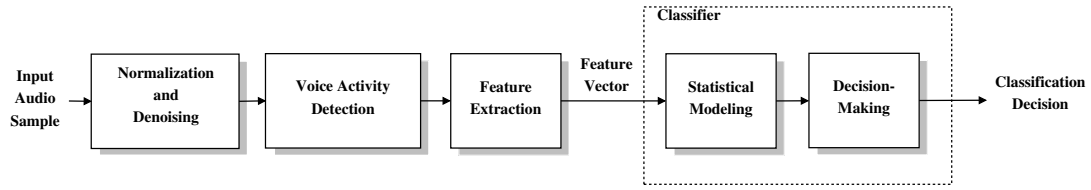


Figure 6.1 – SIMPLIFIED STRUCTURE OF A TYPICAL SPEAKER RECOGNITION SYSTEM.

6.1.1 Normalization and Denoising

After its acquisition via a microphone and its conversion into digital form, an audio sample is resampled, if required, to unify sampling rates of training and enrollment samples. Next, denoising techniques are typically applied to improve the signal-to-noise ratio (SNR) of the input signal. Popular methods for this purpose are spectral subtraction [Boll, 1979], short-time spectral amplitude [Ephraim and Malah, 1984] and Wiener filters [Adami et al., 2002].

6.1.2 Voice Activity Detection

Voice activity detection (VAD) aims at determining when a person is speaking in an audio sample. This allows to discard non-speech segments that do not contain speaker information prior to recognition. VAD can be seen as a binary classification task, where the goal is to classify audio segments as speech or non-speech. Supervised approaches could be used, but in practice, few databases are annotated with speech and non-speech labels over time.

A simple unsupervised approach for VAD relies on the energy of an audio segment. This energy measure can be compared to a threshold to determine whether it is classified as speech or non-speech. A slightly more sophisticated approach consists of training a two Gaussians classifier from the audio segments, where the Gaussian distribution with the higher mean value corresponds to the speech part [Khoury et al., 2012].

Instead of the energy, it is possible to rely on other measurements. In particular, speech signals have a characteristic energy modulation peak around the 4 Hz syllabic rate, which can be efficiently employed to discriminate between speech and, e.g., music [Scheirer and Slaney, 1997]. Other popular strategies consist of zero-crossing rate [Benyassine et al., 1997], periodicity measure [Tucker, 1992] or spectrum analysis [Marzinzik and Kollmeier, 2002].

In practice, VAD has a significant impact on speaker recognition systems and state-of-the-art algorithms tend to fail when the level of background noise is too high.

6.1.3 Feature Extraction

After voice activity detection, speech samples are of various length and only a fraction of the information conveyed is useful for speaker discrimination. Feature extraction aims at mapping the signal into a low-redundancy representation that still conveys information about the speaker. These techniques can be divided into different categories [Kinnunen and Li, 2010].

Short-term spectral features characterize properties of vocal tract as well as the short-term spectral envelope of the signal. They are computed from short segments (usually 20-30 milliseconds), which are extracted from the speech signal at equally-spaced time instants using a sliding window approach. A feature vector is then computed from each segment.
Mel frequency cepstrum coefficients (MFCCs) [Davis and Mermelstein, 1980] are commonly used. This representation is obtained by first taking the discrete Fourier transform (DFT) of the observations, before mapping the powers of the spectrum on a non-linear Mel scale of frequency. Next, the logarithms of the powers at each frequency of the scale are computed before applying the 1D DCT to them. Coefficients are then obtained by taking the amplitudes of the resulting spectrum.
Linear prediction (LP) [Mammone et al., 1996] is an alternative spectrum estimation method to DFT. It is inspired by a popular linear model of speech production developed at the end of the fifties [Fant, 1960]. The linear coefficients are determined using the Levinson-Durbin recursion [Rabiner and Juang, 1993]. They are rarely used as features, but are transformed into more robust and less correlated coefficients such as linear predictive cepstral coefficients (LPCCs) [Huang et al., 2001], or perceptual linear prediction (PLP) coefficients [Hermansky, 1990].

Voice source features characterize the voice source, such as fundamental frequency and glottal pulse shape. They are less discriminative than short-term features, but complementary [Kinnunen and Li, 2010].

Spectro-temporal features refer to formant transitions and energy modulations. First- and second-order time derivative estimates (called delta (Δ) and double-delta ($\Delta\Delta$)) are commonly employed to incorporate some temporal information to features [Furui, 1981]. Time differences between adjacent feature vectors are typically computed and appended to the corresponding feature vector.

Prosodic features rely on non-segmental aspects of speech such as rhythm, syllable stress, speaking rate and intonation. Unlike short-term spectral features, they span over long segments like syllables or words, and reflect differences in speaking style and emotions [Kinnunen and Li, 2010].

Finally, *high-level features* attempt to capture conversation-level characteristics of speakers, such as the usage of certain words and phrases [Doddington, 2001].

6.1.4 Modeling and Classification

Simple template matching approaches such as vector quantization (VQ) were proposed for text-independent speaker recognition [Soong et al., 1985]. However, current state-of-the-art techniques rely on popular machine learning algorithms. Both generative and discriminative models were successfully applied.

In particular, Gaussian mixture models (GMMs) have been widely employed to estimate the feature distribution within each speaker [Reynolds, 1992, Reynolds and Rose, 1995, Reynolds, 1995a]. On the other side, discriminative models such as artificial neural networks [Farrell et al., 1994, Yegnanarayana and Kishore, 2002] and support vector machines (SVMs) [Campbell et al., 2004, 2006a] were proposed to model the boundary between speakers.

Speech utterances usually have a varying number of feature vectors, and it is hence difficult to find a suitable representation for speaker models. Early work proposed to average features over time such that each utterance could be represented as a vector of fixed dimensionality [Markel et al., 1977]. Interestingly, a major trend in the speaker recognition field consists of representing utterances using a single vector, called a *supervector*. These supervectors have been used in combination with SVMs, leading to systems called generalized linear discriminant sequence (GLDS) kernel SVM [Campbell et al., 2006a] and maximum likelihood linear regression (MLLR) supervector SVM [Karam and Campbell, 2007, Stolcke et al., 2007]. Besides, hybrid systems were proposed where a GMM is used for creating feature vectors that then feed a SVM [Campbell et al., 2006b, Dehak and Chollet, 2006, Lee et al., 2008].

In this chapter, we investigate GMM-based supervector techniques, such as inter-session variability (ISV) modeling, joint factor analysis (JFA) and total variability (TV or i-vectors) modeling (see chapter 3). Furthermore, we employ our scalable formulation of probabilistic linear discriminant analysis (PLDA) to classify i-vectors (see chapter 4).

6.2 NIST SRE12 Database

Early studies evaluated speaker recognition systems on databases consisting of a few speakers [Rosenberg, 1973]. Recently, there has been a significant effort in acquiring and distributing larger corpora to researchers. In particular, the National Institute of Standards and Technology (NIST) started a series of text-independent speaker recognition evaluation (SRE) [1] in 1996 [Martin and Przybocki, 2001]. The goal has been to drive the technology forward, to evaluate the state-of-the-art, and to find the most promising algorithms.

1. http://www.nist.gov/itl/iad/mig/sre.cfm

The latest evaluation, NIST SRE12, took place in 2012 and was the largest and most complex SRE up to date [Greenberg et al., 2013]. This evaluation provides a set of target speaker training data (enrollment data) as well as a set of test segments for scoring, which together define an evaluation set (the labels of the test segments were only given after the evaluation). The data were drawn from several corpora collected by the Linguistic Data Consortium (LDC): [2] Mixer (versions 1 to 7) and BEST speaker evaluation. It consists of telephone recorded phone calls, microphone recorded phone calls, and microphone recorded face-to-face interviews. The speakers were encouraged to perform their phone calls in a noisy environment. Besides, artificial noise was added to a subset of segments, with levels of +6 dB or +15 dB.

The NIST SRE12 does not provide data for training and development purposes. Therefore, we rely on the development set generated by the I4U coalition [Saedi et al., 2013]. This allows us to tune speaker recognition systems to conditions similar to the ones of NIST SRE12. The development set consists of data fetched from NIST SRE06, NIST SRE08 and NIST SRE10, which were initially collected by LDC. Besides, a training set was generated based on data extracted from other LDC corpora (Switchboard, Fisher, NIST SRE04, NIST SRE05 and NIST SRE06). In both sets, additional utterances were generated by incorporating noise with levels of +6 dB and +15 dB to simulate NIST SRE12 conditions.

NIST SRE12 evaluation is performed separately for male and female. Several evaluation protocols were released by NIST for SRE12. This protocol (that we refer to as α-*extended*) compares all the probe samples to all the enrolled speaker models. The NIST SRE12 *core* protocol is a subset of this protocol, where only a part of the trials are performed. This α-*extended* protocol corresponds to the initial plan for the NIST SRE12 *extended* protocol. However, the NIST SRE12 *extended* protocol has later been extended with additional probe samples. The resulting protocols have been made available online. [3]

Gender	Group	Number of speakers	Number of segments (train or test)	Number of trials
Male	Train (Idiap)	6,767	33,322	n.a.
	DEV (I4U)	680	19,866	13,508,880
	EVAL	763	29,728	22,682,464
Female	Train (Idiap)	9,198	42,521	n.a.
	DEV (I4U)	1,039	25,980	26,993,220
	EVAL	1,155	43,378	50,101,590

Table 6.1 – NIST SRE12 EVALUATION PROTOCOLS. *This table depicts the number of speakers, utterances and trials in NIST SRE12 (evaluation set,* EVAL*) as well as in the training (*Train*) and development (*DEV*) sets employed in this study.*

2. http://catalog.ldc.upenn.edu/
3. http://pypi.python.org/pypi/xbob.db.nist_sre12

6.3 Systems Description

Five different speaker recognition systems are evaluated, which all share the same preprocessing and feature extraction techniques. These systems are described in the remainder of this section. In addition, a summary is provided in tab. 6.2.

<table>
<tr><th></th><th>GMM</th><th>ISV</th><th>JFA</th><th>TV-Cosine</th><th>TV-PLDA</th></tr>
<tr><td>Denoising</td><td colspan="5">Qualcomm-ICSI-OGI [Adami et al., 2002]</td></tr>
<tr><td>Resampling</td><td colspan="5">8 kHz</td></tr>
<tr><td>VAD</td><td colspan="5">log energy</td></tr>
<tr><td>Feature Extraction</td><td colspan="5">MFCC features ($D_{\boldsymbol{o}} = 60$)
• 20 ms Hamming windowed frames with an overlap of 10 ms
• 19 Mel frequency cepstrum coefficients
+ log energy
+ first- and second-order derivatives
Cepstral mean and variance normalization (CMVN)</td></tr>
<tr><td rowspan="3">Modeling</td><td colspan="5">UBM GMM
• 512 components with diagonal covariance
• 50 k-means iterations
• 50 EM iterations</td></tr>
<tr><td>GMM MAP
• Relevance factor $\tau = 4$</td><td>ISV
• $D_{\boldsymbol{U}} = 200$
•10 EM iterations</td><td>JFA
• $D_{\boldsymbol{U}} = D_{\boldsymbol{V}} \in \{50, 100\}$
•10 EM iterations</td><td colspan="2">Total Variability
• $D_{\boldsymbol{T}} = 400$
• 25 EM iterations
• Whitening
• WCCN</td></tr>
<tr><td colspan="3"></td><td>$d_{\cos}$</td><td>PLDA
• $D_{\boldsymbol{F}} = D_{\boldsymbol{G}} \in \{50, 100\}$
• 200 EM iterations</td></tr>
</table>

Table 6.2 – DESCRIPTION OF THE SPEAKER RECOGNITION SYSTEMS.

A resampling at 8 kHz is first applied when required, since NIST SRE12 contains data encoded at either 8 kHz or 16 kHz. The Qualcomm-ICSI-OGI front end [Adami et al., 2002] is then employed for denoising purposes. Next, VAD is performed by considering the normalized log energy.

After, normalization and VAD, observations are extracted every 10 milliseconds using a Hamming window of 20 milliseconds. In addition to the log-energy, 19 Mel frequency cepstrum coefficients (MFCCs) are computed using a filter bank of 24 filters, together with their first- and second-order derivatives (Δ and $\Delta\Delta$) to integrate spectro-temporal information. From each observation, this results in a feature vector of dimensionality $D_{\boldsymbol{o}} = 60$. Finally, a normalization technique called *cepstral mean and variance normalization* (CMVN) [Strand and Egeberg, 2004] is applied on the remaining speech. The number K of feature vectors extracted from each audio sample depends on the number of segments that the VAD classifies to be speech.

The overall process is illustrated by fig. 6.2.

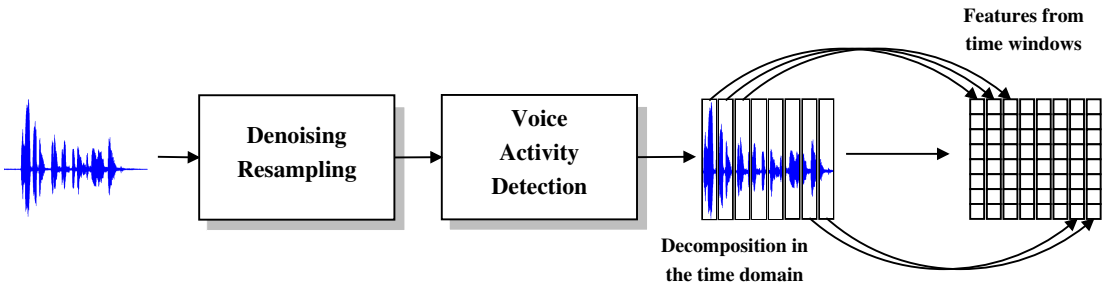


Figure 6.2 – AUDIO FEATURE EXTRACTION. *This figure provides a simplified view of the audio feature extraction process, decomposing the signal in the time domain and obtaining a feature vector from each observation.*

The five speaker recognition systems are based on the modeling techniques introduced in chapter 3 and chapter 4: **GMM**, **ISV**, **JFA**, **TV-Cosine** and **TV-PLDA**.

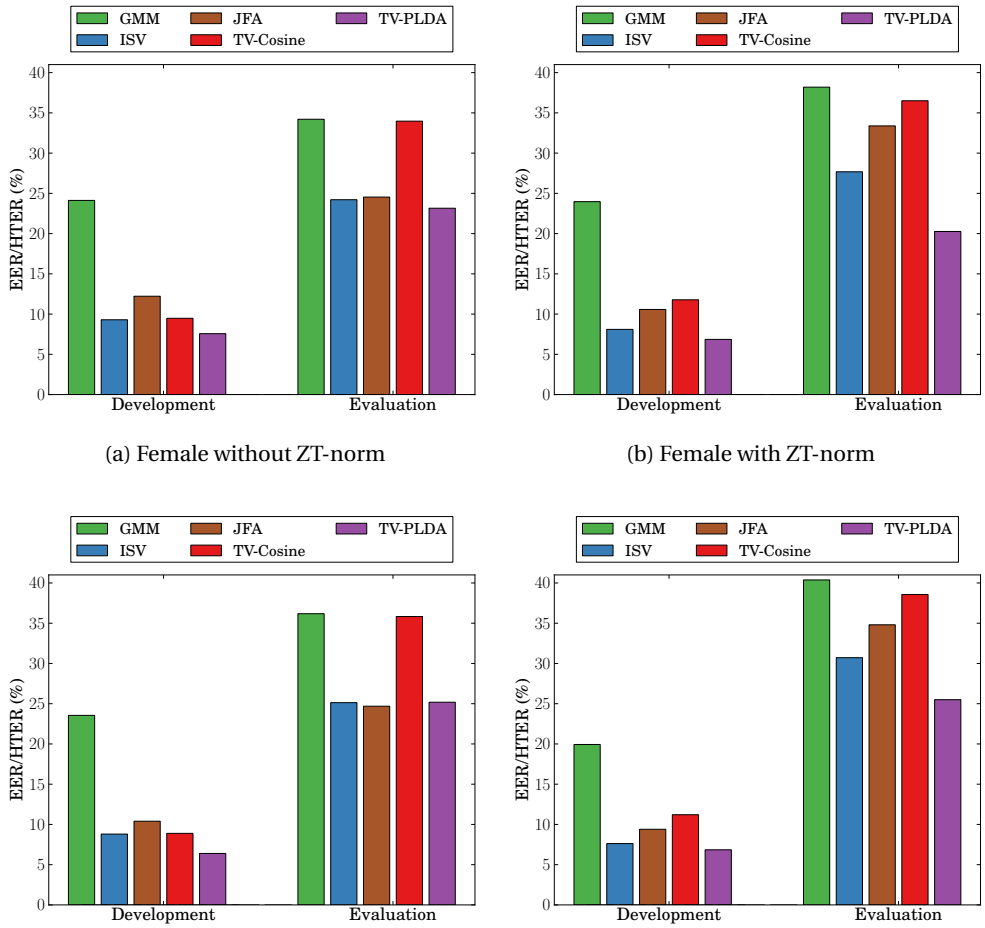
The UBM is shared by all of them. Therefore, at training time, we first derive this UBM, before learning the subspaces $\boldsymbol{U}$, $\boldsymbol{V}$, $\boldsymbol{D}$ and $\boldsymbol{T}$ of the **ISV**, **JFA** and **TV** systems. The UBM consists of 512 Gaussian components with diagonal covariance matrices. It is trained in a gender-independent way using 50 iterations of k-means followed by 50 iterations of EM. Due to the large number of training samples of long duration, this training process is computationally very intensive and still requires several days/weeks after parallelization on a cluster infrastructure.

The next step is to learn the subspaces, which model either within-class or between-class variations. In contrast to the UBM, they are learned in a gender-dependent way, modeling the variability of each gender separately. Furthermore, we had to restrict the search space for these large scale experiments when optimizing parameters. For the **ISV** system, the rank of $\boldsymbol{U}$ is set to $D_{\boldsymbol{U}} = 200$, whereas the rank of the subspaces $\boldsymbol{U}$ and $\boldsymbol{V}$ of **JFA** is set to $D_{\boldsymbol{U}} = D_{\boldsymbol{V}} = \nu$ with $\nu \in \{50, 100\}$. For both **ISV** and **JFA**, 10 iterations of EM are performed. For the **GMM** and **ISV** systems, the relevance factor τ is set to 4. Finally, 25 iterations of EM are performed for training the total variability subspace $\boldsymbol{T}$, and the dimensionality of the i-vectors in **TV-Cosine** and **TV-PLDA** is equal to 400. The scoring techniques of the two **TV** systems are the cosine similarity measure (see eq. (3.59)) and PLDA, respectively. For **TV-PLDA**, the rank of the subspaces $\boldsymbol{F}$ and $\boldsymbol{G}$ is set to $D_{\boldsymbol{F}} = D_{\boldsymbol{G}} = \nu$ with $\nu \in \{50, 100\}$ and 200 iterations of EM are performed.

ZT-norm score normalization [Auckenthaler et al., 2000] is performed, using clean and noisy data from the training set.

Finally, before the generation of the DET curves fig. 6.4, the scores are calibrated using linear logistic regression [Bishop, 2007] (see sec. 7.2.2), before being mapped into compound log-likelihood ratios [Brümmer, 2012], as suggested by participants during the evaluation campaign. [4]

4. http://sites.google.com/site/bosaristoolkit/sre12



(a) Female without ZT-norm

(b) Female with ZT-norm

(c) Male without ZT-norm

(d) Male with ZT-norm

Figure 6.3 – PERFORMANCE OF THE SYSTEMS ON NIST SRE12 (EER AND HTER). *This figure shows the* EER *on the development set and the* HTER *on the evaluation set of NIST SRE12.*

6.4 Experimental Results

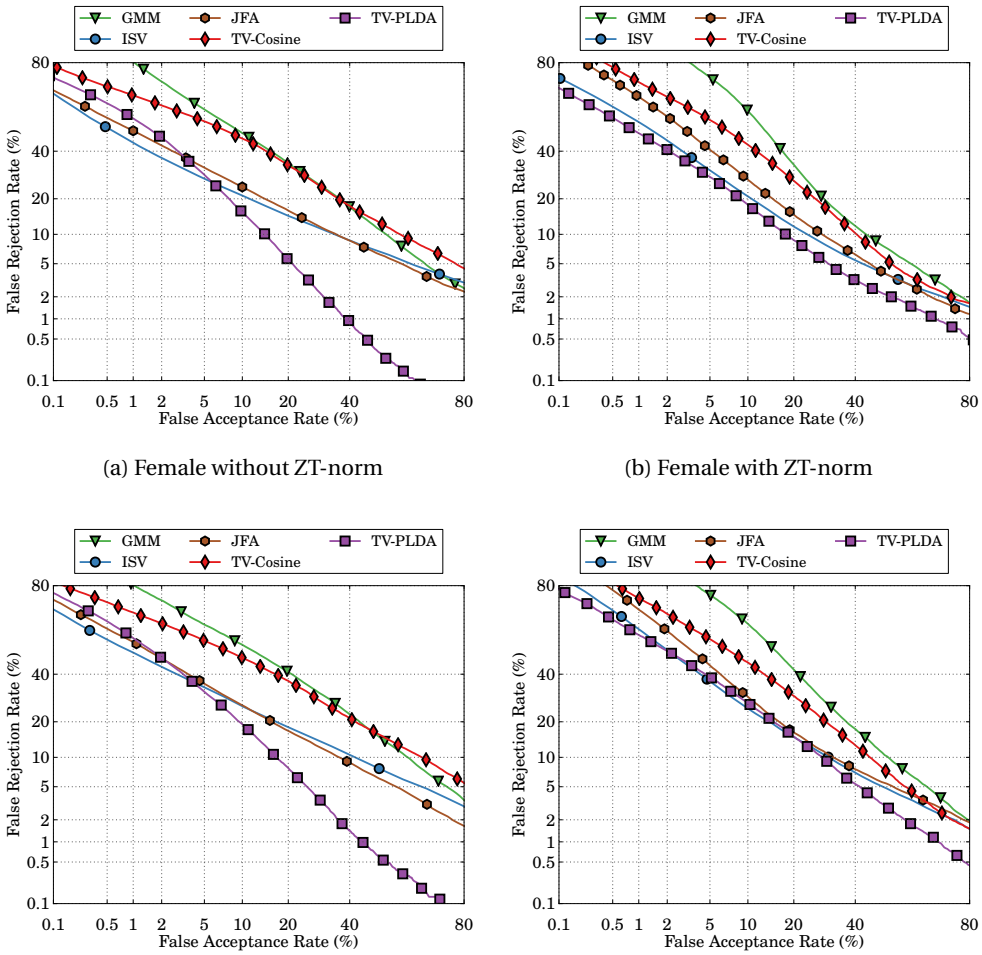
In this section, we evaluate the considered speaker recognition systems on NIST SRE12. The techniques employed by these systems were implemented in Bob [Anjos et al., 2012] (see appendix A). In addition, all the results and the plots reported in this section can be easily regenerated using the satellite package [5] that accompanies this dissertation (see sec. A.3.2).

We report the results both without and with ZT-norm score normalization. We tune the subspace size of **JFA** and **TV-PLDA** on the development set, considering the EER as the value to optimize. No clear consensus emerged among the two different parameterizations. For **TV-PLDA**, a subspace size of $D_F = D_G = 100$ leads to a smaller EER than with $D_F = D_G = 50$, except for male when the ZT-norm score normalization is not applied. Considering **JFA**, a

5. https://pypi.python.org/pypi/xbob.thesis.elshafey2014

subspace size of $D_U = D_V = 100$ is better for male, whereas $D_U = D_V = 50$ is slightly more suitable for female. In the following, we always consider the results obtained with subspaces of dimensionality $D_U = D_V = 100$ and $D_F = D_G = 100$ for **JFA** and **TV-PLDA**, respectively.

First, we report results at a specific operating point, considering the EER on the development set and the HTER on the evaluation set, separately for female and male (fig. 6.3). Second, DET curves on the evaluation set are reported on fig. 6.4, for which additional calibration and normalization steps are performed (see end of sec. 6.3).



(a) Female without ZT-norm

(b) Female with ZT-norm

(c) Male without ZT-norm

(d) Male with ZT-norm

Figure 6.4 – PERFORMANCE OF THE SYSTEMS ON NIST SRE12 (DET CURVES). *This figure shows the DET curves on the evaluation set of NIST SRE12.*

6.4.1 Global Observations

Looking at the results provided in fig. 6.3 and fig. 6.4, three major trends emerge. First, the session variability modeling techniques bring significant improvements compared to the

GMM baseline. In particular, **TV-PLDA** outperforms other systems over a wide range of operating points, followed by **ISV** and **JFA**. Second, the conditions encountered in the NIST SRE12 evaluation set are more challenging than the ones in the development set established by the I4U coalition. Between these two sets, there is indeed a ratio of two to three in terms of error rate, depending on the system considered. Besides, HTER values reported on the evaluation set for all the systems are fairly high, which confirms that NIST SRE12 is a challenging task. Third, a similar performance is observed for both female and male, and the same conclusions can be drawn.

6.4.2 Comparison of the Systems

Comparing the best system, **TV-PLDA**, to the **GMM** baseline, we observe a relative gain in HTER on the evaluation set of 47% and 37% for female and male, respectively. Interestingly, **TV-PLDA** brings significant improvements, when compared to **TV-Cosine** that employs a simple distance on the same input i-vectors. Particularly, this implies that PLDA modeling results in a relative gain in HTER of 45% and 28% for female and male, respectively. This highlights the effectiveness of this modeling technique for the task of speaker recognition, which was also confirmed during the NIST SRE12 campaign [Saedi et al., 2013]. In addition, the scalable formulation proposed in chapter 4 allows a very efficient training (and scoring) process. In contrast to the EM procedures employed for learning the UBM and the subspaces of **ISV**, **TV** and **JFA**, it was not necessary to parallelize the **PLDA** training on a cluster infrastructure.

Similarly, **ISV** outperforms the **GMM** baseline with a relative gain in HTER of 28% and 24%, for female and male, respectively, whereas for **JFA**, relative improvements of 13% and 14% are achieved. **ISV** is of particular interest for scenarios that penalize more false rejections than false acceptances, since its DET curve is below the one of **TV-PLDA**, for both male and female (see fig. 6.4).

6.4.3 Impact of the ZT-norm

As shown on fig. 6.3, the ZT-norm score normalization does not affect all the systems in the same way. Considering the development set, it boosts the performances of **GMM**, **ISV** and **JFA**, whereas, in contrast, it degrades the accuracy of the **TV** systems. For instance, the HTER of the **ISV** system is improved of about 16%, on average. Unfortunately, this normalization does not affect the results on the evaluation set in the same way. For almost all the systems, on both female and male, a degradation in performance is observed. As this normalization relies on samples from the training set, this suggests that the conditions encountered in NIST SRE12 are different from the ones of the training set, despite the generation of samples with additive noise. Therefore, the performance of the systems could possibly be improved by extending the training and I4U development sets with samples that better match the conditions of NIST SRE12.

6.5 Summary and Concluding Remarks

In this chapter, we applied several session variability modeling techniques to the task of speaker recognition. Experiments were conducted on the large NIST SRE12 database. Considering the standard **GMM** approach fed by MFCCs features as a baseline, we evaluated a set of supervector-based techniques, which all rely on these same MFCCs features.

Compared to the **GMM** baseline, these methods lead to a significant performance boost under the challenging NIST SRE12 conditions. Overall, the most accurate results are achieved with **TV-PLDA**, for which an efficient training and scoring procedure has been proposed in chapter 4. **ISV** also offers good performances, especially in scenarios that give more importance to false rejections than false acceptances.

In the next chapter, we apply the same modeling techniques to the task of bimodal face and speaker recognition.

7 Application to Bimodal Recognition

In this chapter we apply the proposed modeling techniques to the task of bimodal face and speaker recognition. Unimodal face and speaker recognition systems often have to deal with a high session variability leading to high error rates. Attempts to build more robust unimodal systems may not be effective because of this intrinsic difficulty. Bimodal recognition seeks to alleviate this drawback by relying on multiple cues.

We begin by discussing related work in the field of bimodal recognition. Next, we introduce our strategy for bimodal and multi-algorithm fusion. Finally, experiments are conducted on the challenging MOBIO database.

7.1 Background

The idea of using acoustic and visual cues for person authentication has started to receive attention in the nineties. This is a particular case of *multibiometric fusion* [Ross and Jain, 2003]. In an authentication scenario, this may force an intruder to spoof several modalities simultaneously, enhancing the reliability of the access control system. In an identification scenario, this is of particular interest when a modality is significantly affected by challenging conditions, as the other one may be available for the rescue.

[Brunelli et al., 1995, Brunelli and Falavigna, 1995] propose to combine the output scores of the unimodal face and speaker recognition subsystems using a statistical approach. As the scores of different systems may be in different ranges, they highlight the necessity of *normalizing the scores*. This normalization commonly consists of two steps. First, an estimation of the distribution of the scores of each subsystem is performed using statistical techniques. Second, these distributions are scaled and translated into a common range. Methods based on Bayesian statistics and on the Neyman–Pearson lemma to fuse scores of several subsystems are described in [Bigün et al., 1997] and [Jain et al., 1999a], respectively.

Instead of combining scores, it is also possible to rely on the *decision* (accept or reject) of several subsystems. A person identification system employing this strategy has been proposed

in [Dieckmann et al., 1997], which is based on a majority or unanimous voting to fuse the decisions of face and speaker recognition subsystems.

More generally, several schemes have been described in the literature to fuse scores or decisions. In particular, [Kittler et al., 1998] establishes a theoretical framework for combining classifiers considering various strategies.

Early work on bimodal face and speaker recognition was often conducted on small in-house databases. The M2VTS project was set up to address this problem, allowing researchers to build and evaluate bimodal authentication systems on well-defined protocols using a significantly larger amount of data. After a preliminary acquisition of a multimodal corpus comprising 37 subjects, this has finally given birth to the larger XM2VTS database [1] of 295 subjects [Messer et al., 1999]. Several researchers evaluated their work on this corpus. [Ben-Yacoub et al., 1999] investigate the use of classification techniques such as support vector machines [Vapnik, 1995], Bayesian classifiers, Fisher's linear discriminant analysis [Fisher, 1922], C4.5 decision trees [Quinlan, 1993] and multilayer perceptrons [Bishop, 2007] to fuse the outputs of unimodal systems. Similarly, [Verlinde and Cholet, 1999] propose to combine systems using k-nearest neighbors-based classifiers [Duda et al., 2000], C4.5 decision trees and logistic regression [Bishop, 2007].

The systems previously described rely on a fusion that is based either on the scores or on the decisions obtained by several unimodal subsystems. There has been attempts to combine the models at the level of the raw data. [Bengio, 2003] proposes to use asynchronous hidden Markov models (AHMMs) in order to build an audio-visual model and evaluates this approach on the M2VTS corpus.

A limitation of the XM2VTS corpus consists of the clean recording conditions which are not realistic enough compared to the real world situations. The BANCA database (see sec. 5.2.2) addresses this issue by using various kinds of recording equipment such as low quality cameras and microphones [Bailly-Baillière et al., 2003]. Since then, mobile phones have considerably evolved and become multimedia devices, which have a front-facing camera in addition to the standard microphone. Hence, it forms an exciting new device that allows researchers to explore the applicability of bimodal authentication in challenging mobile phone environments.

This challenge of bimodal authentication in the mobile phone environment has begun to receive more attention. An international competition was organized in 2010 [Marcel et al., 2010], where researchers evaluated state-of-the-art algorithms for face and speaker authentication using Phase I of the MOBIO database [McCool et al., 2012]. In this evaluation, enrollment was exclusively performed with mobile phone data. It was shown that a combination of these systems produced an impressive bimodal authentication system. Since then researchers have examined methods to perform face [Mau et al., 2010, Wallace et al., 2011], speaker [Perera et al., 2011, Roy et al., 2012] and bimodal [Shen et al., 2010, Motlicek et al., 2012, Khoury et al.,

1. http://www.ee.surrey.ac.uk/CVSSP/xm2vtsdb/

2013a,b] authentication in the challenging mobile phone environment.

In the remainder of this chapter, we investigate the use of the proposed modeling techniques for the task of bimodal authentication in mobile phone environment. In addition, we show the effectiveness of multi-algorithm fusion to further improve the results for face, speaker and bimodal authentication.

7.2 Bimodal and Multi-Algorithm Fusion

7.2.1 Taxonomy of Information Fusion

As depicted in the previous section, several fusion strategies are known in the literature [Dasarathy, 1994]. They can be classified into three main categories:

Low-level fusion also known as *data fusion*, combines multiple sources of raw data to produce new raw data. It is very rarely used in practice, as feature level fusion is commonly preferred.

Intermediate-level fusion or *feature level fusion* combines various features that might come from several raw data sources or even from the same raw data. It is affected by similar issues to those in low-level fusion. In particular, a difficulty arises when data from multiple sources have unbalanced dimensionalities. In addition, synchronization between modalities [Shah et al., 2009] is required.

High-level fusion includes two different kinds of fusion mentioned earlier. *Score fusion* combines matching scores from several systems, whereas *decision fusion* combines decisions (accept or reject). These fusion strategies are very flexible and can be used for multimodal (face and speaker) or multi-algorithm fusion. High-level fusion methods include majority voting methods, fuzzy logic based methods [Lau et al., 2004] and statistical methods.

In this work we choose the score fusion approach due to its ease of use for both multimodal [Motlicek et al., 2012] and multi-algorithm [Pigeon et al., 2000, Brummer et al., 2007, McCool and Marcel, 2009] fusion. A drawback is that this approach neglects the interdependency of a person's spoken utterance and the associated facial movements, which is beyond the scope of this work.

7.2.2 Linear Logistic Regression

We take the well-known statistical *linear logistic regression* approach, which has been successfully employed for combining heterogeneous speaker and face authentication classifiers [Verlinde and Cholet, 1999, Jain et al., 1999b, Pigeon et al., 2000, Brummer et al., 2007, McCool and Marcel, 2009] and for bimodal (face and speaker) authentication [Motlicek et al., 2012, Khoury et al., 2013a,b].

Linear logistic regression combines a set of Q classifiers using a weighted sum. Let the

probe sample $\boldsymbol{\chi}_{\text{test}}$ be processed by Q classifiers, each of which produces an output score $h_q(\boldsymbol{\chi}_{\text{test}}, \mathscr{S}_i)$. These scores are fused using a linear combination:

$$h_{\text{fusion}}(\boldsymbol{\chi}_{\text{test}}, \mathscr{S}_i \mid \boldsymbol{\beta}) = g\left(\beta_0 + \sum_{q=1}^{Q} \beta_q h_q(\boldsymbol{\chi}_{\text{test}}, \mathscr{S}_i)\right), \tag{7.1}$$

where:

$$g(x) = \frac{1}{1+\exp(-x)}, \tag{7.2}$$

and $\boldsymbol{\beta} = [\beta_0, \beta_1, \ldots, \beta_Q]$ are the fusion weights (also known as *regression coefficients*).

The coefficients $\boldsymbol{\beta}$ are computed by estimating the maximum likelihood of the logistic regression model on the scores of the development set. Let $\mathbb{T}_{\text{true}}$ be the set of true claimant access trials, i.e., the set of pairs $\boldsymbol{t} = \{\boldsymbol{\chi}_{\text{test}}, \mathscr{S}_i\}$, where the class (identity) of the test sample $\boldsymbol{\chi}_{\text{test}}$ and of the model $\mathscr{S}_i$ is the same. Let furthermore $\mathbb{T}_{\text{imp}}$ be the set of impostor trials, i.e., the set of pairs $\boldsymbol{t} = \{\boldsymbol{\chi}_{\text{test}}, \mathscr{S}_i\}$, where the classes of the test sample $\boldsymbol{\chi}_{\text{test}}$ and of the model $\mathscr{S}_i$ are different. Let $\mathbb{T} = \mathbb{T}_{\text{true}} \cup \mathbb{T}_{\text{imp}}$. The objective function to maximize is:

$$L(\boldsymbol{\beta}) = -\sum_{\boldsymbol{t} \in \mathbb{T}} \log\left(1 + \exp\left(-y_{\boldsymbol{t}} h_{\text{fusion}}(\boldsymbol{t} \mid \boldsymbol{\beta})\right)\right), \tag{7.3}$$

where:

$$y_{\boldsymbol{t}} = \begin{cases} +1, & \text{if } \boldsymbol{t} \in \mathbb{T}_{\text{true}} \\ -1, & \text{if } \boldsymbol{t} \in \mathbb{T}_{\text{imp}} \end{cases} \tag{7.4}$$

The maximum likelihood estimation procedure converges to a global minimum. In our work, this optimization is done using the *conjugate-gradient* algorithm [Minka, 2001].

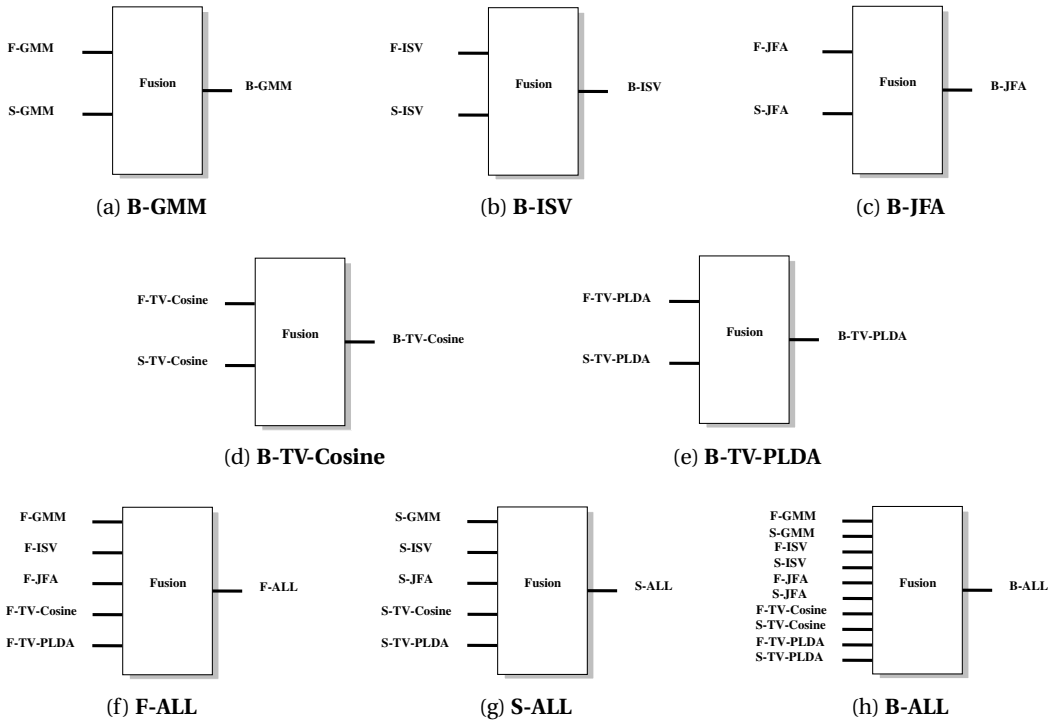
Score fusion performs best when the scores of the classifiers are statistically independent of each other. For this reason we measure the independence and, therewith, the complementary nature of our classifiers. We use the scatter plots (see fig. 7.6) and the *relative common error* (RCE):

$$\text{RCE} = \text{CE} \times \max\left\{\frac{1}{\text{TE}_1}, \frac{1}{\text{TE}_2}, \ldots \frac{1}{\text{TE}_Q}\right\}, \tag{7.5}$$

where CE is the number of *common errors* between the Q classifiers and TE_q is the *total number of errors* of the q^{th} subsystem. The lower RCE is, the more independent the classifiers are.

In this chapter we evaluate the effectiveness of both bimodal and multi-algorithm fusion. This leads to a number of different system combinations, which we outline in fig. 7.1. The top two rows of fig. 7.1 display the five different bimodal fusion systems, while the bottom row shows

the two different multi-algorithm fusion systems and the bimodal multi-algorithm fusion approach that we examine. To differentiate between these systems, we prefix them with **F-**, **S-** and **B-**, for face, speaker and bimodal, respectively.



(a) **B-GMM** (b) **B-ISV** (c) **B-JFA**

(d) **B-TV-Cosine** (e) **B-TV-PLDA**

(f) **F-ALL** (g) **S-ALL** (h) **B-ALL**

Figure 7.1 – FUSION STRATEGIES. *This figure displays different fusion strategies used in this this chapter: (a) - (e) bimodal fusion strategies, (f) - (g) multi-algorithm fusion strategies, (h) bimodal multi-algorithm fusion.*

7.3 MOBIO Database

The MOBIO database [McCool et al., 2012] is a unique bimodal (face and speaker) database as it was captured almost exclusively using mobile phones. It consists of over 61 hours of audio-visual data of 150 people captured within twelve sessions that are usually separated by several weeks. The users answered a set of questions, which varied in type, including:

1. *short response questions* (**p**) such as “what is your address”,
2. *free speech questions*, where the user speaks about any subject for approximately 10 seconds (**f**) or about 5 seconds (**r**), and
3. *predefined text* (**l**) that the user read out.

All of this data were captured on a mobile phone, except for the first session, where data were obtained using both a mobile phone and a laptop computer. One of the unique attributes of this database is that the acquisition device was held by the user, rather than being in a fixed position. As such, the microphone and camera are not fixed and used in an interactive and

uncontrolled manner. This presents several challenges such as high variability of pose and illumination conditions of the face, high variations in the quality of speech, and variability in terms of acoustics. Exemplary images of one subject are given in fig. 7.2.

This challenging mobile phone database has been used to evaluate several face and speaker authentication systems [Marcel et al., 2010] as well as bimodal authentication systems [Shen et al., 2010, Motlicek et al., 2012, Khoury et al., 2013a,b]. The database provides a defined protocol called *mobile-0*, which was initially described for the full database in [Wallace et al., 2011]. This protocol separates the clients of the database into three non-overlapping partitions for training, development and evaluation. The performance is measured in a gender-dependent manner (female and male, respectively). An overview of this initial protocol *mobile-0*, is provided in tab. 7.1. A limitation of this protocol is that only the lower quality biometric data acquired from the mobile phone was used, while the higher quality laptop data were ignored.

We rely on the three protocols introduced in [Khoury et al., 2013a] that explore mismatched conditions by making use of the laptop data. [2] The mismatched conditions that we wish to investigate are the specific cases of enrolling a user with high quality biometric samples (for instance acquired from a laptop computer) and then compared, or tested, using lower quality biometric samples obtained using a mobile phone. Details about the following protocols are given in tab. 7.1.

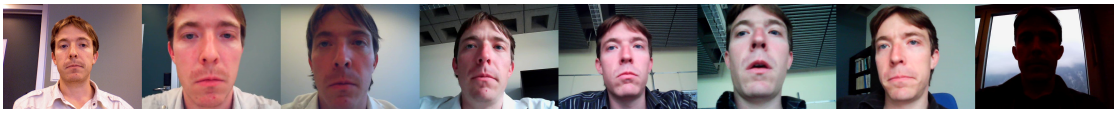


Figure 7.2 – IMAGE SAMPLES FROM THE MOBIO DATABASE. *This figure shows one image of the MOBIO database captured with a laptop on the left, and seven other images of the same subject captured with a mobile phone with significantly varying acquisition conditions.*

mobile-1 is identical to *mobile-0*, except that it includes the laptop data in the training set. This ensures that the same training data are being used for mobile and laptop evaluation (the next protocol). It provides additional 1,050 training samples compared to *mobile-0*. Enrollment and testing is conducted using only mobile phone data.

laptop-1 contains the same training data as *mobile-1*, but enrollment is performed exclusively using laptop data, while testing is conducted exclusively with mobile phone data.

laptop-mobile-1 also consists of the same training data as *mobile-1*. Here, enrollment is performed using both mobile and laptop data, while testing is still conducted exclusively on mobile phone data.

2. The MOBIO database (videos, still images, eye locations and the four evaluation protocols) are available for free at http://www.idiap.ch/dataset/mobio

Table 7.1 – MOBIO EVALUATION PROTOCOLS. *This table gives an overview of the data used in the protocols of the MOBIO database.*

Protocol	**Set**	**Phase I**			**Phase II**	**Nb. videos /client**	**Nb. videos**
		laptop data	mobile data	mobile data	mobile data		
		session-01	session-01	sessions 02-06	sessions 07-12		
		videos/client	videos/client	videos/client/sess.	videos/client/sess.		
mobile-0	Train	-	5**p**+10**f**+5**r**+1**l**	5**p**+10**f**+5**r**+1**l**	5**p**+5**f**+1**l**	192	9,600
	Enroll	-	5**p**	-	-	5	500
	Test	-	-	10**f** + 5**r**	5**f**	105	10,500
mobile-1	Train	5**p**+10**f**+5**r**+1**l**	5**p**+10**f**+5**r**+1**l**	5**p**+10**f**+5**r**+ 1**l**	5**p**+5**f**+1**l**	213	10,650
	Enroll	-	5**p**	-	-	5	500
	Test	-	-	10**f** + 5**r**	5**f**	105	10,500
laptop-1	Train	5**p**+10**f**+5**r**+1**l**	5**p**+10**f**+5**r**+1**l**	5**p**+10**f**+5**r**+ 1**l**	5**p**+5**f**+1**l**	213	10,650
	Enroll	5**p**	-	-	-	5	500
	Test	-	-	10**f** + 5**r**	5**f**	105	10,500
laptop-mobile-1	Train	5**p**+10**f**+5**r**+1**l**	5**p**+10**f**+5**r**+1**l**	5**p**+10**f**+5**r**+ 1**l**	5**p**+5**f**+1**l**	213	10,650
	Enroll	5**p**	5**p**	-	-	10	1,000
	Test	-	-	10**f** + 5**r**	5**f**	105	10,500

7.4 Systems Description

Considering the face modality, we employ the **GMM**, **ISV**, **JFA**, **TV-Cosine** and **TV-PLDA** systems, as previously described in tab. 5.9. For the speaker modality, we employ the systems described in tab. 6.2. There is only a minor difference in the voice activity detection, where the modulation peak around 4 Hz is employed in addition to the log energy [Scheirer and Slaney, 1997]. Next, the fusion between the systems is performed using linear logistic regression, as presented in the previous section.

Furthermore, the cohort set for ZT-norm score normalization is selected from the training data. Like the evaluation, this normalization is performed in a gender-dependent way. Two thirds are used for T-norm and the remaining third is used for Z-norm. For the T-models, we enroll one model per session to cope with the limited number of subjects in the cohort set.

7.5 Experimental Results

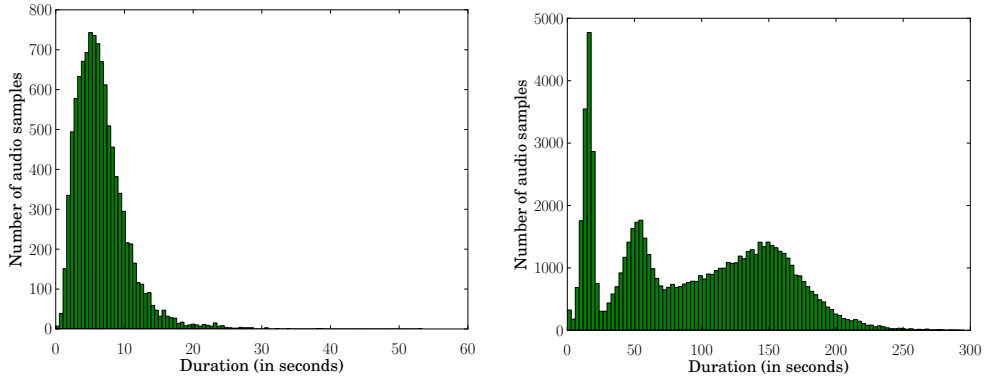
In this section, we evaluate the accuracy of the unimodal and bimodal authentication systems, across the three MOBIO protocols introduced in sec. 7.3. The techniques employed by these authentication systems were implemented in Bob [Anjos et al., 2012] (see appendix A). Again, all the results and the plots reported in this section can be easily regenerated using the satellite package [3] that accompanies this dissertation (see sec. A.3.2).

The dimensionality of the subspaces is tuned on the development set. We report both the EER on the development set and the HTER on the evaluation set. Results for the best systems for each modality are highlighted in bold. We also distinguish the best unimodal single algorithm systems by highlighting them in bold italics.

3. https://pypi.python.org/pypi/xbob.thesis.elshafey2014

7.5.1 Global Observations

Looking at the results provided in tab. 7.2 and tab. 7.3, two general trends are emerging. First, error rates on female subjects are higher than on male subjects. This might be due to the fact that the training set contains more men than women. Second, comparing the results of face authentication (**Face**) and speaker authentication (**Speaker**) systems, it is clear that error rates of **Face** systems are lower than the ones of **Speaker** systems. This is possibly caused by the fact that speech segments are relatively short. Indeed, the average duration of the probe samples of MOBIO after VAD is 6.64 s, while the average duration of the probe samples in NIST SRE12 is 93.8 s (see their distribution in fig. 7.3).



(a) Speech duration in MOBIO

(b) Speech duration in NIST SRE12

Figure 7.3 – SPEECH DURATION ON MOBIO AND NIST SRE12. *This figure compares the distributions of the speech duration of probe samples (after applying VAD) between the MOBIO database and the NIST SRE12 data.*

7.5.2 Comparison of the Modeling Techniques

In this section, our analysis focuses on the results of the *mobile-1* experiments, which are summarized in tab. 7.2. However, similar conclusions might be drawn from the experiments on the other two protocols, that are given in tab. 7.3.

It can be seen that **F-ISV** and **S-ISV** (rows 2 and 8) outperform **F-GMM** and **S-GMM** (rows 1 and 7). Indeed, **ISV** performs consistently well on both modalities, with a relative improvement of at least 17% on the evaluation set, for both male and female.

In contrast, **JFA** only outperforms **GMM** when considering the visual modality (row 3 against 1), while a degradation in performance is observed on the acoustic modality (row 9 against 7). Considering **TV**-based systems, results vary considerably depending on the scoring technique and the modality. **TV-Cosine** is more accurate than **TV-PLDA** when using visual cues (rows 4 and 5). In contrast, there is a slight advantage for **TV-PLDA** when using audio data (rows 11 and 10). Nevertheless, there is always at least one of the two systems that outperforms **GMM** on

Table 7.2 – PERFORMANCE SUMMARY ON THE *mobile-1* PROTOCOL OF MOBIO. *This table reports the* EER *(%) on the development set (*DEV*) and the* HTER *(%) on the evaluation set (*EVAL*) obtained with the* mobile-1 *protocol of MOBIO.*

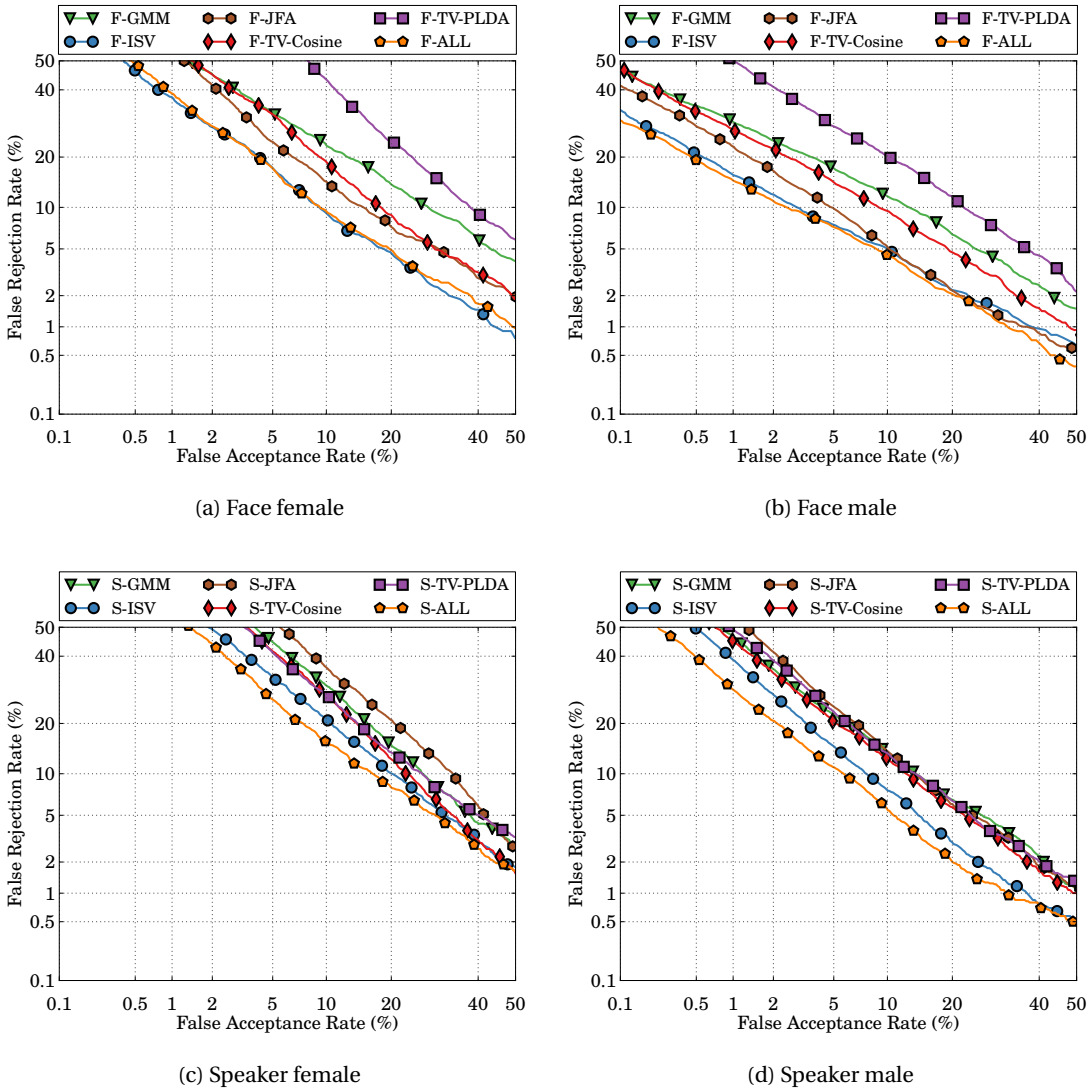
	Systems	Female		Male		
		DEV	EVAL	DEV	EVAL	
Face	**F-GMM**	10.63	18.57	8.77	11.30	1
	F-ISV	***6.35***	***11.89***	***3.41***	***6.19***	2
	F-JFA	8.10	15.81	5.28	7.20	3
	F-TV-Cosine	13.38	15.02	5.08	9.65	4
	F-TV-PLDA	21.59	22.07	10.63	15.21	5
	F-ALL	**5.82**	**11.79**	**3.02**	**6.15**	6
Speaker	**S-GMM**	19.51	17.98	14.96	12.03	7
	S-ISV	15.50	***14.93***	***13.57***	***8.82***	8
	S-JFA	18.90	24.12	14.60	12.78	9
	S-TV-Cosine	18.16	17.85	14.72	11.45	10
	S-TV-PLDA	***14.76***	16.99	16.38	11.55	11
	S-ALL	**11.96**	**12.90**	**11.23**	**7.75**	12
Bimodal	**B-GMM**	7.78	13.04	4.17	4.17	13
	B-ISV	***4.12***	***7.55***	***1.99***	***2.97***	14
	B-JFA	5.61	13.94	2.78	3.43	15
	B-TV-Cosine	8.98	9.77	2.85	4.50	16
	B-TV-PLDA	10.70	11.79	5.92	6.38	17
	B-ALL	**2.96**	**7.49**	**1.31**	**2.38**	18

Table 7.3 – PERFORMANCE SUMMARY ON THE *laptop-1* AND *laptop-mobile-1* PROTOCOLS OF MOBIO. *This table reports the* EER *(%) on the development set (*DEV*) and the* HTER *(%) on the evaluation set (*EVAL*) obtained with the* laptop-1 *and* laptop-mobile-1 *protocols of MOBIO.*

	Systems	*laptop-1*				*laptop-mobile-1*				
		Female		Male		Female		Male		
		DEV	EVAL	DEV	EVAL	DEV	EVAL	DEV	EVAL	
Face	**F-GMM**	18.99	21.03	12.98	18.07	10.42	17.54	7.66	11.13	1
	F-ISV	***11.49***	***13.43***	***6.51***	***9.64***	***4.97***	***11.12***	***2.90***	***5.58***	2
	F-JFA	15.08	15.58	8.02	10.63	8.25	12.13	4.05	6.69	3
	F-TV-Cosine	16.35	18.42	9.29	13.51	11.53	16.00	4.97	9.48	4
	F-TV-PLDA	22.90	23.15	18.62	18.13	20.65	19.83	12.19	14.56	5
	F-ALL	**10.95**	**13.24**	**5.83**	**8.76**	**4.71**	**11.75**	**2.54**	**5.68**	6
Speaker	**S-GMM**	18.83	19.17	16.23	14.65	16.67	16.39	12.74	10.24	7
	S-ISV	***14.81***	***15.88***	***14.16***	***11.50***	***12.21***	***13.46***	***10.40***	***8.63***	8
	S-JFA	20.78	17.72	17.25	15.14	14.22	18.12	12.03	10.64	9
	S-TV-Cosine	14.87	18.16	14.64	12.91	16.83	17.01	13.37	10.57	10
	S-TV-PLDA	15.34	20.29	16.47	14.98	13.65	16.32	13.58	11.92	11
	S-ALL	**11.52**	**14.62**	**11.15**	**9.51**	**9.09**	**11.63**	**8.26**	**6.95**	12
Bimodal	**B-GMM**	9.32	12.49	6.11	7.92	5.77	10.64	3.01	4.70	13
	B-ISV	***4.92***	***7.78***	***3.37***	***4.46***	***2.97***	***6.20***	***1.34***	***2.21***	14
	B-JFA	9.37	9.34	4.40	5.68	4.50	8.73	2.02	2.76	15
	B-TV-Cosine	7.57	11.66	4.89	6.65	7.04	9.36	2.73	4.34	16
	B-TV-PLDA	10.80	15.67	9.89	9.85	8.68	11.24	5.87	6.77	17
	B-ALL	**4.01**	**7.71**	**2.46**	**3.86**	**2.33**	**6.08**	**0.91**	**2.01**	18

each modality for both female and male. A possible explanation that **TV**-based systems are not consistently better is the limited number of subjects in the training set (50 subjects), whereas **TV** typically requires a significant amount of training data.

In addition, DET curves are given in fig. 7.4 to observe the behavior of the systems at various operating points. Considering the face modality, **JFA** is fairly accurate on a wide range of



(a) Face female

(b) Face male

(c) Speaker female

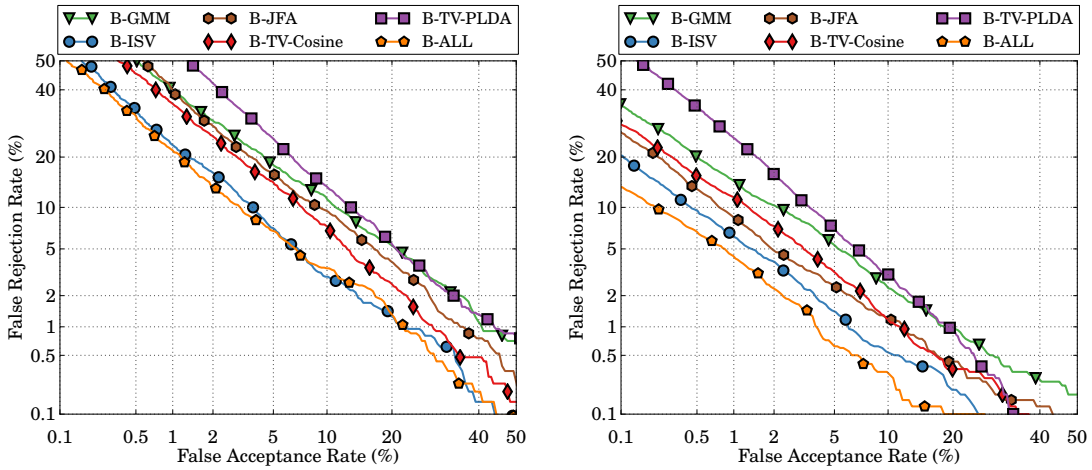
(d) Speaker male

Figure 7.4 – PERFORMANCE OF THE UNIMODAL SYSTEMS ON MOBIO. *This figure shows the DET curves of **GMM**, **ISV**, **JFA**, **TV-Cosine**, **TV-PLDA** and the multi-algorithm fusion system on the evaluation set of MOBIO* mobile-1.

operating points, with performance sometimes comparable to **ISV**. On the other hand, **S-TV-Cosine** and **S-TV-PLDA** offer a good accuracy on audio data, especially for female (fig. 7.4(a)).

7.5.3 Bimodal Authentication

In tab. 7.2 (rows 13 to 17) as well as on the DET curves in fig. 7.5, it can be seen that the bimodal **ISV** system (**B-ISV**) outperforms the other bimodal systems, **B-GMM**, **B-JFA**, **B-TV-Cosine** and **B-TV-PLDA**, on all operating points. Interestingly, the error rates significantly drop for all bimodal systems. For example, on the *mobile-1* female protocol the HTER of the **ISV** system



(a) Bimodal female

(b) Bimodal male

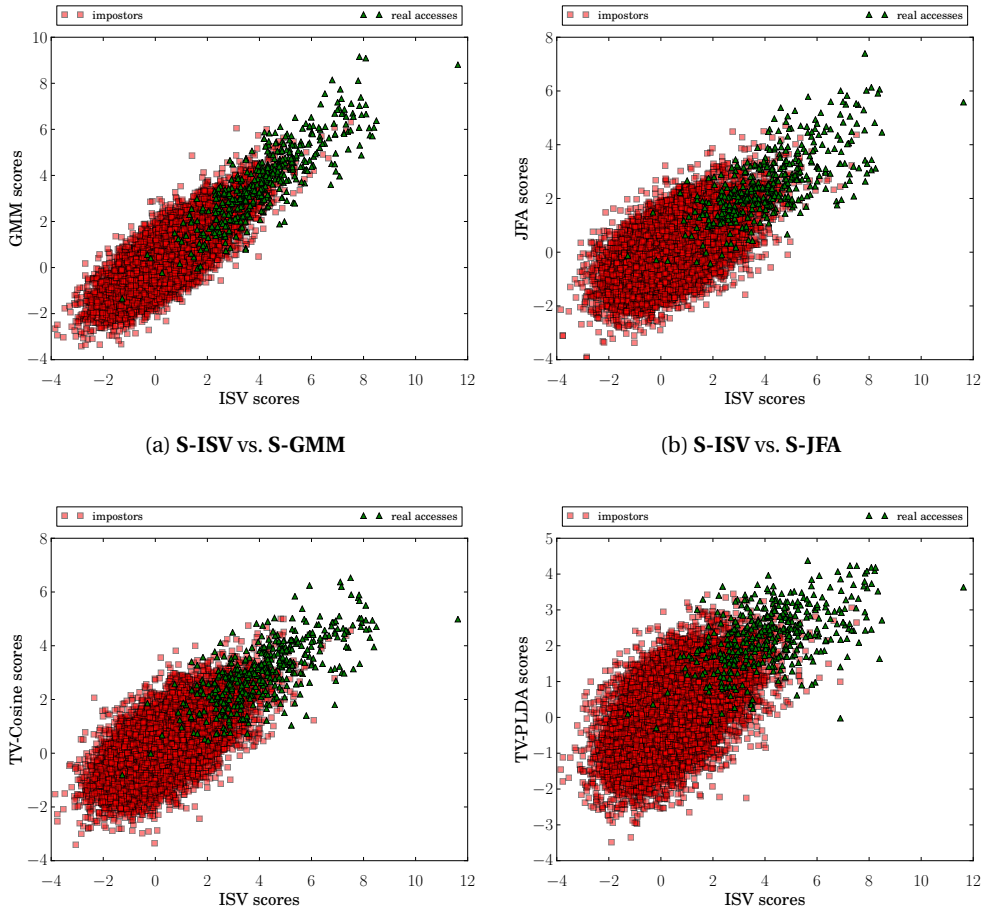
Figure 7.5 – PERFORMANCE OF THE BIMODAL SYSTEMS ON MOBIO. *This figure shows the DET curves of **B-GMM**, **B-ISV**, **B-JFA**, **B-TV-Cosine**, **B-TV-PLDA** and the bimodal multi-algorithm fusion system **B-ALL** on the evaluation set of MOBIO* mobile-1.

decreases from 11.9% (**F-ISV**) and 14.9% (**S-ISV**) to 7.6% (**B-ISV**), a relative performance gain of 36% compared to the best unimodal system. The results on the *mobile-1* male protocol are even more impressive with the HTER of the **ISV** system dropping from 6.2% (**F-ISV**) and 8.8% (**S-ISV**) to 3.0% (**B-ISV**), a relative performance gain of 52% compared to the best unimodal system. This improvement can be explained by the fact that visual and audio modalities are complementary: when a **Face** system fails to take the right decision because of image variability (illumination, head pose, etc.), a **Speaker** system is available to come to the rescue, and vice versa.

7.5.4 Multi-Algorithm Fusion

The fusion of multiple algorithms consistently outperforms single systems, as shown in tab. 7.2 (rows 6, 12 and 18). For example, the HTER of the **Speaker** system on the *mobile-1* male protocol drops from 8.8% (for **ISV**) to 7.8%, which corresponds to a relative improvement of 11%. The impact of the multi-algorithm fusion is higher for **Speaker** than **Face**, as **Speaker** obtains a relative improvement of on average 13% compared to 1% for **Face**. We attribute this larger gain in performance for **Speaker** to the fact that the second best system for **Speaker** (**TV-Cosine**) is more complementary with **ISV** than the second best system for **Face** (**JFA**). Finally, we note that the best bimodal multi-algorithm fusion system (**B-ALL**) outperforms the best unimodal **Face** (**F-ALL**) and **Speaker** (**S-ALL**) systems with a relative improvement of up to 61% and 69%, respectively (for male trials).

To explore the reason for the performance gains from multi-algorithm fusion we examine



(a) **S-ISV** vs. **S-GMM**

(b) **S-ISV** vs. **S-JFA**

(c) **S-ISV** vs. **S-TV-Cosine**

(d) **S-ISV** vs. **S-TV-PLDA**

Figure 7.6 – SCATTER PLOTS OF THE SCORES OF THE SPEAKER AUTHENTICATION SYSTEMS ON MOBIO. *This figure displays scatter plots of scores obtained with **ISV** against the four other speaker authentication systems (**GMM**, **JFA**, **TV-Cosine** and **TV-PLDA**) on the evaluation set of MOBIO* mobile-1 *male.*

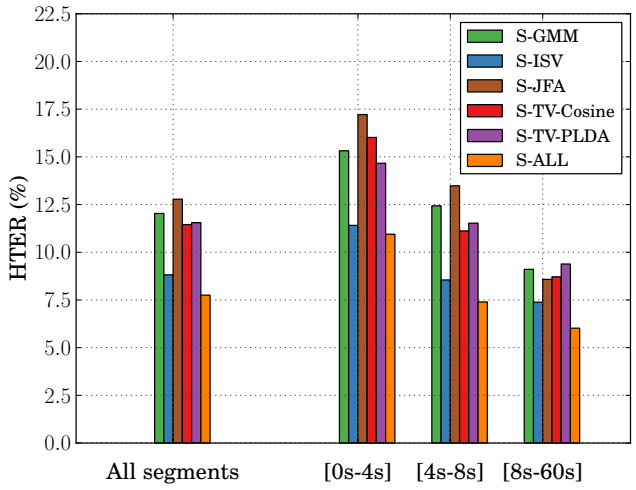
scatter plots of the scores of pairs of systems. Fig. 7.6 shows scatter plots that relate the **ISV** scores to the scores of the four other **Speaker** systems, **GMM**, **JFA**, **TV-Cosine** and **TV-PLDA** on the *mobile-1* male protocol. The scatter plots indicate that fusing **TV-PLDA** and **ISV** scores is a good strategy since the overlap between impostor and true claimant access trials is lower than, e.g., for **ISV** and **GMM**. The small overlap can be explained by the fact that the scoring method used for **TV-PLDA** is significantly different from the ones used for **ISV** and **GMM**. This is supported by the observation that **ISV** and **GMM** scores are more correlated (linear distribution of the points) than **TV-PLDA** and **ISV** scores (more widespread distribution).

In tab. 7.4 we present the *relative common error* (RCE) of multi-algorithm fusion systems (see sec. 7.2.2). Apparently, using audio data, **ISV** and **TV-PLDA** have the lowest percentage

Table 7.4 – RELATIVE COMMONS ERRORS WHEN PERFORMING MULTI-ALGORITHM FUSION ON MOBIO. *This table displays common errors (*CE*), relative common errors (*RCE*) and half total error rates (*HTER*) between systems on the* mobile-1 *male protocol. The column* ***All*** *corresponds to the five systems* ***GMM***, ***ISV***, ***JFA***, ***TV-Cosine*** *and* ***TV-PLDA*** *together.*

	Measure	ISV& GMM	ISV& JFA	ISV& TV-Cosine	ISV& TV-PLDA	All	
Face	CE	4,777	4,031	3,054	1,912	1,105	1
	RCE(%)	80.9	68.3	51.7	***32.4***	**18.7**	2
	HTER(%)	6.16	**6.11**	6.52	6.16	6.15	3
Speaker	CE	8,745	5,883	6511	5,565	2,504	4
	RCE(%)	64.6	47.8	50.9	***41.1***	**20.3**	5
	HTER(%)	9.13	8.72	8.32	**7.69**	7.75	6
Bimodal	CE	2,312	1,554	1,332	806	335	7
	RCE(%)	76.0	51.1	73.8	***26.5***	**11.0**	8
	HTER(%)	2.96	2.76	2.97	**2.38**	2.47	9

of common errors RCE = 41.1%, while **ISV** and **GMM** have the highest common error RCE = 64.6% (row 5). On the visual modality, the lowest percentage of common errors with **ISV** is obtained with **TV-PLDA**, followed by **TV-Cosine**. These results confirm that **TV** is a very helpful system for multi-algorithm fusion, when fusing a system with **ISV**. In addition, this table validates our hypothesis that a low percentage of relative common errors typically leads to an improved HTER (e.g., rows 2 and 3).



(a) HTER for different durations

Speech duration (in seconds)	Percentage of segments
[0 – 4]	22.1%
[4 – 8]	46.7%
[8 – 60]	31.2%

(b) Duration intervals

Figure 7.7 – IMPACT OF THE SPEECH DURATION ON THE SPEAKER AUTHENTICATION SYSTEMS. *This figure shows the performance of the speaker authentication systems with respect to the speech duration, as well as the distribution of the speech duration, on the evaluation set of MOBIO* mobile-1 *male.*

To better understand why multi-algorithm fusion significantly improves the results for **Speaker**, we group the audio probe files into three clusters according to their duration as seen in tab. 7.7(b). The HTER for each of the groups is displayed in fig. 7.7(a). Although **S-ISV** is the best system for any group of probe files, fig. 7.7(a) shows that the relative performance of the other systems depends on the duration of the probe samples. **S-TV-PLDA** is better than **S-TV-Cosine** and **S-JFA** for segments of short duration (< 4 s), followed by **S-GMM**. For segments of relatively long duration (> 8 s), **JFA** is then better than **GMM**, **TV-Cosine** and **TV-PLDA**. Finally, **TV-Cosine** is the best performing system among this four for segments of average duration (between 4 and 8 s). We believe that these observations are possible reasons for multi-algorithm fusion providing a significant boost in performance for **Speaker**.

7.5.5 Comparison of the Protocols

Fig. 7.8 displays the impact of enrollment condition mismatch on face, speaker and bimodal authentication. It shows that all the systems are affected by changing the enrollment conditions (between *mobile-1* and *laptop-1*). However, this impact is larger on **Face** than **Speaker** systems. Overall, for male subjects, the **Face** systems have a relative performance degradation of 41%, while **Speaker** systems lose 22%, on average. This shows that **Face** is more affected by condition mismatch of high versus low sample quality (see fig. 7.2). Besides, the most robust system is **S-TV-Cosine**, which suffers from a relative performance degradation of only 13%.

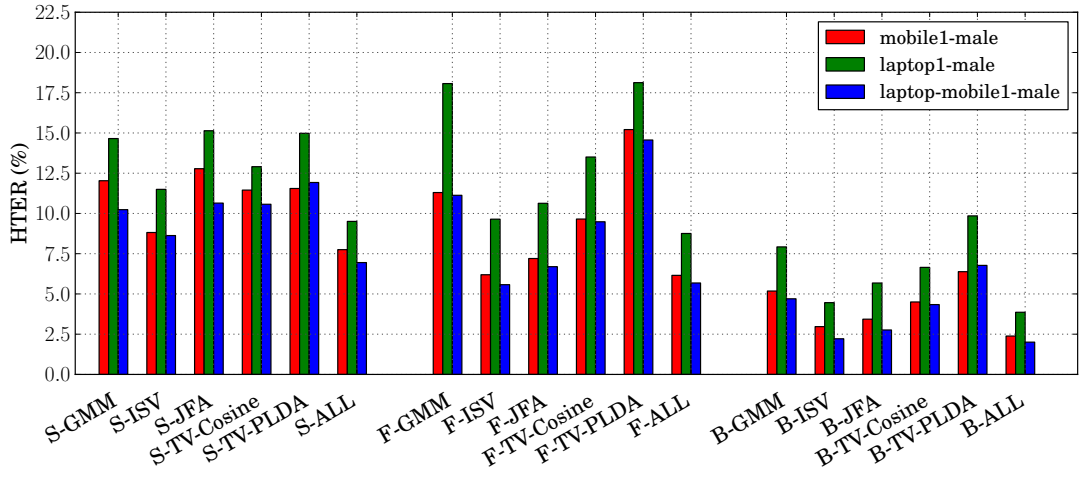


Figure 7.8 – BEHAVIOR OF THE AUTHENTICATION SYSTEMS ON THE DIFFERENT PROTOCOLS OF MOBIO. *This figure shows the* HTER *of the systems on the evaluation set across the different authentication protocols of MOBIO, for male subjects only.*

On the other hand, adding enrollment data as done in the *laptop-mobile-1* protocol improves the performance of almost all **Face** and **Speaker** systems, even though the additional laptop data is quite different to the mobile phone data. In fig. 7.8, the *laptop-mobile-1* protocol typically outperforms the other protocols. In particular, this is more visible for the acoustic modality than the visual one.

7.6 Summary and Concluding Remarks

In this chapter we studied the problem of face, speaker and bimodal authentication in the challenging mobile environment. The study was carried out on the MOBIO database, using three evaluation protocols.

The experimental findings can be summarized as:

1. **ISV** is the most accurate single algorithm system on this database for both face and speaker authentication, achieving a relative improvement in HTER of at least 17% on the *mobile-1* protocol when compared to the **GMM** baseline. A possible explanation is the limited number of subjects present in the training set (50 subjects), which negatively affects **TV**-based systems and **JFA**.
2. Considering the most accurate system (**ISV**), bimodal fusion brings a relative improvement in HTER of at least 36% on the *mobile-1* protocol. This shows the complementarity between acoustic and visual cues in challenging environments.
3. Multi-algorithm fusion provides a consistent performance improvement, particularly for the acoustic modality. This gain can be explained by the fact that the relative performance of the systems depends on the duration of the segments, **TV** being more accurate than **JFA** on short duration segments, but worse on long duration ones.
4. The proposed bimodal multi-algorithm fusion system (**B-ALL**) consistently leads to good performances across the different evaluation protocols. In particular, a relative improvement in HTER of at least 61% is observed when compared to either of the unimodal systems.
5. Face, speaker and bimodal authentication are adversely affected by the significant mismatch between enrollment and test conditions defined in the *laptop-1* protocol. However, the use of additional enrollment data allows to circumvent this problem, as performed when using the *laptop-mobile-1* protocol.

8 Conclusions and Future Work

Automatic face and speaker recognition are usually addressed in completely different ways by the biometric community. However, both tasks are affected by similar problems. While current recognition systems work well in controlled laboratory-like conditions, their performance is strongly affected in challenging real world scenarios. In the latter case, there is typically a mismatch between enrollment and testing conditions, which is coined as session variability.

In this thesis we addressed this problem of session variability using a set of probabilistic models.
First, we considered three approaches derived from Gaussian mixture models (GMM). Two of them, inter-session variability (ISV) modeling and joint factor analysis (JFA), compensate for this mismatch during enrollment as well as testing. The third approach, total variability (TV) modeling, is a front-end extractor that models the variability of the samples globally. Besides, the similarities and the differences between these techniques were assessed both theoretically and empirically.
Next, we presented a scalable formulation of probabilistic linear discriminant analysis (PLDA), an approach that separately models between-class and within-class variations. This formulation is exact and significantly improves the time and memory complexity both at training and test time.
Finally, all these models are scalable and efficient implementations of these techniques were implemented within Bob, an open source framework for signal processing and machine learning developed during my thesis (cf. appendix A).

8.1 Experimental Findings

The proposed techniques were applied to three different tasks: face, speaker and bimodal recognition. Each task was discussed in detail and performances were reported on at least one large database (for each task) using well-defined evaluation protocols. Furthermore, a satellite package of Bob was implemented, which allows to reproduce all the results and plots reported in this dissertation (cf. sec. A.3.2).

We list the experimental findings below.

1. PLDA modeling outperforms several approaches when employed in challenging environments (cf. sec. 5.4.2 and sec. 6.4). In particular, its combination with TV leads to a very accurate system for both face and speaker recognition. However, it requires a sufficiently large training set, both in terms of samples and classes, to be able to capture and to model between-class and within-class variabilities properly. Furthermore, both PLDA-based face recognition systems, **TV-PLDA** and **SIFT-PLDA**, are very successful on the uncontrolled Labeled Faces in the Wild database, which highlights the wide applicability of this approach (cf. sec. 5.4.2).
2. The three GMM-based session variability modeling techniques, ISV, JFA and TV, bring significant improvements when compared to the **GMM** baseline. This has been observed empirically on three different tasks: face, speaker and bimodal recognition (cf. sec. 5.4, sec. 6.4 and sec. 7.5, respectively). In particular, ISV is performing particularly well in a wide range of scenarios.
3. Considering the face recognition task:
   (a) When there is a mismatch between enrollment and testing conditions, head pose variations lead to more degradations in performance than facial expressions or illumination variations (cf. sec. 5.4.1). Experimentally, it was found that the best two systems in case of pose variations are **TV-PLDA** and **JFA** (cf. sec. 5.4.1), providing a relative improvement in HTER of more than 50%.
   (b) Occlusions and especially sunglasses strongly affect the performance of all the systems (cf. sec. 5.4.1). This confirms that the eye region is very important when discriminating people.
   (c) **TV**-based systems and **ISV** perform very well on the large FRGC database (cf. sec. 5.4.2). For instance, a relative improvement in the CAR at FAR = 0.1% of 33% is observed with **TV-PLDA** on experiment *2.0.1* when compared to **GMM**.
   (d) Performances of state-of-the-art systems are still too low to allow face identification in uncontrolled environments on a large database (cf. sec. 5.4.2), as required by e.g., forensic investigation applications.
4. Considering the speaker recognition task, it was shown that the most accurate system on the large NIST SRE12 corpus is **TV-PLDA**, followed by **ISV** (cf. sec. 6.4.1). In particular, a relative improvement in HTER of up to 47% is observed with **TV-PLDA**, compared to the **GMM** baseline (cf. sec. 6.4.2). However, the challenging conditions encountered in this database strongly affect the performances of all the systems.
5. Finally, considering the bimodal recognition task:
   (a) Acoustic and visual cues are highly complementary in challenging environments. Considering the most accurate system on MOBIO (**ISV**), bimodal fusion brings a relative improvement in HTER of at least 36% on the *mobile-1* protocol (cf. sec. 7.5.3).

(b) Multi-algorithm fusion is of particular interest for the acoustic modality when test samples have various durations. This is explained by the fact that the relative performance of the proposed systems depends on the duration of the samples (cf. sec. 7.5.4).

(c) The performance of the systems is negatively impacted when there is a mismatch between enrollment and testing conditions. Using additional enrollment data is one way to address this problem (cf. sec. 7.5.5).

8.2 Directions for Future Work

The following are some directions for future research extending beyond this dissertation.

On a short-term basis:

1. The proposed techniques are generic. They could, hence, be applied to other classification tasks, for example, to other biometric modalities or to visual object recognition in general. Furthermore, these techniques can be fed by any types of features. In particular, features extracted at multiple scales might be employed for the GMM-based techniques instead of simple DCT-like features.
2. For the task of automatic face recognition, the problem of pose still needs to be addressed. One possibility would be to use three-dimensional models at enrollment time, before estimating the pose of the probe sample at test time, and matching it with a suitable representation computed from the enrolled model.
   In addition, techniques to reduce the impact of occlusions, which significantly affect the accuracy of the systems in challenging uncontrolled environment, might be investigated, e.g., by detecting sunglasses prior to feature extraction and/or classification.
3. For the task of speaker recognition, an analysis of the impact of speaker's voice variability (e.g., mood, illness, age, etc.) and the environment (e.g., background noise, microphones, transmission channels, etc.) on state-of-the-art automatic systems might be performed. This would help to clearly identify the bottlenecks to improve the accuracy, and would require a large and recent database with controlled variations.

On a longer-term basis:

4. The scalable formulation of probabilistic linear discriminant analysis might be extended to the case of mixtures of such models. Furthermore, the model could be adapted to other probability distributions, instead of Gaussian ones.
5. The GMM-based session variability modeling techniques could be revisited to jointly model both the data and the variability using a single training process, rather than first modeling the data using GMMs and then the session variability.
6. For the task of bimodal (face and speaker) recognition, the use of the interdependency of a person's spoken utterance and the associated facial movements might be inspected to yield further improvements.

A Bob: a Free Machine Learning and Signal Processing Toolbox

In this appendix, we introduce *Bob*, a free machine learning and signal processing library, which was developed during my thesis. This is a collaborative, easy to use and extensible toolbox, which provides both efficient implementations of several machine learning algorithms as well as a framework to help researchers to publish reproducible research, thanks to its concept of satellite packages. Furthermore, all the modeling techniques described in this thesis were implemented as parts of this toolbox.

A.1 Introduction

Rapid prototyping and testing of novel ideas are the main software requirements of researchers. To increase accessibility to a larger pool of academics and research parties, one must also consider many other aspects while choosing an implementation framework: clarity, simplicity, good documentation, unit testing, efficient use of resources (especially for large-scale experiments), open sourceness and extensibility are among the most important ones. *Bob*[1] is an open source machine learning (ML) and signal processing (SP) library based on an ongoing community effort, which is designed to meet all these requirements.

A number of open source libraries for ML exist in literature. Examples are Java-ML [Abeel et al., 2009], scikit-learn [Pedregosa et al., 2011], Shark [Igel et al., 2008], Torch3 [Collobert et al., 2002], and Torch7 [Collobert et al., 2011]. However, most of them do not provide a complete set of tools for managing all aspects of research experimentation, including database interfaces, plotting utilities and clean implementation for speeding up identified bottlenecks. Among these libraries, Torch3 is notable for its original conceptual design of ML algorithms as a **Dataset**, **Machine** or **Trainer**, where the **Trainer** uses data from the **Dataset** to train the **Machine**.

Bob is a toolbox that (1) gathers a large set of ML and SP tools, (2) takes advantage of well-tested ML concepts from Torch3, (3) provides a researcher friendly Python environment for rapid

1. https://www.idiap.ch/software/bob

development, (4) offers fast C++ implementations of identified bottlenecks, (5) emphasizes on code clarity, documentation, tutorials and thorough testing and (6) allows easy extensions through its concept of satellite packages.

The remainder of this appendix is structured as follows. First, we present the main features of Bob. Next, we describe how researchers can easily extend Bob and share their work in a reproducible way. In particular, we introduce the satellite package associated with this dissertation.

A.2 Overview

Bob is supported on several Linux distributions as well as on Apple OS X. A Microsoft Windows support is planned. It gathers a large set of tools and interfaces implemented in both Python and C++. Python programming allows fast laboratory-like (Matlab-like) development and testing by using scriptable constructions, plotting and built-in reference documentation. Constructions in C++ are exclusively used when developers are faced to speed bottlenecks in their Python scripts. Bob only requires third-party open source software, such as LAPACK [Anderson et al., 1999] or LibSVM [Chang and Lin, 2011].

Multi-dimensional arrays are the main objects used to represent data in Bob. Data can originate from images, audio signals, or features of various kinds. For C++ code, this is achieved by using Blitz++ [Veldhuizen, 1998], whereas for Python code, this is guaranteed by the use of NumPy [Oliphant, 2007]. Data between C++ and Python are exchanged in a transparent and efficient way using a customized code bridge based on the Boost template libraries [Demming and Duffy, 2010].

The code in Bob is organized into modules with loose layering as shown in fig. A.1. This diagram also illustrates the major features provided through Bob's modules: machine learning, math and signal processing, image processing, audio processing, input and output, database support, and performance evaluation.

The *machine learning* modules (**machine; trainer**) include ML algorithms using both generative approaches, discriminative approaches (e.g., multilayer perceptron, support vector machine), data clustering (e.g., k-means), and dimensionality reduction (e.g., principal component analysis, linear discriminant analysis). In particular, the modeling techniques described in chapter 3 (Gaussian mixture model, inter-session variability modeling, joint factor analysis and total variability modeling) and chapter 4 (probabilistic linear discriminant analysis) were all implemented and integrated into this module. The *math and signal processing* modules (**math; sp**) include basic mathematical tools such as eigenvalue decomposition, matrix inversion, and fast Fourier transform (FFT). The *image and audio processing* modules (**ip; ap**) include visual filtering (e.g., Gaussian, median, Gabor), visual features (e.g., SIFT, HOG, LBP), acoustic features (e.g., MFCC, spectrogram) as well as normalization routines. The *input/output* module (**io**) includes features to handle the Hierarchical Data Format version 5

(HDF5) [The HDF Group, 2000-2010] format, which was chosen as default for storing and managing data because of its versatility and portability across different platforms and frameworks. In addition, Bob supports several image, audio and video formats, as well as the Matlab (*.mat*) format. The *database* support module (**db**) provides features to easily query and interface with database protocols for reproducible experimentation. The *performance evaluation* module (**measure**) includes several standard metrics such as the equal error rate (EER), receiver operating characteristic (ROC) and detection error tradeoff (DET) curves.

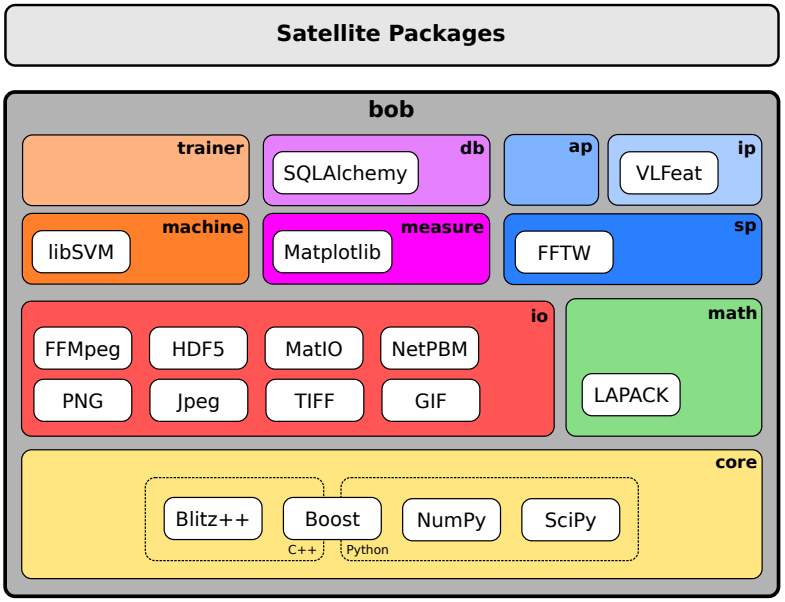


Figure A.1 – Internal Software Organization of the Modules of Bob.

A.3 Reproducible Research and Extensions through Satellite Packages

One of the main motivations in the development of Bob is to foster and to ease *reproducible research* [Price, 1986]. Reproducibility affects impact and visibility of scientific work and should be considered a key aspect of research [Vandewalle, 2012]. Being able to easily rerun and extend the experiments of scientific articles allows researchers to quickly test their own ideas reusing existing tools rather than spending considerable amount of time reimplementing previously developed concepts, focusing on the problem to be solved rather than on the solution. Replicability often involves a publicly available dataset, data accessors that specify how the data should be used, source code (machine learning algorithms, feature extractors and metrics), as well as infrastructure (normally these are *scripts*) to glue all these bricks together and support experimental analysis.

To satisfy these needs, Bob is hosted on GitHub,[2] a collaborative web-based platform for

2. https://github.com/idiap/bob

software development projects that uses the Git revision control system. This allows any user to inspect existing algorithm implementations, to access the documentation, to report bugs, to request features and to provide new patches and features through *pull requests*. In addition, Bob offers the possibility for any researcher to extend the library with new features such as machine learning algorithms. This relies on the concept of *satellite packages*, which may contain source code and documentation for a proposed algorithm, data accessors or scripts to replicate findings.

A.3.1 Example 1: L-BFGS-based Training for Multilayer Perceptrons (MLP)

In the following, we propose a typical example where satellite packages are developed on top of Bob for reproducible research purposes. For this didactic example, the research idea consists of using the L-BFGS optimization technique [Byrd et al., 1995] to train the parameters of an MLP, and to compare it against the R-prop [Riedmiller and Braun, 1993] approach that is available in Bob. For this purpose, experiments are conducted on the MNIST database of handwritten digits [LeCun et al., 1998]. One of the main outcomes of this example is the performance reported on fig. A.2.

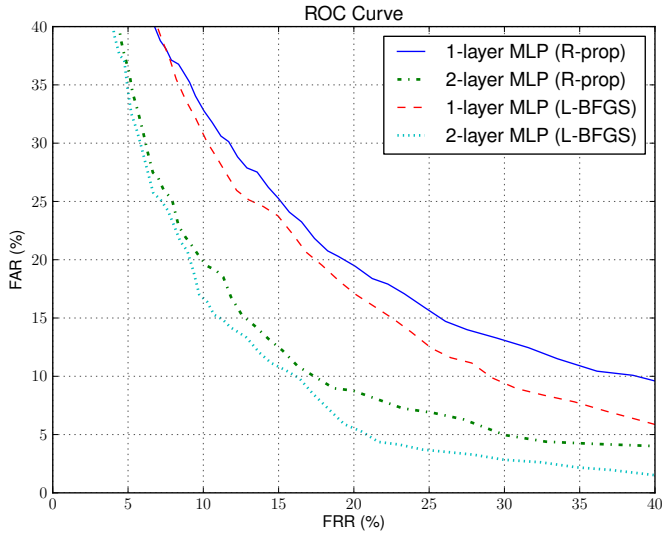


Figure A.2 – PERFORMANCE OF MLP CLASSIFIERS TRAINED USING R-PROP OR L-BFGS. *This figure shows the ROC curves of MLP classifiers trained using R-Prop or L-BFGS evaluated on the test set of the MNIST database of handwritten digits.*

Three satellite packages built on top of Bob have been implemented to address this goal. First, `xbob.db.mnist` [3] provides functionalities to access MNIST samples from Python, respecting the imposed training and testing sets defined in the original protocol. Second, `xbob.mlp.lbfgs` [4] demonstrates how to extend Bob's machine learning features by providing

3. https://pypi.python.org/pypi/xbob.db.mnist
4. https://pypi.python.org/pypi/xbob.mlp.lbfgs

a new L-BFGS-based trainer to the existing MLP implementation. In particular, this implementation relies on the existing MLP machine implementation of Bob, as well as on an existing MLP base trainer class of Bob, which performs forward and backward propagation. This leads to a very simple python implementation for this L-BFGS training procedure that solely focuses on the optimization aspect. Finally, `xbob.paper.example` [5] glues the previous two packages together by providing instructions on how to use the scripts to train MLPs using either the R-Prop trainer of Bob or the proposed L-BFGS-based trainer and to replicate the ROC curves shown on fig. A.2.

These satellite packages are hosted on GitHub to facilitate maintenance and sharing. Besides, releases can be shared and made easy to install on multiple architectures, thanks to the Python Package Index. [6] A list of existing satellite packages as well as detailed instructions about satellite package creation are available online. [7]

A.3.2 Example 2: Satellite Package to Reproduce the Results and Plots of this Dissertation

In this Ph.D. dissertation, we reported results and plots from experiments conducted using several machine learning algorithms on several databases. To satisfy the need of reproducibility, we implemented a satellite package `xbob.thesis.elshafey2014` [8] that provides an environment as well as the scripts required to regenerate any results or plots provided in this manuscript. [9]

This satellite package depends on several other open source packages or softwares, and plays the role of connecting these entities together with the aim of being able to regenerate any plot from the raw data of a database. For instance, all the feature extractors and machine learning algorithms are implemented within Bob, which is, hence, a dependency of this package. In addition, face and speaker recognition toolchains are implemented in two other dependencies `facereclib` [10] [Günther et al., 2012b] and `xbob.spkrec` [11] [Khoury et al., 2014], respectively. These toolchains allow to conduct full experiments, from the segmentation step to the generation of identification or verification trial scores. Furthermore, the evaluation protocols are defined in satellite packages separately for each database. For instance, `xbob.db.multipie` [12] and `xbob.db.nist_sre12` [13] specify the evaluation protocols on a face and speaker recognition database, respectively.

5. https://pypi.python.org/pypi/xbob.paper.example
6. https://pypi.python.org/
7. https://github.com/idiap/bob/wiki/Satellite-Packages
8. https://pypi.python.org/pypi/xbob.thesis.elshafey2014
9. There are two notable exceptions, which are the plots and results given in the appendices. Fig. A.2 should be generated using the satellite package `xbob.paper.example`, whereas the results of appendix B should be generated using the satellite package `xbob.gender.bimodal`.
10. https://pypi.python.org/pypi/facereclib
11. https://pypi.python.org/pypi/xbob.spkrec
12. https://pypi.python.org/pypi/xbob.db.multipie
13. https://pypi.python.org/pypi/xbob.db.nist_sre12

The package is organized around several steps as follows. First, the user downloads the satellite package associated with this dissertation. Second, a script allows to automatically download and configure all the dependencies (except Bob and the CSU baselines, which have to be installed manually). Third, the user is required to download the databases and to set the location of the data within several configuration files. Next, a script allows to generate the trial scores using an evaluation protocol on a specific database for all the systems considered. Finally, another script expects a subset of these trial scores as input, and returns a plot or a table of results as output. Detailed instructions are provided within the package.

A.4 Conclusions

We presented Bob, a free and extensible ML and SP toolbox. Bob is written in a mix of Python and C++, to provide both fast C++ implementations of identified bottlenecks and a friendly Python environment for rapid development. It is organized as a set of modules, which consist of self-explanatory code including documentation and unit testing, and which only require open source third-party libraries. Bob supports a wide range of platforms, including open source distributions. Finally, the toolbox can be easily extended by researchers, thanks to its concept of satellite packages, which foster reproducible research.

In addition, we presented `xbob.thesis.elshafey2014`, the satellite package, which accompanied this dissertation, defining an environment as well as scripts to regenerate any results or plots given in this manuscript.

B Audio-Visual Gender Recognition

In this appendix, we apply several variability modeling techniques to the task of gender recognition. More precisely, we explore the use of Gaussian mixture models (GMM), inter-session variability (ISV) modeling and total variability (TV) modeling for both unimodal (visual or audio) and bimodal gender recognition.

B.1 Introduction

Information about gender, age, ethnicity, and emotional state are important ingredients that lead to rich behavioral informatics. Such information can be extracted from visual or audio modalities. In this appendix, we focus on the problem of gender recognition using both visual and audio cues.

Automatic gender recognition is crucial for a number of applications of human-computer or human-robot interaction. It serves to (1) enrich the metadata of visual and audio documents in an indexing and retrieval system, (2) improve the efficiency and the accuracy of person (both face and speaker) recognition, diarization and surveillance systems by reducing the search space to subjects from the same gender, and by building gender-dependent models, which are often better than gender-independent models, (3) enhance human-machine interaction by suggesting user-friendly interface (e.g., gaming, social networks) and personalized advertisements (e.g., interactive voice response system, in-store cameras), (4) increase the intelligibility of human-robot interaction, and (5) collect passive demographic data.

Due to these various applications, the problem of automatic gender recognition has recently received significant attention. Researchers have often addressed this problem with a unimodal aspect. For visual-based gender recognition, readers can refer to [Moghaddam and Yang, 2002, Sun et al., 2006, Mäkinen and Raisamo, 2008a,b, Alexandre, 2010]. For audio-based gender recognition, one can cite the work of [Harb and Chen, 2003, Hu et al., 2012, DeMarco and Cox, 2011, Burkhardt et al., 2010, Schuller et al., 2010, Li et al., 2013]. In contrast, only few existing works (e.g., [Liu et al., 2007, Pronobis and Magimai-Doss, 2008]) have taken into account

both visual and audio cues to solve the problem of gender recognition. They have shown that audio-visual fusion can improve the accuracy of gender classification systems especially under degraded conditions and temporal unavailability of one of the modalities. However, their evaluations were conducted on in-house or small databases, using simplistic unimodal systems.

In this appendix, we explore the total variability (TV) and inter-session variability (ISV) modeling techniques (see chapter 3) for both speech-based and face-based gender recognition problems. We apply linear logistic regression (see sec. 7.2.2) to combine the two modalities at the decision level. The proposed systems are compared to several unimodal and bimodal state-of-the-art algorithms. The experimental evaluation is conducted on the FERET and LFW databases for visual-based, on the NIST-SRE database for audio-based and on the MOBIO dataset for audio-visual gender recognition.

B.1.1 Gender Recognition from Images

Gender recognition from images of faces has recently received significant attention, and several approaches were explored. In [Moghaddam and Yang, 2002], the authors show that support vector machines (SVMs) are superior to linear discriminant analysis (LDA), nearest-neighbor and radial basis function networks. They conduct their experiments on images selected from the FERET database [Phillips et al., 2000]. However, the experimental protocol suffers from a lack of information that prevents the reproducibility of the results.

[Mäkinen and Raisamo, 2008a] put some effort towards publishing the details about the protocol used on FERET, and, thus, help to benchmark the different approaches. One of the findings of their work is that SVMs (on raw pixels) are slightly superior to neural networks (on raw pixels) and AdaBoost (on Haar-like features). [Alexandre, 2010] proposes an approach that combines several SVM classifiers applied on intensity, shape and texture features gathered at different scales. This approach obtains a better accuracy than the techniques presented in [Mäkinen and Raisamo, 2008a]. The drawback of the FERET database is that the images are acquired in controlled conditions, and the dataset corresponding to the available protocol [Mäkinen and Raisamo, 2008a] is small (only 411 face images).

[Gallagher and Chen, 2009] use contextual features to recognize people's gender in images of groups of people (family portraits, wedding photos, etc.). Their images were collected from Flickr (uncontrolled conditions) and made available for researchers [Gallagher and Chen, 2008]. However, they do not provide a standard evaluation protocol.

More recently an evaluation protocol [1] on the Labeled Faces in the Wild (LFW) database was proposed as one of the BeFIT (Benchmarking Facial Image Analysis Technologies) challenges. As for Gallagher's database, LFW images are acquired in realistic scenarios under large variability in illumination, facial expressions and head pose (see sec. 5.2.2). [Dago-Casas et al.,

1. http://face.cs.kit.edu/431.php

2011] use this protocol to evaluate state-of-the-art systems. They found that Gabor jets and local binary patterns (LBPs) obtain similar accuracies, and perform generally better than raw pixels. They also found that SVMs work slightly better than LDA.

B.1.2 Gender Recognition from Speech

Several approaches were proposed to cope with the problem of speech-based gender recognition. In [Harb and Chen, 2003], Mel frequency spectral coefficients (MFSC) with neural networks are used. Their database was collected from French and English radio stations. However, the details needed to replicate the experiments are not provided. [Hu et al., 2012] propose a two-stage classifier where pitch thresholding is applied in the first stage, and Mel frequency cepstral coefficients (MFCC) extraction followed by GMM-based classification is done in the second stage. [DeMarco and Cox, 2011] describe an unsupervised system that jointly uses MFCC-based and pitch-based classifiers. Any disagreement between the two classifiers is then resolved by using a pitch-shifting mechanism. In both [Hu et al., 2012] and [DeMarco and Cox, 2011], the experiments are conducted on clean data (TIDIGITS in [Hu et al., 2012] and TIMIT in [DeMarco and Cox, 2011]). This explains the high accuracies reported in their work.

In 2010, a challenge on gender and age detection was conducted [Schuller et al., 2010] on the *aGender* database [Burkhardt et al., 2010], which contains recordings from German telephone speech. Several gender recognition algorithms were explored on this database such as GMM, SVM, MLP, GMM-Mean-SVM, GMM-MLLR-SVM, using both prosodic and acoustic features. Readers can refer to the work in [Li et al., 2013] where seven sub-systems based on SVM and GMM are evaluated and combined. aGender contains one group of children speakers. However, its evaluation protocol does not distinguish between female and male children. This makes it difficult to be used independently for gender recognition.

In [Kockmann et al., 2010], the use of JFA was investigated. However, none of the recently proposed ISV and TV modeling techniques were explored.

B.1.3 Audio-Visual Gender Recognition

Contrarily to unimodal gender recognition, audio-visual gender recognition has not been well explored in the literature. To the best of our knowledge, the first attempt of recognizing gender using acoustic and visual cues was done in [Walawalkar et al., 2003]. In this work, the authors found that SVMs classify better than nearest-neighbor and k-nearest neighbors. The main drawback of their work is that they use two separate unimodal databases to compare their audio and visual systems. This prevents them from making an objective and fair comparison between the two unimodal systems, and furthermore, it prevents them from combining both modalities to improve the performance of their system.

This issue was partially solved in [Liu et al., 2007] where an audio-visual database is used. In their work, the audio-based gender classifier relies on GMM, whereas the visual-based

classifier relies on SVM. The acoustic features used are the MFCC coefficients and their first derivatives, while the visual features used are the intensities of the pixels. At the fusion level, they combine the incompatible scores (posterior probability for GMM and distances for SVM) from the two classifiers using a naive linear combination that is tuned directly on the test set. They reported gender classification accuracies of 85%, 84.75% and 91.25% on audio, visual and audio-visual cues, respectively. The main drawback of this work is the use of a private database without giving the full details about the conditions in which the data were collected.

B.2 Proposed Audio-Visual Gender Recognition

In this appendix, we address the task of gender recognition by modeling the feature distribution using GMMs. Several classification and extraction techniques can be applied on top of this modeling to both visual and audio modalities. One possibility is to rely on the generative probabilistic framework for classification based on GMMs, introduced for speaker recognition in [Reynolds and Rose, 1995, Reynolds et al., 2000] and then successfully applied to speech-based gender recognition [Li et al., 2013]. Furthermore, to cope with the problem of high intra-class variability, we additionally investigate two recent session variability modeling techniques derived from GMMs: ISV and JFA. To the best of our knowledge, none of these two methods were used for gender recognition.

All these techniques have been described in details in chapter 3. We provide a succinct description in the remainder of this section.

B.2.1 Feature Distribution Modeling using GMM

Two separate feature extraction processes are employed for images and audio samples. For both visual and audio data, we employ similar DCT-based and MFCC-based techniques as the ones described in chapter 7.

B.2.2 Gaussian Mixture Modeling

To use GMMs for gender recognition, we need to train a GMM $\mathscr{S}_i$ for both genders ($i \in \{\text{male}, \text{female}\}$) from a set of enrollment samples. There are different ways to train GMMs. We employ the expectation-maximization algorithm to seek a maximum-likelihood estimate for each class (cf. sec. 3.3.2). Once gender-specific models, $\mathscr{S}_{\text{female}}$ and $\mathscr{S}_{\text{male}}$, are enrolled, the probability that a test sample $\boldsymbol{\chi}_{\text{test}}$ is from the class male is given by a log-likelihood ratio (LLR) score:

$$h_{\mathbf{GMM}}\left(\boldsymbol{\chi}_{\text{test}}\right) = \ln P\left(\boldsymbol{\chi}_{\text{test}} \mid \mathscr{S}_{\text{male}}\right) - \ln P\left(\boldsymbol{\chi}_{\text{test}} \mid \mathscr{S}_{\text{female}}\right). \tag{B.1}$$

B.2.3 Inter-Session Variability Modeling

A powerful approach that relies on a UBM $\boldsymbol{\mu}$ (see sec. 3.3.2) is inter-session variability (ISV) modeling (cf. sec. 3.4). It aims to estimate and suppress the effects of within-class variations in order to create more discriminant gender models.

ISV assumes that session variability results in an additive offset to the mean supervector $\boldsymbol{s}_i$ of the gender model. This offset can be added directly to the normal mean-only MAP adaptation representation. Given a sample $\boldsymbol{\chi}$, the mean supervector $\boldsymbol{\mu}$ of the GMM that best represents this sample is:

$$\boldsymbol{\mu} = \boldsymbol{m} + \boldsymbol{U}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{z}, \tag{B.2}$$

where $\boldsymbol{U}$ is a subspace that constrains the possible session effects, $\boldsymbol{x}$ is its associated latent session variable, while $\boldsymbol{D}\boldsymbol{z}$ represents the gender-specific offset (cf. sec. 3.4).

Similarly to GMM, ISV scoring relies on a LLR, using compensated GMMs as follows:

$$h_{\text{ISV}}\left(\boldsymbol{\chi}_{\text{test}}\right) = \ln \frac{P\left(\boldsymbol{\chi}_{\text{test}} \mid \boldsymbol{m} + \boldsymbol{U}\boldsymbol{x}_{\text{male}} + \boldsymbol{D}\boldsymbol{z}_{\text{male}}\right)}{P\left(\boldsymbol{\chi}_{\text{test}} \mid \boldsymbol{m} + \boldsymbol{U}\boldsymbol{x}_{\text{female}} + \boldsymbol{D}\boldsymbol{z}_{\text{female}}\right)} \tag{B.3}$$

B.2.4 Total Variability Modeling

The second approach we investigate to address the problem of gender recognition is total variability (TV) modeling (cf. sec. 3.5).

TV modeling aims to extract low-dimensional factors $\boldsymbol{v}$, so-called *i-vectors*, from samples $\boldsymbol{\chi}$. More formally, TV can be described in the GMM mean supervector space by:

$$\boldsymbol{\mu} = \boldsymbol{m} + \boldsymbol{T}\boldsymbol{v}, \tag{B.4}$$

where $\boldsymbol{\mu}$ is the mean supervector of the GMM that best represents the sample, $\boldsymbol{T}$ the low-dimensional total variability subspace and $\boldsymbol{v}$ the low-dimensional i-vector. $\boldsymbol{T}$ is learned by maximizing the likelihood over a large training set (see sec. 3.5.1).

In contrast to ISV, TV does not explicitly perform session compensation and scoring. Therefore, we employed the preprocessing and session compensation algorithms given in sec. 3.5: whitening, length normalization and within-class covariance normalization.

Once session compensation has been performed, any classification technique might be used. We investigate the simple and efficient cosine similarity measure (cf. eq. (3.59)), as well as SVMs [Vapnik, 1995], leading to two systems **TV-Cosine** and **TV-SVM**, respectively.

B.3 Experimental Evaluation

In this section, we evaluate the accuracy of unimodal and bimodal gender recognition systems on several databases. For both visual and audio modalities, four gender recognition systems are employed, relying on the modeling and classification techniques described in sec. B.2. We call them **GMM**, **ISV**, **TV-SVM** and **TV-Cosine**, respectively.

The description of the systems is similar to the one given in tab. 5.9 and tab. 6.2. GMMs are composed of 512 Gaussian components with diagonal covariance matrices, and the ranks of the subspaces are respectively set to 50 for ISV (matrix $\boldsymbol{U}$) and 400 for TV (matrix $\boldsymbol{T}$), respectively. Given the small size of the training set of FERET, the TV subspace has a rank of 200 on this database. One difference is in the GMM training, where 10 iterations of k-means, followed by 25 iterations of maximum-likelihood are performed.

The results reported in this appendix cannot be reproduced using the satellite package associated with this dissertation. However, a separate satellite package based on the bob toolbox [Anjos et al., 2012] has been made available online. [2]

Similarly to [Mäkinen and Raisamo, 2008a,b, Dago-Casas et al., 2011], the evaluation metrics used in our work are the accuracy (Acc), the true positive rate (TPR) and the true negative rate (TNR) that are defined by:

$$\text{Acc} = \frac{\text{TP}+\text{TN}}{\text{P}+\text{N}}, \quad \text{TPR} = \frac{\text{TP}}{\text{P}}, \quad \text{TNR} = \frac{\text{TN}}{\text{N}}, \tag{B.5}$$

where TP is the number of samples correctly classified as positive (i.e., male), TN the number of samples correctly classified as negative (i.e., female), P the total number of positive samples and N the total number of negative samples. Furthermore, we used a variant of the receiver operating characteristic (ROC) curve that plots the fraction of males classified correctly in terms of the fraction of females classified incorrectly [Mäkinen and Raisamo, 2008b].

B.3.1 Face-based Gender Recognition

The problem of face-based gender recognition has been tackled in [Mäkinen and Raisamo, 2008a, Alexandre, 2010]. In their work, the experiments rely on a subset of the FERET database [Phillips et al., 2000], for which an evaluation protocol is already established. [3] For the sake of comparison, we conducted a set of experiments on this small corpus (411 images), using the same annotations and the same protocol. Another drawback of using this database is the well controlled recording conditions of the images.

In contrast to FERET, images of the LFW database [4] [Huang et al., 2007b] were acquired in an uncontrolled environment, leading to higher variability in term of pose, illumination

2. https://pypi.python.org/pypi/xbob.gender.bimodal
3. http://www.sis.uta.fi/~em55910/datasets/
4. http://vis-www.cs.umass.edu/lfw/

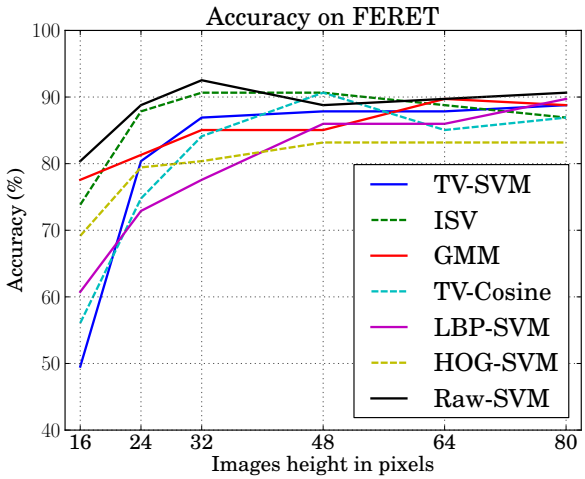


Figure B.1 – IMPACT OF IMAGE RESOLUTION ON GENDER RECOGNITION ON FERET. *This figure shows the accuracy of the systems on the FERET database on varying image resolution. The height and width are set to identical values after cropping.*

and expression. In addition, the number of samples is significantly larger (13,233 images). Experiments are conducted on this corpus using the BeFIT evaluation protocol (see sec. B.1.1).

We evaluated our proposed systems on both databases, using a very similar setup.

First, images are rotated, scaled and cropped to a fixed size, according to eye coordinate annotations and using a parametrization similar to [Mäkinen and Raisamo, 2008a].

For the four proposed systems (**GMM**, **ISV**, **TV-SVM** and **TV-Cosine**), we rely on parts-based features, as described in sec. 5.3.3. Compared to tab. 5.9, the differences are the resolution of the cropping and the non-application of the preprocessing step [Tan and Triggs, 2010]. We indeed observed a degradation in performance when applying this technique.

We also evaluate other baselines that apply SVM on raw pixels (**Raw-SVM**) or on LBP features (**LBP-SVM**), as proposed in [Mäkinen and Raisamo, 2008a], as well as SVM on histogram of oriented gradients (HOG) [Dalal and Triggs, 2005] (**HOG-SVM**).

On the FERET corpus, we first evaluate all the systems at different image resolutions. Fig. B.1 shows that the accuracy of the systems is stabilizing when image resolution is increasing. Therefore, we set the resolution of cropped images to the reasonable value of 80×80 in further experiments.

Additionally, tab. B.1 compares the accuracy of our systems to the results published in [Mäkinen and Raisamo, 2008a], using their image resolution and cropping. At the largest resolution of 48×48, results suggest that the proposed **TV-SVM**, **ISV** and **GMM** systems outperform the baselines. In particular, **ISV** reaches an accuracy of 90.7%, compared to 84.0% for the best system of [Mäkinen and Raisamo, 2008a] (**Raw-SVM**).

Table B.1 – GENDER RECOGNITION ACCURACY ON FERET. *This table reports the accuracy (%) of the systems on FERET.*

Resolution	TV-SVM	ISV	GMM	TV-Cosine	HOG-SVM	LBP-SVM	Neural Network	Raw-SVM	AdaBoost
						[Mäkinen and Raisamo, 2008a]			
24 × 24	80.4	87.9	81.3	74.8	72.9	76.9	84.2	82.6	81.5
48 × 48	86.9	90.7	85.0	84.1	77.6	82.1	82.9	84.0	83.9

Experiments conducted on LFW show similar trends, the **ISV** system providing a good accuracy, as reported in fig. B.2. Nevertheless, the **TV-SVM** system significantly outperforms other systems and achieves state-of-the-art performances on this corpus (accuracy of 94.6% as shown in tab. B.2), compared to the best previously published results [Dago-Casas et al., 2011]. Looking at the errors made by **TV-SVM** (examples are depicted in fig. B.3), results suggest that the high intra-class variability remains one of the main challenges. This variability is caused by recording conditions such as pose, illumination and expression on one hand, and accessories, hair styles and make-up on the other hand.

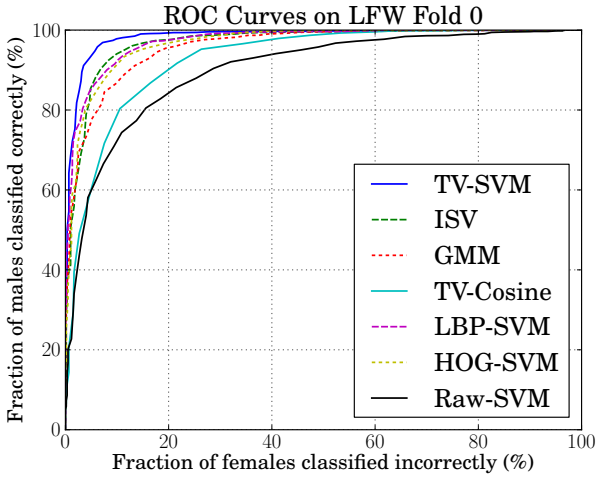


Figure B.2 – GENDER RECOGNITION ACCURACY ON THE FIRST FOLD OF LFW. *This figure reports the accuracy (%) of all the systems on the first fold of the LFW database.*

Table B.2 – GENDER RECOGNITION ACCURACY ON LFW. *This table reports the accuracy (%), the true positive rate (*TPR *in %, for males) and the true negative rate (*TNR *in %, for females) on LFW after 5-fold cross-validation. The image resolution employed by each system is given in brackets.*

System	Acc	TPR	TNR
TV-SVM (80 × 80)	94.6	97.4	85.0
Gabor-PCA-SVM (120 × 105) [Dago-Casas et al., 2011]	94.0	97.5	82.2
LBP-PCA-SVM (120 × 105) [Dago-Casas et al., 2011]	93.8	97.0	83.0
Raw-PCA-SVM (120 × 105) [Dago-Casas et al., 2011]	89.2	95.4	68.1

Figure B.3 – MISCLASSIFIED SAMPLES BY TV-SVM ON THE FIRST FOLD OF LFW. *This figure shows misclassified samples (top row: females; bottom row: males) by the proposed **TV-SVM** gender recognition system. These are original images aligned with funneling from the LFW database, fold* 0.

Table B.3 – NIST SRE PARTITIONING FOR GENDER RECOGNITION. *This table reports the number of male and female speakers and the number of utterances in the training, development and evaluation sets of the NIST-SRE protocol for gender recognition.*

	Training	**Development**	**Evaluation**
NIST SRE series	2006	2010	2012
Number of male speakers	481	235	763
Number of female speakers	659	261	1,155
Number of utterances	14,735	22,848	73,106

B.3.2 Speech-based Gender Recognition

We evaluate our gender recognition systems on audio data from the MIXER corpus [Cieri et al., 2004], which is provided by NIST since 2004 for the task of speaker recognition. The training set uses data from NIST SRE (Speaker Recognition Evaluation) 2006, while the development and the evaluation sets use data from NIST SRE 2010 and 2012, respectively. The recordings were collected in uncontrolled conditions (e.g., microphone, telephone, synthetic noise, real noise, duration variability, etc.). Statistics on the number of male and female speakers, and the number of utterances are reported in tab. B.3. To the best of our knowledge, this is the first large scale gender recognition experiment conducted on audio data.

Acoustic features are extracted at equally-spaced time instants using a sliding window approach. For all the proposed systems (**GMM**, **ISV**, **TV-SVM** and **TV-Cosine**), we rely on MFCCs features, as described in sec. 6.3. The only difference is in the voice activity detection, which is performed using jointly the normalized log energy and the 4 Hz modulation energy [Scheirer and Slaney, 1997] as in chapter 7. The resulting acoustic feature vectors are of dimensionality $D_{\boldsymbol{o}} = 60$.

Results in fig. B.4 clearly show that **TV-SVM** and **ISV** outperform the state-of-the-art **GMM** system by up to 11% of relative gain.

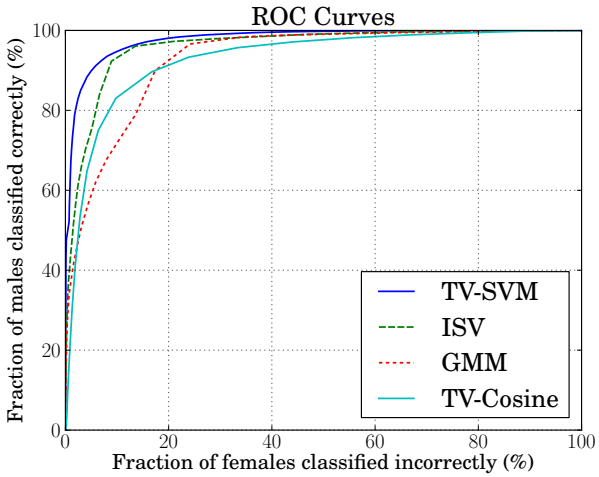


Figure B.4 – ACCURACY OF THE GENDER RECOGNITION SYSTEMS ON NIST SRE. *This figure shows the ROC curves of the different systems on the evaluation set of the NIST SRE dataset.*

B.3.3 Bimodal Gender Recognition

We evaluated bimodal gender recognition on the MOBIO database, which consists of 61 hours of audio-visual data of 150 people captured within twelve sessions (see sec. 7.3). This corpus is challenging since the data are acquired on mobile devices with real noise. It has been used to evaluate several speaker, face and bimodal recognition systems (see chapter 7 or [McCool et al., 2012]). The extracted images contain faces with uncontrolled illumination, facial expression, and occlusion, while the extracted speech segments are relatively short, partially even less than two seconds. A new protocol for gender recognition is established, with separate training, development and evaluation sets, each containing 50 identities.

For each modality, we employ the same features and parametrization as the ones introduced for the unimodal systems (cf. sec. B.3.1 and sec. B.3.2). The combination of the two modalities is performed using score fusion (cf. sec. 7.2.1). For this purpose, we use the linear logistic regression approach, which has been successfully employed for combining heterogeneous speaker classifiers (cf. sec. 7.2.2 and [Pigeon et al., 2000]).

Let an audio-visual test sample $\boldsymbol{\chi}_{\text{test}} = (\boldsymbol{\chi}^{\text{a}}_{\text{test}}, \boldsymbol{\chi}^{\text{v}}_{\text{test}})$ be processed by both audio and visual systems. Each system produces an output score, $h_s^{\text{audio}}(\boldsymbol{\chi}^{\text{a}}_{\text{test}})$ and $h_s^{\text{visual}}(\boldsymbol{\chi}^{\text{v}}_{\text{test}})$ for audio and visual cues, respectively. The final fused score is expressed by the logistic function:

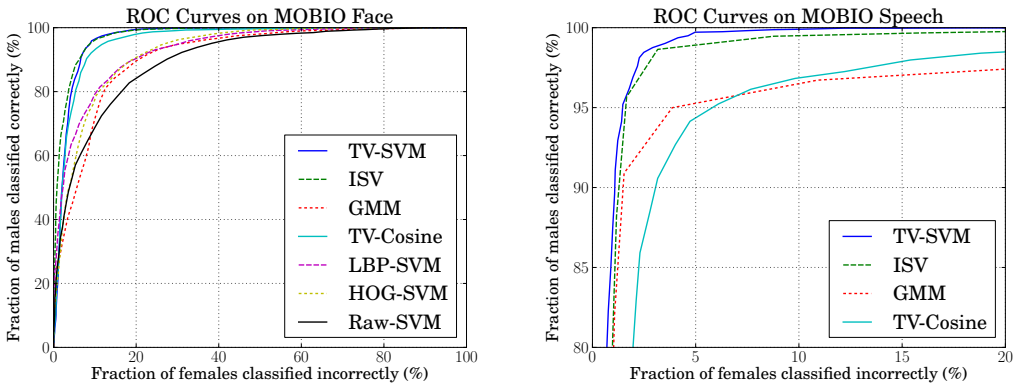
$$h_s^{\text{fusion}}(\boldsymbol{\chi}_{\text{test}}) = g\left(\beta_0 + \beta_1 h_s^{\text{audio}}(\boldsymbol{\chi}^{\text{a}}_{\text{test}}) + \beta_2 h_s^{\text{visual}}(\boldsymbol{\chi}^{\text{v}}_{\text{test}})\right), \tag{B.6}$$

where:

$$g(x) = \frac{1}{1+\exp(-x)}, \tag{B.7}$$

and $\boldsymbol{\beta} = [\beta_0, \beta_1, \beta_2]$ being the regression coefficients that are computed by estimating the maximum likelihood of the logistic regression model on the scores of the development set.

Performances of unimodal gender recognition systems on the MOBIO database are shown in fig. B.5. For the visual modality, **TV-SVM**, **ISV** and **TV-Cosine** outperform the **Raw-SVM** and **LBP-SVM** baselines. For the audio modality, **TV-SVM**, **ISV** and **GMM** achieve very high performances.



(a) Visual (b) Audio

Figure B.5 – ACCURACY OF THE BIMODAL GENDER RECOGNITION SYSTEMS ON MOBIO. *This figure shows the ROC curves for both visual and audio modalities on the evaluation set of MOBIO. For the audio modality, a zoom is performed in the region of interest, as a high accuracy is achieved.*

Interestingly, when comparing the two modalities (tab. B.4), a significantly higher accuracy (96.8%) is achieved with the audio modality, compared to the visual one (92.2%). In addition, for speech-based gender recognition, the classification rates are comparable for the two classes male and female. In contrast, for face-based gender recognition, there is a large gap between male (TPR) and female (TNR) classification rates.

Table B.4 – GENDER RECOGNITION ACCURACY ON MOBIO. *This table reports the accuracy (%), the true positive rate (*TPR *in %, for males) and the true negative rate (*TNR *in %, for females) of the systems on the evaluation set of MOBIO.*

		TV-SVM	ISV	GMM	TV-Cosine
Face	**Acc**	91.9	92.2	83.6	88.8
	TPR	94.0	94.6	88.6	88.1
	TNR	87.8	87.5	73.9	90.1
Speech	**Acc**	96.8	96.2	95.2	93.2
	TPR	96.0	95.1	93.6	91.0
	TNR	98.4	98.4	98.4	97.4

We investigate the fusion of several unimodal systems. Results depicted in fig. B.6 show that the fusion of the two modalities allows to drastically reduce the error rate, reaching an accuracy of about 98% for the **TV-SVM** and **ISV** systems.

In addition, real classification examples of the **TV-SVM** systems (both unimodal and bimodal

ones) are illustrated in fig. B.7. Sample 1 (first column) is classified correctly by all the unimodal and bimodal systems. In contrast, samples 2 to 5 are only classified correctly by one of the two unimodal systems, but the bimodal fusion is still able to take the right decision. This suggests that, when a modality is affected by challenging conditions (e.g., noise or accessories), the other modality is available to come to the rescue. Sample 6 is a very challenging case, where both modalities are subject to high deformation.

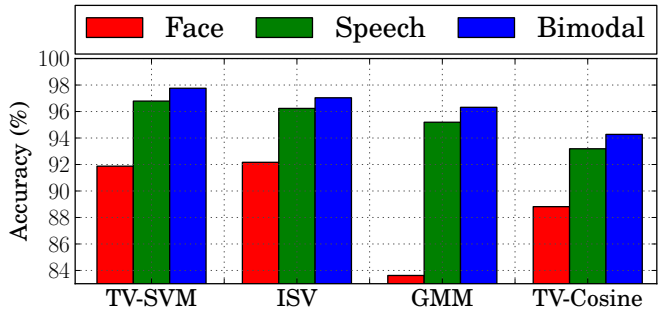


Figure B.6 – PERFORMANCE OF THE GENDER RECOGNITION SYSTEMS ON MOBIO. *This figure reports the accuracy (%) of several unimodal and bimodal systems on the evaluation set of MOBIO.*

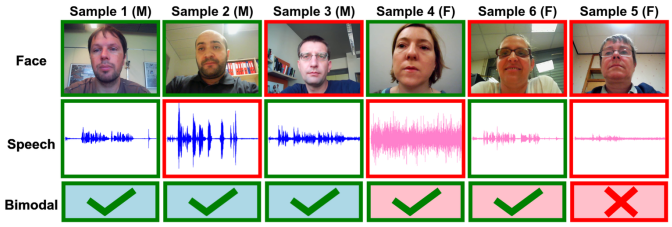


Figure B.7 – CLASSIFICATION EXAMPLES ON MOBIO. *This figure shows few classification examples of the bimodal **TV-SVM** system. Each column corresponds to a test sample, while each row corresponds to a modality (face, speech and bimodal, respectively). A green box around a cell indicates that the sample has been classified correctly, while red indicates misclassification.*

B.4 Conclusions

This appendix investigates the problem of audio, visual and bimodal gender recognition with two different variability modeling techniques: ISV and TV. For visual gender recognition, state-of-the-art performances are achieved on both FERET and LFW databases. For the audio modality, the large-scale evaluation conducted on NIST SRE shows that the **TV-SVM** system is achieving an accuracy of 92.5%. In addition, experiments were carried out on the bimodal MOBIO database. Results show that our proposed **TV-SVM** and **ISV** systems outperform state-of-the-algorithms on both modalities. Furthermore, additional improvements are obtained by combining them using score fusion based on linear logistic regression. The final accuracy of the bimodal system is around 98%.

Bibliography

T. Abeel, Y. V. de Peer, and Y. Saeys. Java-ML: A machine learning library. *Journal of Machine Learning Research*, 10:931–934, 2009.

A. G. Adami, L. Burget, S. Dupont, H. Garudadri, F. Grézl, H. Hermansky, P. Jain, S. S. Kajarekar, N. Morgan, and S. Sivadas. QUALCOMM-ICSI-OGI features for ASR. In *International Conference on Spoken Language Processing (ICSLP / INTERSPEECH)*, pages 4–7, September 2002.

Y. Adini, Y. Moses, and S. Ullman. Face recognition: The problem of compensating for changes in illumination direction. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 19(7):721–732, 1997. ISSN 0162-8828. doi: 10.1109/34.598229.

T. Ahonen, A. Hadid, and M. Pietikäinen. Face recognition with local binary patterns. In *European Conference on Computer Vision (ECCV)*, volume 3021, pages 469–481, 2004. ISBN 978-3-540-21984-2. doi: 10.1007/978-3-540-24670-1_36.

T. Ahonen, A. Hadid, and M. Pietikainen. Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 28(12):2037–2041, 2006. ISSN 0162-8828. doi: 10.1109/TPAMI.2006.244.

L. A. Alexandre. Gender recognition: A multiscale decision fusion approach. *Pattern Recognition Letters*, 31(11):1422–1427, 2010. ISSN 0167-8655. doi: http://dx.doi.org/10.1016/j.patrec.2010.02.010.

E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. D. Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition, 1999.

A. Anjos, L. El Shafey, R. Wallace, M. Günther, C. McCool, and S. Marcel. Bob: a free signal processing and machine learning toolbox for researchers. In *Proceedings of the ACM Multimedia Conference (ACMMM)*, pages 1449–1452, October 2012. ISBN 978-1-4503-1089-5. doi: 10.1145/2393347.2396517. URL http://www.idiap.ch/software/bob/.

O. Arandjelović, G. Shakhnarovich, J. Fisher, R. Cipolla, and T. Darrell. Face recognition with image sets using manifold density divergence. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 581–588, 2005. doi: 10.1109/CVPR.2005.151.

D. Arthur and S. Vassilvitskii. k-means++: the advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA'07, pages 1027–1035, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics. ISBN 978-0-898716-24-5.

B. S. Atal. Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification. *The Journal of the Acoustical Society of America*, 55 (6):1304–1312, 1974. doi: 10.1121/1.1914702.

C. Atanasoaei. *Multivariate Boosting with Look-up Tables for Face Processing*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne (EPFL), 2012.

R. Auckenthaler, M. Carey, and H. Lloyd-Thomas. Score normalization for text-independent speaker verification systems. *Digital Signal Processing*, 10(1-3):42–54, 2000. ISSN 1051-2004. doi: 10.1006/dspr.1999.0360.

E. Bailly-Baillière, S. Bengio, F. Bimbot, M. Hamouz, J. Kittler, J. Mariéthoz, J. Matas, K. Messer, V. Popovici, F. Porée, B. Ruiz, and J.-P. Thiran. The BANCA database and evaluation protocol. In *International Conference on Audio- and Video-Based Biometric Person Authentication (AVBPA)*, 2003.

B. Balas, D. Cox, and E. Conwell. The effect of personal familiarity on the speed of face recognition. In *Annual Conference of the Cognitive Science Society*, pages 36–41, 2006.

D. J. Bartholomew, M. Knott, and I. Moustaki. *Latent Variable Models and Factor Analysis: A Unified Approach*, volume 899. Wiley, 2011.

A. T. Basilevsky. *Statistical Factor Analysis and Related Methods: Theory and Applications*, volume 418. John Wiley & Sons, 2009.

G. Baudat and F. Anouar. Generalized discriminant analysis using a kernel approach. *Neural computation*, 12(10):2385–2404, 2000.

H. Beigi. *Fundamentals of Speaker Recognition*. Springer, 2011.

P. N. Belhumeur, P. Hespanha, and D. J. Kriegman. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 19(7):711–720, July 1997. ISSN 0162-8828. doi: 10.1109/34.598228.

S. Ben-Yacoub, Y. Abdeljaoued, and E. Mayoraz. Fusion of face and speech data for person identity verification. *IEEE Transactions on Neural Networks (TNN)*, 10(5):1065–1074, 1999. ISSN 1045-9227. doi: 10.1109/72.788647.

S. Bengio. Multimodal authentication using asynchronous HMMs. In *International Conference on Audio- and Video-Based Biometric Person Authentication (AVBPA)*, volume 2688, pages 770–777. Springer-Verlag, 2003. ISBN 978-3-540-40302-9. doi: 10.1007/3-540-44887-X_89.

A. Benyassine, E. Shlomot, H. yu Su, D. Massaloux, C. Lamblin, and J.-P. Petit. ITU-T Recommendation G.729 Annex B: a silence compression scheme for use with G.729 optimized for V.70 digital simultaneous voice and data applications. *IEEE Communications Magazine*, 35 (9):64–73, 1997. ISSN 0163-6804. doi: 10.1109/35.620527.

E. Bigün, J. Bigün, B. Duc, and S. Fischer. Expert conciliation for multi modal person authentication systems by Bayesian statistics. In *International Conference on Audio- and Video-based Biometric Person Authentication (AVBPA)*, volume 1206, pages 291–300. Springer Berlin Heidelberg, 1997. ISBN 978-3-540-62660-2. doi: 10.1007/BFb0016008.

C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 2007. ISBN 0387310738.

W. W. Bledsoe. Man-machine facial recognition: Report on a large-scale experiment. Technical report, Panoramic Research, Inc., 1966.

S. F. Boll. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on Acoustics, Speech and Signal Processing (TASSP)*, 27(2):113–120, 1979. ISSN 0096-3518. doi: 10.1109/TASSP.1979.1163209.

R. H. Bolt, F. S. Cooper, E. E. D. Jr, P. B. Denes, J. M. Picket, and K. N. Stevens. Speaker identification by speech spectrograms: A scientists' view of its reliability for legal purposes. *The Journal of the Acoustical Society of America*, 47:597, 1970.

A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Three-dimensional face recognition. *International Journal of Computer Vision (IJCV)*, 64(1):5–30, 2005.

N. Brümmer. LLR transformation for SRE'12, 2012.

N. Brummer, L. Burget, J. Cernocky, O. Glembek, F. Grezl, M. Karafiat, D. van Leeuwen, P. Matejka, P. Schwarz, and A. Strasheim. Fusion of heterogeneous speaker recognition systems in the STBU submission for the NIST speaker recognition evaluation 2006. *IEEE Transactions on Speech, Audio and Language Processing (TSALP)*, 15(7):2072–2084, September 2007. ISSN 1558-7916. doi: 10.1109/TASL.2007.902870.

N. Brümmer, L. Burget, P. Kenny, P. Matějka, E. V. de, M. Karafiát, M. Kockmann, O. Glembek, O. Plchot, D. Baum, and M. Senoussauoi. ABC system description for NIST SRE 2010. In *NIST Speaker Recognition Workshop*, pages 1–20, 2010.

R. Brunelli and D. G. Falavigna. Person identification using multiple cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 17(10):955–966, 1995. ISSN 0162-8828. doi: 10.1109/34.464560.

R. Brunelli, D. G. Falavigna, T. Poggio, and L. Stringa. Automatic person recognition by acoustic and geometric features. *Machine Vision and Applications*, 8(5):317–325, 1995. ISSN 0932-8092. doi: 10.1007/BF01211493.

L. Burget, M. Fapšo, V. Hubeika, O. Glembek, M. Karafiát, M. Kockmann, P. Matějka, P. Schwarz, and J. Černocký. BUT system description: NIST SRE 2008. In *NIST Speaker Recognition Evaluation Workshop*, pages 1–4, 2008.

L. Burget, O. Plchot, S. Cumani, O. Glembek, P. Matějka, and N. Brümmer. Discriminatively trained probabilistic linear discriminant analysis for speaker verification. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 4832–4835, 2011. doi: 10.1109/ICASSP.2011.5947437.

F. Burkhardt, M. Eckert, W. Johannsen, and J. Stegmann. A database of age and gender annotated telephone speech. In *International Conference on Language Resources and Evaluation (LREC'10)*, 2010.

R. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific and Statistical Computing*, 16(5):1190–1208, 1995.

J. P. Campbell. Speaker recognition: a tutorial. *Proceedings of the IEEE*, 85(9):1437–1462, 1997. ISSN 0018-9219. doi: 10.1109/5.628714.

W. M. Campbell, J. P. Campbell, D. A. Reynolds, D. A. Jones, and T. R. Leek. Phonetic speaker recognition with support vector machines. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1377–1384. MIT Press, 2004.

W. M. Campbell, J. P. Campbell, D. A. Reynolds, E. Singer, and P. Torres-Carrasquillo. Support vector machines for speaker and language recognition. *Computer Speech & Language*, 20 (2-3):210–229, 2006a. ISSN 0885-2308. doi: 10.1016/j.csl.2005.06.003.

W. M. Campbell, D. E. Sturim, and D. A. Reynolds. Support vector machines using GMM supervectors for speaker verification. *IEEE Signal Processing Letters*, 13(5):308–311, 2006b. ISSN 1070-9908. doi: 10.1109/LSP.2006.870086.

F. Cardinaux, C. Sanderson, and S. Marcel. Comparison of MLP and GMM classifiers for face verification on XM2VTS. In *International Conference Audio- and Video-Based Biometric Person Authentication (AVBPA)*, pages 911–920, 2003. doi: 10.1007/3-540-44887-X_106.

F. Cardinaux, C. Sanderson, and S. Bengio. Face verification using adapted generative models. In *IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, pages 825–830, 2004. doi: 10.1109/AFGR.2004.1301636.

F. Cardinaux, C. Sanderson, and S. Bengio. User authentication via adapted statistical models of face images. *IEEE Transactions on Signal Processing (TSP)*, 54(1):361–373, 2006. ISSN 1053-587X. doi: 10.1109/TSP.2005.861075.

C. Champod and D. Meuwly. The inference of identity in forensic speaker recognition. *Speech Communication*, 31(2–3):193 – 203, 2000. ISSN 0167-6393. doi: 10.1016/S0167-6393(99)00078-3.

C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions Intelligent System Technologies*, 2(3):27:1–27:27, 2011.

T. Chen, W. Yin, X. S. Zhou, D. Comaniciu, and T. S. Huang. Total variation models for variable lighting face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 28(9):1519–1524, 2006. ISSN 0162-8828. doi: 10.1109/TPAMI.2006.195.

S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *IEEE International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 539–546, 2005. doi: 10.1109/CVPR.2005.202.

C. Cieri, J.-P. Campbell, H. Nakasone, D. Miller, and K. Walker. The mixer corpus of multilingual, multichannel speaker recognition data. In *LREC*. European Language Resources Association, 2004.

R. Collobert, S. Bengio, and J. Mariéthoz. Torch: a modular machine learning software library. Technical report, 2002. URL http://www.torch.ch/torch3/.

R. Collobert, K. Kavukcuoglu, and C. Farabet. Torch7: a matlab-like environment for machine learning. *NIPS Workshop BigLearn*, 2011. URL http://www.torch.ch/.

T. Cootes, C. Taylor, D. Cooper, and J. Graham. Active shape models - their training and application. *Computer Vision and Image Understanding (CVIU)*, 61(1):38–59, 1995. ISSN 1077-3142. doi: 10.1006/cviu.1995.1004.

T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active Appearance Models. *IEEE Transactions Pattern on Pattern Analysis and Machine Intelligence (TPAMI)*, 23(6):681–685, June 2001. ISSN 0162-8828. doi: 10.1109/34.927467.

P. Dago-Casas, D. Gonzalez-Jimenez, L. Yu, and J. Alba-Castro. Single- and cross- database benchmarks for gender classification under unconstrained settings. In *IEEE International Conference on Computer Vision Workshops (CVPRW)*, 2011.

N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 886–893, June 2005.

B. V. Dasarathy. *Decision fusion*. IEEE Computer Society Press, 1994. ISBN 978-0818644528.

S. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech and Signal Processing (TASSP)*, 28(4):357–366, 1980. ISSN 0096-3518. doi: 10.1109/TASSP.1980.1163420.

N. Dehak. *Discriminative and generative approaches for long- and short-term speaker characteristics modeling: application to speaker verification*. PhD thesis, Ecole de Technologie Supérieure (Canada), 2009.

N. Dehak and G. Chollet. Support vector GMMs for speaker verification. In *IEEE Odissey: the Speaker and Language Recognition Workshop*, pages 1–4, 2006. doi: 10.1109/ODYSSEY.2006.248131.

N. Dehak, R. Dehak, P. Kenny, N. Brümmer, P. Ouellet, and P. Dumouchel. Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification. In *INTERSPEECH*, pages 1559–1562, 2009.

N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing (TASLP)*, 19(4):788–798, 2011. ISSN 1558-7916.

A. DeMarco and S. Cox. An accurate and robust gender identification algorithm. In *INTERSPEECH*, pages 2429–2432, 2011.

R. Demming and D. Duffy. *Introduction to the Boost C++ Libraries; Volume I - Foundations*. Number v. 1. Datasim Education Bv, 2010.

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 (Methodological)*, 39(1):1–38, 1977.

U. Dieckmann, P. Plankensteiner, and T. Wagner. Sesam: A biometric person identification system using sensor fusion. *Pattern Recognition Letters*, 18(9):827–833, 1997. ISSN 0167-8655. doi: 10.1016/S0167-8655(97)00063-9.

G. R. Doddington. Speaker recognition based on idiolectal differences between speakers. In *INTERSPEECH*, pages 2521–2524, 2001.

R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2000. ISBN 0471056693.

H. K. Ekenel, M. Fischer, Q. Jin, and R. Stiefelhagen. Multi-modal person identification in a smart environment. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007. doi: 10.1109/CVPR.2007.383388.

L. El Shafey, A. Anjos, M. Günther, E. Khoury, I. Chingovska, F. Moulin, and S. Marcel. Bob: A free library for reproducible machine learning. Idiap-RR Idiap-Internal-RR-59-2013, Idiap, July 2013a. URL https://pypi.python.org/pypi/xbob.paper.example. submitted.

L. El Shafey, E. Khoury, and S. Marcel. Audio-visual gender recognition in uncontrolled environment using variability modeling techniques. Idiap-RR Idiap-Internal-RR-87-2013, Idiap, 2013b. URL https://pypi.python.org/pypi/xbob.gender.bimodal. submitted.

L. El Shafey, C. McCool, R. Wallace, and S. Marcel. A scalable formulation of probabilistic linear discriminant analysis: Applied to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(7):1788–1794, July 2013c. URL https://pypi.python.org/pypi/xbob.paper.tpami2013.

Y. Ephraim and D. Malah. Speech enhancement using a minimum mean-square error short-time spectral amplitude estimator. *IEEE Transactions on Acoustics, Speech and Signal Processing (TASSP)*, 32(6):1109–1121, 1984. ISSN 0096-3518. doi: 10.1109/TASSP.1984.1164453.

M. Everingham, J. Sivic, and A. Zisserman. "Hello! My name is... Buffy" – automatic naming of characters in TV video. In *British Machine Vision Conference*, 2006.

G. Fant. *Acoustic Theory of Speech Production*. Mouton, 1960.

K. R. Farrell, R. J. Mammone, and K. T. Assaleh. Speaker recognition using neural networks and conventional classifiers. *IEEE Transactions on Speech and Audio Processing (TSAP)*, 2 (1):194–205, 1994. ISSN 1063-6676. doi: 10.1109/89.260362.

R. A. Fisher. On the Mathematical Foundations of Theoretical Statistics. *Philosophical Transactions of the Royal Society of London, A*, 222:309–368, 1922.

R. A. Fisher. The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics*, 7 (2):179–188, 1936. ISSN 2050-1439. doi: 10.1111/j.1469-1809.1936.tb02137.x.

Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, August 1997. ISSN 0022-0000. doi: 10.1006/jcss.1997.1504.

B. Fröba and A. Ernst. Face detection with the modified census transform. In *IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, pages 91–96, 2004. doi: 10.1109/AFGR.2004.1301514.

S. Furui. Cepstral analysis technique for automatic speaker verification. *IEEE Transactions on Acoustics, Speech and Signal Processing (TSAP)*, 29(2):254–272, 1981. ISSN 0096-3518. doi: 10.1109/TASSP.1981.1163530.

A. Gallagher and T. Chen. Clothing cosegmentation for recognizing people. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

A. Gallagher and T. Chen. Understanding images of groups of people. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 256–263, 2009.

W. Gao, B. Cao, S. Shan, D. Zhou, X. Zhang, and D. Zhao. The CAS-PEAL large-scale Chinese face database and evaluation protocols. Technical report, Joint Research & Development Laboratory, CAS, 2004.

W. Gao, B. Cao, S. Shan, X. Chen, D. Zhou, X. Zhang, and D. Zhao. The CAS-PEAL large-scale Chinese face database and baseline evaluations. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 38(1):149–161, 2008. ISSN 1083-4427. doi: 10.1109/TSMCA.2007.909557.

D. Garcia-Romero and C. Y. Espy-Wilson. Analysis of i-vector length normalization in speaker recognition systems. In *INTERSPEECH*, pages 249–252, 2011.

J. Gauvain and C.-H. Lee. Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *IEEE Transactions on Speech and Audio Processing (TSAP)*, 2 (2):291–298, 1994. ISSN 1063-6676. doi: 10.1109/89.279278.

O. Glembek. *Optimization of Gaussian Mixture Subspace Models and related scoring algorithms in speaker verification*. PhD thesis, Brno University of Technology, 2012.

O. Glembek, L. Burget, N. Dehak, N. Brümmer, and P. Kenny. Comparison of scoring methods used in speaker recognition with joint factor analysis. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4057–4060, 2009. doi: 10.1109/ICASSP.2009.4960519.

C. S. Greenberg, V. M. Stanford, A. F. Martin, M. Yadagiri, G. R. Doddington, J. J. Godfrey, and J. Hernandez-Cordero. The 2012 NIST speaker recognition evaluation. In *INTERSPEECH*, pages 1971–1975, 2013.

R. Gross and V. Brajovic. An image preprocessing algorithm for illumination invariant face recognition. In J. Kittler and M. S. Nixon, editors, *Internation Conference on Audio- and Video-Based Biometric Person Authentication (AVBPA)*, volume 2688 of *Lecture Notes in Computer Science*, pages 10–18. Springer Berlin Heidelberg, June 2003. ISBN 978-3-540-40302-9. doi: 10.1007/3-540-44887-X_2.

R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-PIE. In *IEEE International Conference on Automatic Face Gesture Recognition (FG)*, pages 1–8, 2008. doi: 10.1109/AFGR.2008.4813399.

M. Guillaumin, J. Verbeek, and C. Schmid. Is that you? metric learning approaches for face identification. In *IEEE International Conference on Computer Vision (ICCV)*, pages 498–505, 2009. doi: 10.1109/ICCV.2009.5459197.

M. Günther, D. Haufe, and R. P. Würtz. Face recognition with disparity corrected Gabor phase differences. In *International Conference on Artificial Neural Networks and Machine Learning (ICANN)*, volume 7552, pages 411–418. Springer Berlin Heidelberg, 2012a. ISBN 978-3-642-33268-5. doi: 10.1007/978-3-642-33269-2_52.

M. Günther, R. Wallace, and S. Marcel. An open source framework for standardized comparisons of face recognition algorithms. In *European Conference on Computer Vision - Workshops and Demonstrations (ECCV WD)*, volume 7585 of *Lecture Notes in Computer Science*, pages 547–556, 2012b. ISBN 978-3-642-33884-7. doi: 10.1007/978-3-642-33885-4_55. URL https://pypi.python.org/pypi/facereclib.

M. Günther, L. El Shafey, and S. Marcel. 2D Face Recognition: An Experimental and Reproducible Research Survey. Idiap-RR Idiap-Internal-RR-01-2103, June 2013. submitted.

J. Hamm and D. D. Lee. Grassmann discriminant analysis: a unifying view on subspace-based learning. In *International Conference on Machine Learning*, pages 376–383, 2008.

H. Harb and L. Chen. Gender identification using a general audio classifier. In *International Conference on Multimedia and Expo (ICME)*, volume 2, pages 733–736, 2003.

A. O. Hatch, S. S. Kajarekar, and A. Stolcke. Within-class covariance normalization for SVM-based speaker recognition. In *INTERSPEECH*, 2006.

B. Heisele, P. Ho, and T. Poggio. Face recognition with support vector machines: Global versus component-based approach. In *IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 688–694, 2001. doi: 10.1109/ICCV.2001.937693.

H. Hermansky. Perceptual linear predictive (PLP) analysis of speech. *Journal of the Acoustical Society of America*, 87(4):1738–1752, April 1990.

G. Heusch and S. Marcel. Face authentication with salient local features and static Bayesian network. In *IEEE International Conference on Biometrics (ICB)*, pages 878–887, 2007.

Y. Hu, D. Wu, and A. Nucci. Pitch-based gender identification with two-stage classification. *Security and Communication Networks*, 5(2):211–225, 2012.

G. B. Huang, V. Jain, and E. Learned-Miller. Unsupervised joint alignment of complex images. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1–8, 2007a. doi: 10.1109/ICCV.2007.4408858.

G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled Faces in the Wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007b.

X. Huang, A. Acero, and H.-W. Hon. *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. Prentice Hall PTR, 1st edition, 2001. ISBN 0130226165.

C. Igel, T. Glasmachers, and V. Heidrich-Meisner. Shark. *Journal of Machine Learning Research*, 9:993–996, 2008.

ITU. ISO/IEC 10918-1| ITU-T Recommendation T.81, 1993. URL http://www.w3.org/Graphics/JPEG/itu-t81.pdf.

A. K. Jain, L. Hong, and Y. Kulkarni. A multimodal biometric system using fingerprint, face and speech. In *International Conference on Audio-and Video-based Biometric Person Authentication (AVBPA)*, pages 182–187, 1999a.

A. K. Jain, S. Prabhakar, and S. Chen. Combining multiple matchers for a high security fingerprint verification system. *Pattern Recognition Letters*, 20(11):1371–1379, 1999b.

A. K. Jain, P. Flynn, and A. A. Ross. *Handbook of Biometrics*. Springer-Verlag New York, Inc., 2007. ISBN 038771040X.

D. J. Jobson, Z. ur Rahman, and G. A. Woodell. A multiscale retinex for bridging the gap between color images and the human observation of scenes. *IEEE Transactions on Image Processing (TIP)*, 6(7):965–976, 1997a. ISSN 1057-7149. doi: 10.1109/83.597272.

D. J. Jobson, Z. ur Rahman, and G. A. Woodell. Properties and performance of a center/surround retinex. *IEEE Transactions on Image Processing (TIP)*, 6(3):451–462, 1997b. ISSN 1057-7149. doi: 10.1109/83.557356.

T. Kanade. *Picture Processing System by Computer Complex and Recognition of Human Faces*. PhD thesis, Kyoto University, November 1973.

T. Kanade. Computer recognition of human faces. *Interdisciplinary Systems Research*, 47, 1977.

A. Kapoor, S. Baker, S. Basu, and E. Horvitz. Memory constrained face recognition. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2539–2546, 2012. doi: 10.1109/CVPR.2012.6247971.

Z. N. Karam and W. M. Campbell. A new kernel for SVM MLLR based speaker recognition. In *INTERSPEECH*, pages 290–293, 2007.

P. Kenny. Bayesian speaker verification with heavy-tailed priors. In *Odyssey*, June 2010.

P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel. Joint factor analysis versus eigenchannels in speaker recognition. *IEEE Transactions on Audio, Speech, and Language Processing (TASLP)*, 15(4):1435–1447, 2007. ISSN 1558-7916. doi: 10.1109/TASL.2006.881693.

P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel. A study of inter-speaker variability in speaker verification. *IEEE Transactions on Audio, Speech and Language Processing (TASLP)*, 16(5):980–988, 2008. ISSN 1558-7916. doi: 10.1109/TASL.2008.925147.

E. Khoury, L. El Shafey, and S. Marcel. The Idiap Speaker Recognition Evaluation System at NIST SRE 2012. In *NIST Speaker Recognition Conference*. NIST, December 2012. URL https://pypi.python.org/pypi/xbob.spkrec.nist_sre12.

E. Khoury, L. El Shafey, C. McCool, M. Günther, and S. Marcel. Bi-modal biometric authentication on mobile phones in challenging conditions. *Image and Vision Computing*, October 2013a.

E. Khoury, M. Günther, L. El Shafey, and S. Marcel. On the improvements of uni-modal and bi-modal fusions of speaker and face recognition for mobile biometrics. In *Biometric Technologies in Forensic Science*, October 2013b.

E. Khoury, L. El Shafey, and S. Marcel. Spear: An open source toolbox for speaker recognition based on Bob. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014. URL https://pypi.python.org/pypi/bob.spear.

T.-K. Kim, J. Kittler, and R. Cipolla. On-line learning of mutually orthogonal subspaces for face recognition by image sets. *IEEE Transactions on Image Processing (TIP)*, 19(4):1067–1074, 2010. ISSN 1057-7149. doi: 10.1109/TIP.2009.2038621.

T. Kinnunen and H. Li. An overview of text-independent speaker recognition: From features to supervectors. *Speech Communication*, 52(1):12–40, 2010. ISSN 0167-6393. doi: 10.1016/j.specom.2009.08.009.

M. Kirby and L. Sirovich. Application of the Karhunen-Loeve procedure for the characterization of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 12(1):103–108, 1990. ISSN 0162-8828. doi: 10.1109/34.41390.

J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 20(3):226–239, 1998. ISSN 0162-8828. doi: 10.1109/34.667881.

J. C. Klontz and A. K. Jain. A case study on unconstrained facial recognition using the Boston marathon bombings suspects. Technical Report MSU-CSE-13-4, Department of Computer Science, Michigan State University, May 2013.

M. Kockmann, L. Burget, and J. Černocký. Brno university of technology system for interspeech 2010 paralinguistic challenge. In *INTERSPEECH*, volume 2010, pages 2822–2825, 2010.

C. W. Lau, B. Ma, H. M. Meng, Y. S. Moon, and Y. Yam. Fuzzy logic decision fusion in a multi-modal biometric system. In *International Conference on Spoken Language Processing (ICSLP)*, 2004.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998.

K.-A. Lee, C. You, H. Li, T. Kinnunen, and D. Zhu. Characterizing speech utterances for speaker verification with sequence kernel svm. In *INTERSPEECH*, pages 1397–1400, 2008.

K.-C. Lee, J. Ho, M.-H. Yang, and D. J. Kriegman. Video-based face recognition using probabilistic appearance manifolds. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 313–320, 2003. doi: 10.1109/CVPR.2003.1211369.

Z. Lei, M. Pietikainen, and S. Z. Li. Learning discriminant face descriptor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 99(PrePrints):1, 2013. ISSN 0162-8828. doi: http://doi.ieeecomputersociety.org/10.1109/TPAMI.2013.112.

M. Li, K. J. Han, and S. Narayanan. Automatic speaker age and gender recognition using acoustic and prosodic level information fusion. *Computer Speech Language*, 27(1):151–167, 2013.

P. Li and S. J. D. Prince. *Advance in Face Image Analysis: Techniques and Technologies*, chapter Probabilistic Methods for Face Registration and Recognition. Idea Group Publishing, 2010.

P. Li, Y. Fu, U. Mohammed, J. H. Elder, and S. J. D. Prince. Probabilistic models for inference about identity. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 34(1):144–157, January 2012. ISSN 0162-8828. doi: 10.1109/TPAMI.2011.104.

S. Z. Li and A. K. Jain. *Handbook of Face Recognition*. Springer-Verlag New York, Inc., 2005. ISBN 038740595X.

S.-H. Lin, S.-Y. Kung, and L.-J. Lin. Face recognition/detection by probabilistic decision-based neural network. *IEEE Transactions on Neural Networks (TNN)*, 8(1):114–132, 1997. ISSN 1045-9227. doi: 10.1109/72.554196.

M. Liu, X. Xu, and T. S. Huang. Audio-visual gender recognition. In *International Symposium on Multispectral Image Processing and Pattern Recognition*, 2007.

D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal Computer Vision*, 60(2):91–110, November 2004. ISSN 0920-5691. doi: 10.1023/B:VISI.0000029664.99615.94.

S. Lucey and T. Chen. A GMM parts based face representation for improved verification through relevance adaptation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages II–855–II–861 Vol.2, 2004. doi: 10.1109/CVPR.2004.1315254.

Y. M. Lui, D. S. Bolme, P. J. Phillips, J. R. Beveridge, and B. A. Draper. Preliminary studies on the Good, the Bad, and the Ugly face recognition challenge problem. In *IEEE International Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 9–16, 2012. doi: 10.1109/CVPRW.2012.6239209.

J. B. MacQueen. Some Methods for classification and analysis of multivariate observations. In *Procedings of the Fifth Berkeley Symposium on Math, Statistics, and Probability*, volume 1, pages 281–297. University of California Press, 1967.

E. Mäkinen and R. Raisamo. Evaluation of gender classification methods with automatically detected and aligned faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 30(3):541–547, 2008a.

E. Mäkinen and R. Raisamo. An experimental comparison of gender classification methods. *Pattern Recognition Letters*, 29(10):1544–1556, July 2008b.

R. J. Mammone, X. Zhang, and R. P. Ramachandran. Robust speaker recognition: A feature-based approach. *IEEE Signal Processing Magazine*, 13(5):58–71, 1996. ISSN 1053-5888. doi: 10.1109/79.536825.

S. Marcel, C. McCool, P. Matějka, T. Ahonen, J. Černocký, S. Chakraborty, V. Balasubramanian, S. Panchanathan, C. H. Chan, J. Kittler, N. Poh, B. Fauve, O. Glembek, O. Plchot, Z. Jančík, A. Larcher, C. Lévy, D. Matrouf, J.-F. Bonastre, P.-H. Lee, J.-Y. Hung, S.-W. Wu, Y.-P. Hung, L. Machlica, J. Mason, S. Mau, C. Sanderson, D. Monzo, A. Albiol, H. V. Nguyen, L. Bai,

Y. Wang, M. Niskanen, M. Turtinen, J. A. Nolazco-Flores, L. P. Garcia-Perera, R. Aceves-Lopez, M. Villegas, and R. Paredes. On the results of the first mobile biometry (MOBIO) face and speaker verification evaluation. In *International Conference on Pattern Recognition (ICPR)*, volume 6388, pages 210–225, 2010. ISBN 978-3-642-17710-1. doi: 10.1007/978-3-642-17711-8_22.

J. Markel, B. Oshika, and J. Gray, A. Long-term feature averaging for speaker recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing (TASSP)*, 25(4):330–337, 1977. ISSN 0096-3518. doi: 10.1109/TASSP.1977.1162961.

A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki. The DET curve in assessment of detection task performance. In *European Conference on Speech Communication and Technology*, pages 1895–1898, 1997.

A. F. Martin and M. A. Przybocki. The NIST speaker recognition evaluations: 1996-2001. In *2001: A Speaker Odyssey - The Speaker Recognition Workshop*, 2001.

A. Martínez and R. Benavente. The AR face database. Technical Report 24, Computer Vision Center, 1998.

M. Marzinzik and B. Kollmeier. Speech pause detection for noise spectrum estimation by tracking power envelope dynamics. *IEEE Transactions on Speech and Audio Processing (TSAP)*, 10(2):109–118, 2002. ISSN 1063-6676. doi: 10.1109/89.985548.

S. Mau, S. Chen, C. Sanderson, and B. Lovell. Video face matching using subset selection and clustering of probabilistic Multi-Region Histograms. In *International Conference of Image and Vision Computing (ICIVC)*, 2010.

C. McCool and L. El Shafey. Notes on Probabilistic Linear Discriminant Analysis. Idiap-Com Idiap-Com-03-2013, Idiap, 6 2013.

C. McCool and S. Marcel. Parts-based face verification using local frequency bands. In *IEEE/IAPR International Conference on Biometrics*, 2009.

C. McCool, S. Marcel, A. Hadid, M. Pietikainen, P. Matejka, J. Cernocky, N. Poh, J. Kittler, A. Larcher, C. Levy, D. Matrouf, J.-F. Bonastre, P. Tresadern, and T. Cootes. Bi-modal person recognition on a mobile phone: Using mobile phone data. In *IEEE International Conference on Multimedia and Expo Workshops (ICMEW): Hot Topics in Mobile Multimedia*, pages 635–640, July 2012. doi: 10.1109/ICMEW.2012.116.

C. McCool, R. Wallace, M. McLaren, L. El Shafey, and S. Marcel. Session variability modelling for face authentication. *IET Biometrics*, 2(3):117–129, September 2013. ISSN 2047-4938.

G. J. Mclachlan and K. E. Basford. *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, New York, 1988.

G. J. Mclachlan and D. Peel. *Finite Mixture Models*. Wiley Series in Probability and Statistics. Wiley-Interscience, October 2000. ISBN 9780471006268.

K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maître. XM2VTSDB: The extended M2VTS database. In *International Conference on Audio and Video-based Biometric Person Authentication (AVBPA)*, pages 72–77, 1999.

S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K. Mullers. Fisher discriminant analysis with kernels. In *Neural Networks for Signal Processing IX*, pages 41–48, 1999.

T. P. Minka. Algorithms for maximum-likelihood logistic regression. Technical Report 758, CMU Statistics Department, 2001.

B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 19(7):696–710, 1997. ISSN 0162-8828. doi: 10.1109/34.598227.

B. Moghaddam and M.-H. Yang. Learning gender with support faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 24(5):707–711, 2002.

B. Moghaddam, W. Wahid, and A. Pentland. Beyond Eigenfaces: Probabilistic matching for face recognition. In *IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, pages 30–35, 1998. doi: 10.1109/AFGR.1998.670921.

P. Motlicek, L. El Shafey, R. Wallace, C. McCool, and S. Marcel. Bi-modal authentication in mobile environments using session variability modelling. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR)*, November 2012.

T. Oliphant. Python for scientific computing. *Computing in Science Engineering*, 9(3):10–20, 2007.

D. V. Ouellette. Schur complements and statistics. *Linear Algebra and its Applications*, 36: 187–295, 1981.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

L. Perera, R. Lopez, and J. Flores. Speaker verification in different database scenarios. *Computación y Sistemas*, 15:17–26, 2011.

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

P. Phillips, P. Flynn, T. Scruggs, K. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek. Overview of the face recognition grand challenge. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 947–954, 2005. doi: 10.1109/CVPR.2005.268.

P. Phillips, P. Flynn, T. Scruggs, K. Bowyer, and W. Worek. Preliminary face recognition grand challenge results. In *International Conference on Automatic Face and Gesture Recognition (FG)*, pages 15–24, 2006. doi: 10.1109/FGR.2006.87.

P. J. Phillips. Support vector machines applied to face recognition. In *Advances in Neural Information Processing Systems 11*, pages 803–809. MIT Press, 1999.

P. J. Phillips, J. R. Beveridge, B. A. Draper, G. Givens, A. J. O'Toole, D. S. Bolme, J. Dunlop, Y. M. Lui, H. Sahibzada, and S. Weimer. An introduction to the Good, the Bad, & the Ugly face recognition challenge problem. In *IEEE International Conference on Automatic Face Gesture Recognition and Workshops (FG)*, pages 346–353, 2011. doi: 10.1109/FG.2011.5771424.

S. Pigeon, P. Druyts, and P. Verlinde. Applying logistic regression to the fusion of the NIST'99 1-speaker submissions. *Digital Signal Processing*, 10(1-3):237–248, 2000.

N. Pinto, J. J. DiCarlo, and D. D. Cox. How far can you get with a modern face recognition test set using only simple features? In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2591–2598, 2009. doi: 10.1109/CVPR.2009.5206605.

N. Poh, C. H. Chan, J. Kittler, S. Marcel, C. McCool, E. Ruá, J. Castro, M. Villegas, R. Paredes, V. Štruc, N. Pavešić, A. Salah, H. Fang, and N. Costen. An evaluation of video-to-video face verification. *IEEE Transactions on Information Forensics and Security (TIFS)*, 5(4):781–801, 2010. ISSN 1556-6013. doi: 10.1109/TIFS.2010.2077627.

K. Price. Anything you can do, i can do better (no you can't)... *Computer Vision, Graphics, and Image Processing*, 36:387–391, March 1986.

S. J. Prince and J. H. Elder. Probabilistic linear discriminant analysis for inferences about identity. In *IEEE International Conference on Computer Vision (ICCV)*, volume 0, pages 1–8, 2007. ISBN 978-1-4244-1630-1. doi: http://doi.ieeecomputersociety.org/10.1109/ICCV.2007.4409052.

M. Pronobis and M. Magimai-Doss. Integrating audio and vision for robust automatic gender recognition. Idiap-RR Idiap-RR-73-2008, Idiap, 11 2008.

J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., 1993. ISBN 1-55860-238-0.

L. Rabiner and B.-H. Juang. *Fundamentals of Speech Recognition*. Prentice-Hall, Inc., 1993. ISBN 0-13-015157-2.

K. Ramírez-Gutiérrez, D. Cruz-Pérez, and H. Pérez-Meana. Face recognition and verification using histogram equalization. In *WSEAS International Conference on Applied Computer Science*, ACS'10, pages 85–89, 2010. ISBN 978-960-474-231-8.

D. A. Reynolds. *A Gaussian mixture modeling approach to text-independent speaker identification*. Ph.D. dissertation, Georgia Institute of Technology, 1992.

D. A. Reynolds. Speaker identification and verification using Gaussian mixture speaker models. *Speech Communication*, 17(1–2):91–108, 1995a. ISSN 0167-6393. doi: 10.1016/0167-6393(95)00009-D.

D. A. Reynolds. Automatic speaker recognition using Gaussian mixture speaker models. *The Lincoln Laboratory Journal*, pages 173–192, 1995b.

D. A. Reynolds. Comparison of background normalization methods for text-independent speaker verification. In *European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 963–966, September 1997.

D. A. Reynolds. An overview of automatic speaker recognition technology. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 4, pages 4072–4075, 2002. doi: 10.1109/ICASSP.2002.5745552.

D. A. Reynolds and R. C. Rose. Robust text-independent speaker identification using Gaussian mixture speaker models. *IEEE Transactions on Speech and Audio Processing (TSAP)*, 3(1):72–83, 1995. ISSN 1063-6676. doi: 10.1109/89.365379.

D. A. Reynolds and P. A. Torres-Carrasquillo. Approaches and applications of audio diarization. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 5, pages 953–956, 2005. doi: 10.1109/ICASSP.2005.1416463.

D. A. Reynolds, T. F. Quatieri, and R. B. Dunn. Speaker verification using adapted Gaussian mixture models. *Digital Signal Processing*, 10(1-3):19–41, January 2000. ISSN 1051-2004. doi: 10.1006/dspr.1999.0361.

M. Riedmiller and H. Braun. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In *IEEE International Conference on Neural Networks*, pages 586–591, 1993.

A. E. Rosenberg. Listener performance in speaker verification tasks. *IEEE Transactions on Audio and Electroacoustics (TAE)*, 21(3):221–225, 1973. ISSN 0018-9278. doi: 10.1109/TAU.1973.1162454.

A. Ross and A. Jain. Information fusion in biometrics. *Pattern Recognition Letters*, 24(13):2115–2125, 2003.

S. Roweis. EM algorithms for PCA and SPCA. In *Advances in Neural Information Processing Systems (NIPS)*, pages 626–632. MIT Press, 1998.

H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 203–208, 1996. doi: 10.1109/CVPR.1996.517075.

A. Roy, M. Magimai-Doss, and S. Marcel. A fast parts-based approach to speaker verification using boosted slice classifiers. *IEEE Transactions on Information Forensics and Security*, 7:241–254, 2012.

D. L. Ruderman and W. Bialek. Statistics of natural images: Scaling in the woods. In *Advances in Neural Information Processing Systems (NIPS)*, pages 551–558, 1993.

R. Saedi, K. A. Lee, T. Kinnunen, T. Hasan, B. Fauve, P.-M. Bousquet, E. Khoury, P. L. S. Martinez, J. M. K. Kua, C. You, H. Sun, A. Larcher, P. Rajan, V. Hautamäki, C. Hanilci, B. Braithwaite, G.-H. Rosa, S. O. Sadjadi, G. Liu, H. Boril, N. Shokouhi, D. Matrouf, L. El Shafey, P. Mowlaee, J. Epps, T. Thiruvaran, D. Van Leeuwen, B. Ma, H. Li, J.-F. Bonastre, S. Marcel, J. Mason, and E. Ambikairajah. I4U submission to NIST SRE 2012: a large-scale collaborative effort for noise-robust speaker verification. In *INTERSPEECH*, August 2013.

C. Sanderson and K. K. Paliwal. Fast features for face authentication under illumination direction changes. *Pattern Recognition Letters*, 24(14):2409–2419, October 2003. ISSN 0167-8655. doi: 10.1016/S0167-8655(03)00070-9.

E. Scheirer and M. Slaney. Construction and evaluation of a robust multifeature speech/music discriminator. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 2, pages 1331–1334, 1997.

B. Schuller, S. Steidl, A. Batliner, F. Burkhardt, L. Devillers, C. Muller, and S. Narayanan. The INTERSPEECH 2010 paralinguistic challenge. In *INTERSPEECH*, pages 2794–2797, September 2010.

D. Shah, K. J. Han, and S. S. Narayanan. A low-complexity dynamic face-voice feature fusion approach to multimodal person recognition. In *IEEE International Symposium on Multimedia*, pages 24–31, 2009.

A. Shashua and T. Riklin-Raviv. The quotient image: Class-based re-rendering and recognition with varying illuminations. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 23(2):129–139, 2001. ISSN 0162-8828. doi: 10.1109/34.908964.

L. Shen, N. Zheng, S. Zheng, and W. Li. Secure mobile services by face and speech based personal authentication. In *IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS)*, pages 97–100, 2010.

P. Sinha, B. Balas, Y. Ostrovsky, and R. Russell. Face recognition by humans: Nineteen results all computer vision researchers should know about. *Proceedings of the IEEE*, 94(11):1948–1962, November 2006. ISSN 0018-9219. doi: 10.1109/JPROC.2006.884093.

L. Sirovich and M. Kirby. Low-dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America A*, 4(3):519–524, March 1987. doi: 10.1364/JOSAA.4.000519.

F. Soong, A. Rosenberg, L. Rabiner, and B.-H. Juang. A vector quantization approach to speaker recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 10, pages 387–390, 1985. doi: 10.1109/ICASSP.1985.1168412.

L. J. Spreeuwers, A. J. Hendrikse, and K. J. Gerritsen. Evaluation of automatic face recognition for automatic border control on actual data recorded of travellers at Schiphol Airport. In *Proceedings of the 11th International Conference of the Biometrics Special Interest Group (BIOSIG)*, pages 1–6, September 2012.

J. Stallkamp, H. K. Ekenel, H. Erdoğan, R. Stiefelhagen, and A. Erçil. Video-based driver identification using local appearance face recognition. In *Workshop on DSP in Mobile and Vehicular Systems*, June 2007.

H. Steinhaus. Sur la division des corps matériels en parties. *Bulletin de l'Académie Polonaise des Sciences, Classe 3*, 4:801–804, 1957. ISSN 0001-4095.

A. Stolcke, S. S. Kajarekar, L. Ferrer, and E. Shrinberg. Speaker recognition with session variability normalization based on MLLR adaptation transforms. *IEEE Transactions on Audio, Speech, and Language Processing (TASLP)*, 15(7):1987–1998, 2007. ISSN 1558-7916. doi: 10.1109/TASL.2007.902859.

O. M. Strand and A. Egeberg. Cepstral mean and variance normalization in the model domain. In *COST278 and ISCA Tutorial and Research Workshop (ITRW) on Robustness Issues in Conversational Interaction*, August 2004.

N. Sun, W. Zheng, C. Sun, C. Zou, and L. Zhao. Gender classification based on boosting local binary pattern. 3972:194–201, 2006. doi: 10.1007/11760023_29.

K.-K. Sung and T. Poggio. Example-based learning for view-based human face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 20(1):39–51, 1998. ISSN 0162-8828. doi: 10.1109/34.655648.

X. Tan and B. Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE Transactions on Image Processing (TIP)*, 19(6):1635–1650, 2010. ISSN 1057-7149. doi: 10.1109/TIP.2010.2042645.

The HDF Group. Hierarchical Data Format, Version 5. http://www.hdfgroup.org/HDF5, 2000-2010.

O. Thyes, R. Kuhn, P. Nguyen, and J.-C. Junqua. Speaker identification and verification using Eigenvoices. In *ICSLP/INTERSPEECH: Proccedings of the 6th International Conference on Spoken Language Processing*, pages 242–245, October 2000.

M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 61:611–622, 1999.

R. Tucker. Voice activity detection using a periodicity measure. *IEE Proceedings I (Communications, Speech and Vision)*, 139(4):377–380, 1992. ISSN 0956-3776.

M. A. Turk and A. P. Pentland. Face recognition using Eigenfaces. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–591, 1991a. doi: 10.1109/CVPR.1991.139758.

M. A. Turk and A. P. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3 (1):71–86, January 1991b. ISSN 0898-929X. doi: 10.1162/jocn.1991.3.1.71.

C. Vair, D. Colibro, F. Castaldo, E. Dalmasso, and P. Laface. Loquendo - Politecnico di Torino's 2006 NIST speaker recognition evaluation system. In *INTERSPEECH*, pages 1238–1241, 2007.

P. Vandewalle. Code sharing is associated with research impact in image processing. *Computing in Science and Engineering*, pages 42–47, 2012.

V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., 1995. ISBN 0-387-94559-8.

T. L. Veldhuizen. Arrays in Blitz++. In *Proceedings of the International Scientific Computing in Object-Oriented Parallel Environments*, pages 223–230. Springer, 1998.

P. Verlinde and G. Cholet. Comparing decision fusion paradigms using k-nn based classifiers, decision trees and logistic regression in a multi-modal identity verification application. In *International Conference on Audio and Video-Based Biometric Person Authentication (AVBPA)*, pages 188–193, 1999.

P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 511–518, 2001. doi: 10.1109/CVPR.2001.990517.

R. J. Vogt and S. Sridharan. Explicit modelling of session variability for speaker verification. *Computer Speech & Language*, 22(1):17–38, January 2008. ISSN 0885-2308. doi: 10.1016/j.csl.2007.05.003.

R. J. Vogt, B. J. Baker, and S. Sridharan. Modelling session variability in text-independent speaker verification. In *EUROSPEECH/INTERSPEECH: Proceedings of the 9th European Conference on Speech Communication and Technology*, pages 3117–3120, Lisbon, Portugal, 2005. International Speech Communication Association (ISCA).

L. Walawalkar, M. Yeasin, A. Narasimhamurthy, and R. Sharma. Support vector learning for gender classification using audio and visual cues: A comparison. *International Journal of Pattern Recognition and Artificial Intelligence*, 17(03):417–439, 2003.

R. Wallace and M. McLaren. Total variability modelling for face verification. *IET Biometrics*, 1: 188–199, December 2012. ISSN 2047-4938.

R. Wallace, M. McLaren, C. McCool, and S. Marcel. Inter-session variability modelling and joint factor analysis for face authentication. In *International Joint Conference on Biometrics (IJCB)*, pages 1–8, 2011. doi: 10.1109/IJCB.2011.6117599.

R. Wallace, M. McLaren, C. McCool, and S. Marcel. Cross-pollination of normalization techniques from speaker to face authentication using Gaussian mixture models. *IEEE Transactions on Information Forensics and Security*, 7(2):553–562, 2012. ISSN 1556-6013. doi: 10.1109/TIFS.2012.2184095.

H. Wang, S. Z. Li, Y. Wang, and J. Zhang. Self quotient image for face recognition. In *IEEE International Conference on Image Processing (ICIP)*, volume 2, pages 1397–1400 Vol.2, 2004. doi: 10.1109/ICIP.2004.1419763.

X. Wang, C. Zhang, and Z. Zhang. Boosted multi-task learning for face verification with applications to web image and video search. In *IEEE Internation Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 142–149, 2009. doi: 10.1109/CVPR.2009.5206736.

B. C. Welsh and D. P. Farrington. *Crime prevention effects of closed circuit television: a systematic review*. Home Office, August 2002.

L. Wiskott, J.-M. Fellous, N. Krüger, and C. von der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 19(7):775–779, 1997. ISSN 0162-8828. doi: 10.1109/34.598235.

M.-H. Yang, D. Roth, and N. Ahuja. A SNoW-based face detector. In *Advances in Neural Information Processing Systems 12 (NIPS)*, pages 855–861. MIT Press, 1999.

B. Yegnanarayana and S. Kishore. AANN: an alternative to GMM for pattern recognition. *Neural Networks*, 15(3):459 – 469, 2002. ISSN 0893-6080. doi: 10.1016/S0893-6080(02)00019-9.

L. Zhang, R. Chu, S. Xiang, S. Liao, and S. Z. Li. Face detection based on multi-block LBP representation. In *International Conference on Advances in Biometrics (ICB)*, volume 4642, pages 11–18, 2007. ISBN 978-3-540-74548-8. doi: 10.1007/978-3-540-74549-5_2.

W. Zhang, S. Shan, W. Gao, X. Chen, and H. Zhang. Local Gabor binary pattern histogram sequence (LGBPHS): a novel non-statistical model for face representation and recognition. In *IEEE International Conference on Computer Vision (ICCV)*, volume 1, pages 786–791, October 2005. doi: 10.1109/ICCV.2005.147.

Laurent El Shafey

Curriculum Vitae

Rue d'Oche 14
1920 Martigny, Switzerland
+41 76 47 57 297
+41 27 565 64 02
laurent.el-shafey@idiap.ch
www.idiap.ch/~lelshafey
Nationality: French

General Area of Interest and Expertise

- Machine Learning, Biometrics, Computer Vision
- Software Engineering

Current

2010–2014 **Ph.D. student**, *Idiap Research Institute*, Martigny, Switzerland.
Enrolled in the Electrical Engineering Doctoral program (EDEE) at the Ecole Polytechnique Fédérale de Lausanne, Switzerland

current topic *Biometric Face and Speaker Recognition*

thesis director Prof. Hervé Bourlard

thesis co-director Dr. Sébastien Marcel

description My thesis aims at investigating probabilistic models for face, speaker and bimodal (face and speech) recognition. In addition, my work is geared towards reproducible research, by actively contributing to the development of the open source signal processing and machine learning toolbox Bob.

Experience

February 2013 **Ph.D. Secondment**, *Netherlands Forensic Institute*, The Hague, Netherlands.

April 2012 – June 2012) **Ph.D. Secondment**, *Intelligent Systems' laboratory, Halmstad University*, Halmstad, Sweden.

October 2007 – April 2008 (Part time) **Teaching assistant**, *Multimedia Communication Lab, Technische Universität Darmstadt*, Darmstadt, Germany.
Communication Networks II lecture (hold by Prof. R. Steinmetz)

June 2006 – August 2006 **Internship**, *Safran*, Osny, France.
R&D Printing Terminals Business Unit

June 2005 – July 2005 **Internship**, *Alcatel-Lucent*, Paris, France.
Intellectual Property Unit

Education

2006–2008 **Diplom Informatiker (M.Sc. in Computer Science; Overall: Very Good) (double degree program)**, *Technische Universität Darmstadt (TUD)*, Darmstadt, Germany.

- Diploma Thesis in Computer Vision (Multimodal Interactive Systems Department)
- Supervisor: Dr. Christian Wojek (Prof. Bernt Schiele Laboratory)

2004–2006 **M.Sc. in Electrical Engineering (Class of 2007)**, *SUPELEC (Ecole Supérieure d'Electricité)*, Gif-sur-Yvette, France.

- French Grande Ecole of engineering

2001–2004 **Mathématiques Supérieures et Spéciales MPSI/MP**, *Lycée Pasteur*, Neuilly-sur-Seine, France.

- Intensive preparation for French Grande Ecole of engineering

2001 **Baccalauréat, specializing in Mathematics**, *Lycée Sainte Croix*, Neuilly-sur-Seine, France.

Research Projects Directly Involved in

- Bayesian Biometrics for Forensics (BBfor2, European FP7 project)
- Biometrics Evaluation and Testing (BEAT, European FP7 project)

Additional Expertise

Programming Languages	C++, Python, Bash, C, Java, SQL, Caml, Scheme, Go
Applications and Tools	Matlab, Latex, Maple, GCC, GDB, Valgrind, Vim
Operating Systems	Linux, Microsoft Windows, Mac OS X

Languages

French	native
English	fluent
German	basic

Publications

In journals

[1] Laurent El Shafey, Chris McCool, Roy Wallace, and Sébastien Marcel. A Scalable Formulation of Probabilistic Linear Discriminant Analysis: Applied to Face Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1788–1794, July 2013. `https://pypi.python.org/pypi/xbob.paper.tpami2013`.

[2] Laurent El Shafey, André Anjos, Manuel Günther, Elie Khoury, Ivana Chingovska, François Moulin, and Sébastien Marcel. Bob: A Free Library for Reproducible Machine Learning. *Submitted*, July 2013. `https://pypi.python.org/pypi/xbob.paper.example`.

[3] Chris McCool, Roy Wallace, Mitchell McLaren, Laurent El Shafey, and Sébastien Marcel. Session variability modelling for face authentication. *IET Biometrics*, 2(3):117–129, September 2013.

[4] Elie Khoury, Laurent El Shafey, Chris McCool, Manuel Günther, and Sébastien Marcel. Bi-modal biometric authentication on mobile phones in challenging conditions. *To appear in Image and Vision Computing*, August 2014.

[5] Manuel Günther, Laurent El Shafey, and Sébastien Marcel. 2D Face Recognition: An Experimental and Reproducible Research Survey. *Submitted*, June 2013.

[6] Abhishek Dutta, Manuel Günther, Laurent El Shafey, Sébastien Marcel, Raymond Veldhuis, and Luuk Spreeuwers. Impact of Eye Detection Error on Face Recognition Performance. *Submitted*, August 2013.

In conference proceedings

[1] Laurent El Shafey, Roy Wallace, and Sébastien Marcel. Face Verification using Gabor Filtering and Adapted Gaussian Mixture Models. In *Proceedings of the 11th International Conference of the Biometrics Special Interest Group*, pages 397–408. GI-Edition, September 2012.

[2] Laurent El Shafey, Elie Khoury, and Sébastien Marcel. Audio-Visual Gender Recognition in Uncontrolled Environment Using Variability Modeling Techniques. In *Submitted*, 2013. `https://pypi.python.org/pypi/xbob.gender.bimodal`.

[3] André Anjos, Laurent El Shafey, Roy Wallace, Manuel Günther, Chris McCool, and Sébastien Marcel. Bob: a free signal processing and machine learning toolbox for researchers. In *Proceedings of the ACM Multimedia Conference*, October 2012. `www.idiap.ch/software/bob`.

[4] Petr Motlicek, Laurent El Shafey, Roy Wallace, Chris McCool, and Sébastien Marcel. Bi-Modal Authentication in Mobile Environments Using Session Variability Modelling. In *Proceedings of the 21st International Conference on Pattern Recognition*, November 2012.

[5] Elie Khoury, Laurent El Shafey, and Sébastien Marcel. The Idiap Speaker Recognition Evaluation System at NIST SRE 2012. In *NIST Speaker Recognition Conference*. NIST, December 2012. `https://pypi.python.org/pypi/xbob.spkrec.nist_sre12`.

[6] Elie Khoury, Manuel Günther, Laurent El Shafey, and Sébastien Marcel. On the Improvements of Uni-modal and Bi-modal Fusions of Speaker and Face Recognition

for Mobile Biometrics. In *Biometric Technologies in Forensic Science*, October 2013. `https://pypi.python.org/pypi/xbob.paper.BTFS2013`.

[7] Elie Khoury, Bostjan Vesnicer, Javier Franco-Pedroso, Ricardo Violato, Zenelabidine Boulkenafet, Luis-Miguel Mazaira Fernandez, Mireia Diez, Justina Kosmala, Houssemeddine Khemiri, Tomas Cipr, Rahim Saedi, Manuel Günther, Jerneja Zganec-Gros, Ruben Zazo Candil, Flávio Simões, Messaoud Bengherabi, Augustin Alvarez Marquina, Mikel Penagarikano, Alberto Abad, Mehdi Boulayemen, Petr Schwarz, David Van Leeuwen, Javier Gonzalez-Domınguez, Mário Uliani Neto, Elhocine Boutellaa, Pedro Gomez Vilda, Amparo Varona, Dijana Petrovska-Delacretaz, Pavel Matejka, Joaquin Gonzalez-Rodrıguez, Tiago de Freitas Pereira, Farid Harizi, Luis Javier Rodriguez-Fuentes, Laurent El Shafey, Marcus Angeloni, German Bordel, Gérard Chollet, and Sébastien Marcel. The 2013 Speaker Recognition Evaluation in Mobile Environment. In *The 6th IAPR International Conference on Biometrics*, June 2013.

[8] Rahim Saedi, Kong Aik Lee, Tomi Kinnunen, Tawfik Hasan, Benoit Fauve, Pierre-Michel Bousquet, Elie Khoury, Pablo Luis Sordo Martinez, Jia Min Karen Kua, Changhuai You, Hanwu Sun, Anthony Larcher, Padmanabhan Rajan, Ville Hautamäki, Cemal Hanilci, Billy Braithwaite, Gonzalez-Hautamäki Rosa, Seyed Omid Sadjadi, Gang Liu, Hynek Boril, Navid Shokouhi, Driss Matrouf, Laurent El Shafey, Pejman Mowlaee, Julien Epps, Tharmarajah Thiruvaran, David Van Leeuwen, Bin Ma, Haizhou Li, Jean-François Bonastre, Sébastien Marcel, John Mason, and Eliathamby Ambikairajah. I4U Submission to NIST SRE 2012: a large-scale collaborative effort for noise-robust speaker verification. In *INTERSPEECH*, August 2013.

[9] Elie Khoury, Laurent El Shafey, and Sébastien Marcel. Spear: An open source toolbox for speaker recognition based on Bob. In *Proceedings of the 39th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014. `https://pypi.python.org/pypi/bob.spear`.