Imperial College London
Department of Computing

Machine Learning Techniques for Automated Analysis of Facial Expressions

Ognjen Rudovic

December, 2013

Supervised by Prof. Maja Pantic

Submitted in part fulfilment of the requirements for the degree of PhD in Computing and the Diploma of Imperial College London. This thesis is entirely my own work, and, except where otherwise indicated, describes my own research.

The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work.

Abstract

Automated analysis of facial expressions paves the way for numerous next-generation-computing tools including affective computing technologies (proactive and affective user interfaces), learner-adaptive tutoring systems, medical and marketing applications, etc. In this thesis, we propose machine learning algorithms that head toward solving two important but largely understudied problems in automated analysis of facial expressions from facial images: pose-invariant facial expression classification, and modeling of dynamics of facial expressions, in terms of their temporal segments and intensity. The methods that we propose for the former represent the pioneering work on pose-invariant facial expression analysis. In these methods, we use our newly introduced models for pose normalization that achieve successful decoupling of head pose and expression in the presence of large out-of-plane head rotations, followed by facial expression classification. This is in contrast to most existing works, which can deal only with small in-plane head rotations. We derive our models for pose normalization using the Gaussian Process (GP) framework for regression and manifold learning. In these, we model the structure encoded in relationships between facial expressions from different poses and also in facial shapes. This results in the models that can successfully perform pose normalization either by warping facial expressions from non-frontal poses to the frontal pose, or by aligning facial expressions from different poses on a common expression manifold. These models solve some of the most important challenges of pose-invariant facial expression classification by being able to generalize to various poses and expressions from a small amount of training data, while also being largely robust to corrupted image features and imbalanced examples of different facial expression categories. We demonstrate this on the task of pose-invariant facial expression classification of six basic emotions.

The methods that we propose for temporal segmentation and intensity estimation of facial expressions represent some of the first attempts in the field to model facial expression dynamics. In these methods, we use the Conditional Random Fields (CRF) framework to define dynamic models that encode the spatio-temporal structure of the expression data, reflected in ordinal and temporal relationships between temporal segments and intensity levels of facial expressions. We also propose several means of addressing the subject variability in the data by simultaneously exploiting various priors, and the effects of heteroscedasticity and context of target facial expressions. The resulting models are the first to address simultaneous classification and temporal segmentation of facial expressions of six basic emotions, and dynamic modeling of intensity of facial expressions of pain. Moreover, the context-sensitive model that we propose for intensity estimation of spontaneously displayed facial expressions of pain and Action Units (AUs), is the first approach in the field that performs context-sensitive modeling of facial expressions in a principled manner.

Acknowledgements

I would like to thank Prof. Maja Pantic for her invaluable guidance and support over the course of the time I spent doing my PhD. I am also tremendously grateful to Prof. Vladimir Pavlovic, for very fruitful discussions and his advice on the direction my research should take. I would also like to thank Dr. Ioannis Patras for his support in the early stage of my PhD. I want also to express my heartfelt gratitude to Susan Peneycad, Dr. Mark Shuttleworth and Prof. Charles Drage for their unceasing support during writing of my thesis. My appreciation also goes to my colleagues at Imperial College London, and especially the i-BUG group, for useful research discussions, but most of all for being such great friends. My biggest thanks go to my family for their unconditional love and support. This thesis is dedicated to my grandparents.

Contents

1 Introduction **9**
1.1 Introduction . . . 9
1.2 Problem Space . . . 11
1.3 Potential Applications . . . 17
1.4 Contributions . . . 19
1.5 Thesis Outline . . . 28

2 Automated Analysis of Facial Expressions: The State of The Art **29**
2.1 Facial Expression Analysis: Overview . . . 29
2.2 Pre-processing and Feature Extraction . . . 30
2.3 Machine Analysis of Facial Expressions . . . 36
2.4 Relation to Our Work . . . 48

I Pose-invariant Facial Expression Analysis from Static Images **53**

3 Gaussian Processes for Pose-invariant Facial Expression Analysis **55**
3.1 Introduction . . . 55
3.2 Why GPs? . . . 56
3.3 Gaussian Processes . . . 57
3.4 Summary of Proposed Methods . . . 60

4 Coupled Gaussian Processes for Pose-invariant Facial Expression Classification **63**
4.1 Introduction . . . 63
4.2 Methodology . . . 64
4.3 Experiments . . . 70
4.4 Conclusions . . . 80

5 Shape-conformed Gaussian Process Regression for Pose Normalization 83
5.1 Introduction 83
5.2 Methodology 84
5.3 Experiments 89
5.4 Conclusions 94

6 Discriminative Shared Gaussian Process Latent Variable Model for Multi-view Facial Expression Classification 97
6.1 Introduction 97
6.2 Methodology 99
6.3 Experiments 104
6.4 Conclusions 107

7 Pose-invariant Facial Expression Analysis: Conclusions and Future Work 109

II Analysis of Facial Expression Dynamics from Image Sequences 113

8 Conditional Ordinal Random Fields (CORF) for Analysis of Facial Expression Dynamics 115
8.1 Introduction 115
8.2 Why CORF? 116
8.3 Conditional Ordinal Random Fields 117
8.4 Summary of Proposed Methods 122

9 Multi-output CORF for Classification and Temporal Segmentation of Facial Expressions of Emotions 125
9.1 Introduction 125
9.2 Methodology 127
9.3 Experiments 135
9.4 Conclusions 141

10 Kernel CORF for Temporal Segmentation of AUs 143
10.1 Introduction 143
10.2 Methodology 144
10.3 Experiments 149
10.4 Conclusions 152

11 Heteroscedastic KCORF for Intensity Estimation of Facial Expressions of Pain **153**
11.1 Introduction 153
11.2 Methodology 155
11.3 Experiments 158
11.4 Conclusions 162

12 Context-sensitive CORF for Intensity Estimation of AUs and Facial Expressions of Pain **163**
12.1 Introduction 163
12.2 Methodology 165
12.3 Experiments 172
12.4 Conclusions 183

13 Analysis of Facial Expression Dynamics: Conclusions and Future Work **187**

14 Final Conclusions **193**

Bibliography **195**

Chapter 1

Introduction

Contents

1.1 Introduction 9
1.2 Problem Space 11
1.3 Potential Applications 17
1.4 Contributions 19
1.5 Thesis Outline 28

1.1 Introduction

The face is one of the most powerful channels of nonverbal communication [51]. Facial expressions communicate emotion, and signal intentions, alertness, pain, and personality traits. They also regulate intersubjective behavior, and communicate psychiatric and biomedical status, among other functions [61, 51, 144]. So, it is not surprising that facial expression has been a focus of research into human behavior for over a hundred years. In the seminal work on facial expression of emotion [49], Charles Darwin described in detail the specific facial expressions associated with emotions in animals and humans. He argued that all mammals show emotions reliably in their faces. In the later influential study on facial expression of humans [62], Paul Ekman suggested that the six basic emotions (anger, fear, disgust, happiness, sadness and surprise, see Fig.1.2) are universally displayed across different cultures. In more recent works, Ekman & colleagues [98, 42, 60, 63, 61] defined rules for describing and analyzing facial expressions of emotions, but also of cognitive states, such as interest, boredom, confusion, stress, etc. Because of the theoretical interest of cognitive and medical scientists, and also many practical applications in medicine and for human-computer interaction, among others, the need to automate the analysis of facial expressions is ever growing.

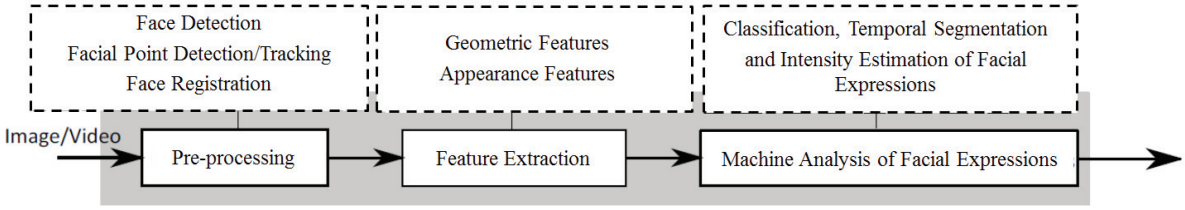


Figure 1.1: A typical system for automated analysis of facial expressions. Given an input image or image sequence, the first step consists of pre-processing of the target image(s). This is performed by (i) localizing a face, (ii) detecting a set of facial points, which are then used to perform (iii) face registration. Once the face is registered, the next step is facial feature extraction. Different geometric and/or appearance features can be used, which are usually chosen depending on the target task. The final step is machine interpretation of facial expressions.

A typical system for automated analysis of facial expressions is based on computer algorithms that attempt to interpret facial motions and facial feature changes from facial images [189]. The outline of such a system is given in Fig.1.1. The system presented performs facial expression analysis in three steps: image pre-processing, facial feature extraction, and machine interpretation of target facial expressions. Although humans perform these three steps with little or no effort, development of an automated system that accomplishes this is rather difficult [144]. For this reason, automated analysis of facial expressions has been an active research area within the computer vision and machine learning community over the last 15 years. The work presented in this thesis proposes different machine learning algorithms for addressing some of the most commonly encountered problems in automated analysis of facial expressions. In particular, we focus on two important problems: pose-invariant facial expression classification from static images, and analysis of facial expression dynamics, in terms of classification of temporal segments and intensity of facial expressions from image sequences.

In the remainder of this Chapter, we first describe the general problem space of facial expression analysis, and introduce the most commonly encountered modeling challenges. We then discuss several practical applications that spring from research into automated analysis of facial expressions. We then give a 'big picture' describing the problems addressed in this thesis and how we solved them. We then describe in more detail our main contributions, and give the outline of the thesis.

Figure 1.2: An example of prototypic facial expressions of six basic emotions (disgust, happiness, sadness, anger, fear and surprise) from [146].

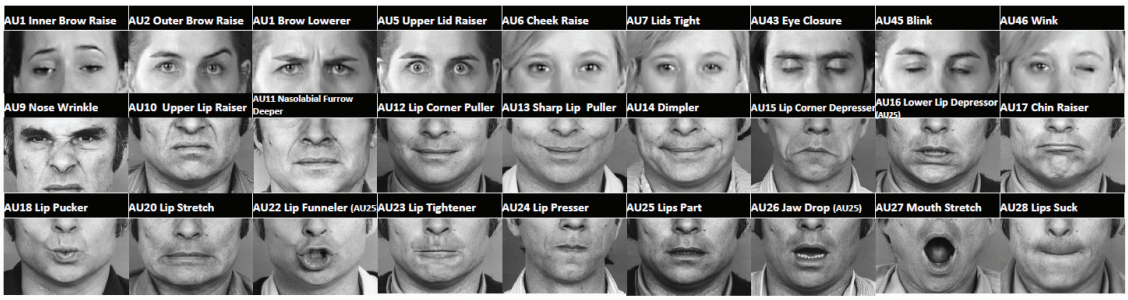


Figure 1.3: 32 atomic facial muscle actions named Action Units (AUs): 9 AUs in the upper face, 18 in the lower face.

1.2 Problem Space

1.2.1 Level of Description

Facial expressions can be described at different levels [189]. Two main streams in the current research on automatic analysis of facial expressions consider facial affect (emotion) and facial muscle action (action unit) [144]. These two streams stem directly from the message and sign judgment approaches for facial expression measurement [42]. The message judgment aims to directly decode the meaning conveyed by a facial display (e.g., in terms of the six basic emotions (Fig.1.2) proposed by Ekman [62]). The sign judgment instead aims to study the physical signal used to transmit the message (such as raised cheeks or depressed lips). Thus, the message judgment is all about interpretation, where the sign judgment attempts to be objective, leaving inference about the conveyed message to a higher order decision making [144]. To describe the latter, the *Facial Action Coding System* (FACS) [60] defines 32 atomic facial muscle actions named Action Units (AUs): 9 AUs in the upper face, 18 in the lower face (Fig.1.3), and 5 AUs that cannot be exclusively attributed to either. Additionally it encodes a number of miscellaneous actions, such as eye gaze direction and head pose, and 14 descriptors

for miscellaneous actions. Over the past 30 years, extensive research has been conducted by psychologists and neuroscientists using the FACS in various aspects of facial expression analysis. For example, the FACS has been used to demonstrate differences between polite and amused smiles [8], deception detection [65], facial signals of suicidal and non-suicidal depressed patients [79], as well as voluntary and evoked expressions of pain [59, 61].

When it comes to automated analysis of facial expressions, most of the systems developed so far employ the message judgment approach [25, 153]. This is mainly due to the simplicity of coding the facial expressions into a small number of affective states (e.g., the six basic emotions). By contrast, the automated analysis of AUs is far more challenging. This is because of a large number of possible AUs, more subtle changes in facial texture, as well as the burden of manually coding of AUs in order to build training datasets. This is even more true in the case of intensity and temporal segments of AUs (Sec.1.2.3). Nevertheless, the research trend is shifting toward automated analysis of AUs [144], as they provide a more comprehensive and objective way of describing facial expressions, especially when dealing with spontaneously displayed facial expressions (Sec.1.2.2). Also, since every possible facial expression can be described as a combination of AUs [60], the sign-judgment approach can be used to infer affective states defined by the message judgment approach (see Table 1.1).

In addition to the categorical description of affective states in the message judgment approach (e.g., in terms of the six basic emotions), affective states can also be described using the continuous (dimensional) model [163]. This model suggests that emotional states can be described in a two-dimensional circular space, containing arousal and valence dimensions. The valence describes the pleasantness, with positive (pleasant) on one end (e.g. happiness), and negative (unpleasant) on the other (e.g. disgust). The other dimension is arousal. For example, sadness is described with low arousal level, whereas surprise with high arousal level [174]. Whether the categorical or dimensional approach is better for describing affective states is open to debate. In this thesis, we adopt the categorical approach. For related works on the dimensional approach to emotion analysis, see [138].

1.2.2 Spontaneous vs. Posed Facial Expressions

The difference between spontaneous and posed facial expressions is an important factor that must be considered when designing systems for automated analysis of facial expressions. Because the latter are usually recorded in more constrained environments by asking subjects to simulate the expression of the target affective state, both the semantic content and the physical realization of spontaneous and posed facial expressions differ considerably [59, 61].

Table 1.1: The list of AUs involved in some of the facial expressions described using the message judgment approach.

	AUs
FACS:	upper face: 1, 2, 3, 4, 5, 6, 7, 43, 45, 46; lower face: 9, 10, 11, 12, 13, 15, 16, 17, 18, 20, 21, 22, 23, 24, 25, 26, 27, 28; other: 31, 37, 38
anger:	4, 5, 7, 10, 17, 22, 23, 24, 25, 26
disgust:	9, 10, 16, 17, 25, 26
fear:	1, 2, 4, 5, 20, 25, 26, 27
happiness:	6, 12, 25
sadness:	1, 4, 6, 11, 15, 17
surprise:	1, 2, 5, 26, 27
pain:	4, 6, 7, 9, 10, 12, 20, 25, 26, 27, 43
cluelessness:	1, 2, 5, 15, 17, 22
speech:	10, 14, 16, 17, 18, 20, 22, 23, 24, 25, 26, 28

Neuroanatomical evidence suggests that spontaneous and posed facial expressions are controlled by different mechanisms, resulting in different activation patterns of the facial muscles [59, 61]. Specifically, spontaneously displayed facial expressions are characterized by synchronized, smooth, symmetrical, and reflex-like muscle movements, while the posed ones are subject to volitional real-time control and tend to be less smooth [144]. This, in turn, significantly affects the dynamics of the displayed facial expressions in terms of their temporal segments and intensity, as well as the co-occurrences of AUs. These are key factors for distinguishing between various affective states (see Sec. 1.2.3). Therefore, although the majority of automated systems for facial expression analysis have been developed for posed facial displays, mainly due to data availability, their performance is expected to downgrade substantially when applied to spontaneous facial displays. This has also been emphasized by cognitive and computer scientists whose main criticism of the existing works is that the methods designed using the posed data are not applicable in real-life situations, where there are subtle changes in facial expressions of the displayed facial behavior rather than the exaggerated changes that typify the posed expressions [152]. In addition, the effects of head pose (as the subjects tend to move while being recorded), and illumination changes (especially in outdoor environments), are far more pronounced when dealing with spontaneous facial expressions [189].

Designing the models using posed data is important. It allows us to analyze the influence of different factors on facial expressions in controlled environments, such as the head pose variation. However, due to its practical applicability, current research is shifting toward automated analysis of spontaneous facial expressions (produced in a reflex-like manner). With the

release of new datasets, many works on automated analysis of spontaneous facial expressions have emerged over the last several years (e.g., [199, 17, 133]). Yet, most of the works proposed rely on the models that are the same or similar to those used for posed data, despite the fact that the spontaneous data bring new modeling challenges (e.g., how to account for the impact of facial expression dynamics and context), requiring a more sophisticated modeling approach. The models proposed in this thesis try to meet some of those challenges.

1.2.3 Morphology and Dynamics of Facial Expressions

Morphology of Facial Expressions. Morphology and dynamics are two aspects of facial expressions that are crucial for their interpretation [144]. Face morphology refers to the facial configuration observed from static images. As mentioned above, in the message judgment approach, the facial configuration in the target images can be described in terms of the presence/absence of certain affective states (e.g., pain), whereas in the sign-judgment approach, the facial configuration is described by the presence/absence of AUs using FACS. However, using the latter approach is far more difficult due to the large number of possible combinations of AUs (more than 7,000), many of which are commonly observed in spontaneous facial expressions [169]. Besides the burden that human annotators are faced with when coding such facial configurations, automating this process is rather difficult. Most of the systems for automated analysis of AUs perform detection of each AU independently (e.g., [200, 144, 39]). Although this approach may be valid for co-occurring AUs that are additive, it is suboptimal for modeling non-additive AUs, in the case of which one action masks another, or creates a new and distinctive set of appearances [60]. For instance, AU4 (brow lowerer) appears differently depending on whether it occurs alone or in combination with AU1 (inner brow raise). When AU4 occurs alone, the brows are drawn together and lowered. In AU1+4, the brows are drawn together but are raised due to the action of AU1. Thus, for accurate detection of AUs, and, in turn, their successful interpretation in terms of affective states (e.g., emotions), it is important to account for correlations between different AUs. Although this has been addressed in several works that attempt to simultaneously detect multiple AUs (e.g., [192, 191]), it is an open problem that is beyond the scope of this thesis.

While most of the works on automated facial expression analysis have focused on binary description of facial configuration in terms of presence/absence of affective states and/or AUs, a more precise approach to encoding configuration of facial expressions is in terms of their intensity. The intensity of facial expressions usually refers to the relative degree of change in facial expression as compared to a relaxed, neutral facial expression [144]. For example, in the

case of smile, the intensity of the expression can be quantified in terms of the relative degree of upward and outward movement of the corners of the mouth, that is, as the degree of perceivable activity in the Zygomaticus Major muscle (AU12) away from its resting, relaxed state [57]. It has been shown experimentally that the expression decoding accuracy and the perceived intensity of the underlying affective state vary linearly with the physical intensity of the facial display [80]. Therefore, explicit analysis of expression intensity variation is very important for accurate interpretation of facial expressions [144]. In particular, knowing the expression intensity is essential for distinguishing between spontaneous and posed facial expressions, as well as for inferring their meaning. For example, a full-blown smile and a smirk, both coded as AU12 but with different intensities, have very different meanings (e.g., enjoyment vs. sarcasm). To our knowledge, there is not an objective way in the message judgment approach to directly encode the intensity of affective states. However, some authors (e.g., [173, 104]) have proposed describing the intensity of facial expressions of the six basic emotions as a path on a low dimensional manifold of facial features, capturing variation in target facial displays during temporal development of emotion. In the case of the sign-judgment approach, FACS [60] defines the intensity of each AU in a range from absent to maximal intensity on a five-point ordinal scale. This approach is deemed objective for encoding the intensity of AUs, as each intensity score corresponds to a pre-defined level of facial appearance variation.

However, distinguishing different intensities of AUs is not an easy task, for several reasons. First and foremost, intensity of AUs is characterized by subtle variability in the subject-specific facial expressiveness. Namely, each subject may have a different level of expressiveness (e.g., extrovert vs. introvert), which, in turn, makes it difficult to grasp what constitutes the maximal level of change in their facial appearance. Because different people may gesticulate differently, for some the appearance of cheek dimples is their most intense smile, while for others that is just a slight intensity smile. Also, as noted in [60], the intense muscular contractions are usually combined with the subject's physical characteristics to produce changes in appearance that vary across subjects. Second, co-occurrence of AUs affects the criteria for scoring their intensity. The criteria for scoring the intensity of AU7 (lid tightener), for instance, are changed significantly if AU7 appears with a maximal intensity of AU43 (eye closure) [60]. Third, a change in lighting, head position, or transient shadows can give the impression of different AU intensity. These and many other factors make AU intensity coding a far more difficult task than AU detection, even for human coders, let alone for machine analysis. Nevertheless, due to its practical importance for facial expression interpretation, automated analysis of AU intensity has recently received significant attention from many researchers [130, 133].

Dynamics of Facial Expressions. In contrast to face morphology, which can be described from static images, the dynamics of facial expressions are reflected in changes of facial expressions in the temporal domain. These dynamics are typically described in terms of co-occurrences of different AUs over time, as well as in terms of timing, speed and duration of their temporal segments (neutral, onset, apex, and offset) and intensity levels. Here, the neutral temporal segments of an AU refers to the part of an image sequence where there is no manifestation of activation of the muscle corresponding to the target AU. It is followed by the onset segment, where the intensity of the muscle activation increases towards the apex segment, the plateau when the intensity of the muscle activation stabilizes. Finally, the offset segment represents the progressive muscular relaxation towards the neutral phase. Temporal segmentation of an affective state (e.g., surprise) can be attained in the same manner. It is important to note that the temporal segments and the intensity levels account for different aspects of facial expression dynamics. While the former describes dynamics of facial appearance changes relative to the maximum level within an image sequence, the latter does so relative to the overall maximum level of the appearance change.

The dynamics described above are important for interpretation of facial expressions [9]. As emphasized in [212], they are essential for categorization of complex psychological states, such as various types of pain and mood. Furthermore, they represent a critical factor for interpretation of social behaviors such as, for example, social inhibition, embarrassment, amusement and shame [61], while being highly correlated with trustworthiness, dominance, and attractiveness in social interactions [67]. They are also a key parameter for differentiating between posed and spontaneous facial displays [43, 59]. For instance, spontaneous smiles are smaller in amplitude, longer in total duration, and slower in onset and offset time than posed smiles (e.g., a polite smile) [59, 144]. Similarly, the study in [43] showed that spontaneous smiles, in contrast to posed smiles, can have multiple apexes (multiple rises of the mouth corners – AU12) and are accompanied by other AUs that appear either simultaneously with AU12 or follow AU12 within 1 second. The intensity of AUs also influences their dynamics: if AU7 appears in combination with a maximal intensity of AU43 (eye closure), the timing of these AUs changes [60]. In spite of these findings, except for a small number of works (e.g., [191, 200, 102]), the majority of past works on automated analysis of facial expressions do not attempt modeling of their dynamics. Moreover, none of those works exploits dynamics reflected in changes of facial expression intensity. In this thesis, we focus particularly on modeling of dynamics of facial expressions in terms of their temporal segments and intensity.

1.2.4 Context Dependency

Facial expressions do not usually convey one type of message exclusively [144]. For instance, squinted eyes may be interpreted as sensitivity of the eyes to bright light if this action is a reflex, but also as an expression of dislike if it is displayed when seeing someone passing by [152]. Also, as mentioned in Sec.1.2.3, different people gesticulate differently, so the meaning of observed facial expressions may depend on the subject showing them. Similarly, knowing where the subject is (e.g., indoors or outdoors) and what he/she is doing (e.g., whether he is watching a horror or a comedy film) are just some of many factors that can influence the meaning of the displayed facial expressions. Thus, for successful interpretation of facial expressions, it is important to know the context in which the observed expression has been displayed [149]. To summarize the key aspects of the context in which the facial expressions occur, [149] suggested the *W5+* context model. This model answers the context questions: *who* (the observed subject, and, e.g., his/her age and gender), *where* (e.g., environmental characteristics such as illumination), *what* (e.g., the task-related cues such as bowed head while reading), *when* (e.g., the timing of facial actions), *why* (the context stimulus such as humorous videos), and *how* (e.g., the information is passed on by means of facial expression intensity or a combination of AUs). The authors of [149] argue that answering all the context questions, or, depending on the target task, a group of them (particularly the context question *who*), is essential for reliable interpretation of facial expressions. Despite this, most existing approaches to automated analysis of facial expressions are context-insensitive since they attempt to answer only the context question *how*, without taking into account the other context questions. A few approaches perform modeling of facial expression dynamics by also answering the context question *when*. However, approaches that perform modeling of more than these two context questions are yet to be developed. As a first step toward this, in this thesis we propose a context-sensitive approach that can be used to answer all the context questions in a principled way, and we demonstrate this on the context questions *who, how* and *when*.

1.3 Potential Applications

There are many potential applications that can be developed from research into automated analysis of facial expressions. Below we list some of the most interesting.

- **Computer science.** Engineering automated systems with the capability to sense and understand facial expressions can enable technologies like ambient intelligence and ubiquitous computing. Affective interfaces, implicit-tagging-based multimedia retrieval,

multi-player games, and online services would all be facilitated or enhanced by such technology.

- **Robotics.** The development of new models and algorithms for automated analysis of facial expressions would enable the development of robots capable of understanding human behavior in both indoor and outdoor environments (e.g., the design of robot companions, robots as tourist guides, etc.).

- **Basic Science Research.** Facial behavior is an important variable for a large number of studies on human emotion, cognition, and communication [144]. Also, social signals, such as mimicry and rapport, are a focus of research in child developmental studies, in negotiations and intersubjectal influence, and studies of couple and family counseling. Systems for automated analysis of facial expressions would greatly speed up the current research as they could replace the lengthy and tedious manual analysis of the behavior under study.

- **Medicine.** Many disorders in neurology and psychiatry (e.g., autism spectrum disorder, schizophrenia, suicidal depression, Parkinson's disease) involve aberrations in the display and interpretation of facial behavior. Automated analysis of facial expressions could provide increased reliability, sensitivity, and precision needed to explore the relationship between facial behavior and mental disorder. This should be in the patient's natural environment rather than in a lab, where patients are usually "on guard". Not only would this lead to new insights and diagnostic methods, but could also be used to develop supportive technologies aimed at reducing severity of the disorders [15]. Also, remote monitoring and management of conditions such as pain and depression, remote assessment of drug effectiveness, remote counseling, etc., would be possible, leading to more advanced subjectal wellness technologies.

- **Digital Economy and Commercial Applications.** Automated measurement of consumers' preferences from their facial expressions in response to product adverts would have a profound impact on market research analysis, as this would open up the possibility of conducting mass-market research studies. It would also enable the next generation of in-vehicle supportive technology, automatic assessment of drivers' stress levels, detection of micro sleeps, etc. It would facilitate the development of truly intelligent tutoring systems by enabling automatic assessment of students' interest levels, comprehension, and enjoyment in online- and E-education.

The examples mentioned above are only a few of many potential applications of automated analysis of facial expressions.

1.4 Contributions

1.4.1 A Big Picture

The problem space described in Sec.1.2 is a general problem, and many specific problems of automated analysis of facial expressions can spring from it. In this thesis, we focus on two. The first is pose-invariant classification of facial expressions of the six basic emotions from static images. The second is analysis of dynamics of facial expressions from image sequences, in terms of classification of temporal segments of the six basic emotions and AUs, and also intensity estimation of pain and AUs.

Pose-invariant classification of facial expressions. Many real-world applications relate to spontaneous interactions (e.g., meeting summarization, gaming, monitoring of patients in hospitals, etc.), resulting in the facial-expression data that appear in multiple views/poses either because of the head motion or the camera position. Most of the existing methods deal with images, or image sequences, in which the subjects depicted are relatively still and in a nearly frontal view [221]. While those methods can deal with small in-plane head motion, their performance is expected to lessen significantly in the case of large out-of-plane head motion. To tackle this, we propose a system based on a set of novel machine learning algorithms that are specifically devised for performing pose-normalization of a set of facial features, extracted from static images of facial expressions in various poses. This is followed by classification of the pose-normalized facial expressions into the target emotion categories. To achieve successful pose-normalization, and consequently, pose-invariant facial expression classification, the proposed methods have to meet a number of challenges. We list the most important ones below.

- **How to deal with images of facial expressions with large out-of-plane head rotations?** The goal of pose-invariant facial expression classification to classify facial expressions from arbitrary poses. This can be achieved by decoupling variation due to rigid facial motions, caused by changes in head pose, and non-rigid facial motions, caused by facial expressions, and by classifying the latter into target emotion categories. However, decoupling pose and expression across a large number of poses is challenging mainly because these are non-linearly coupled in 2D images. Therefore, devising pose-normalization algorithms that are capable of preserving facial expression details in the

presence of out-of-plane pose variation, while being largely invariant to differences among observed subjects, illumination changes, etc., is of paramount importance for attaining pose-invariant facial expression classification.

- **How to generalize from a small number of examples of facial expressions in non-frontal poses?** Most of the existing datasets for facial expression analysis contain images of near frontal view facial expressions. Having examples of all facial expression categories in all possible non-frontal views is infeasible due to continuity of the pose space. For this reason, a few existing datasets provide facial images captured at a limited number of discrete poses, but these data are scarce. Therefore, the challenge here is how to devise methods that can generalize from a small amount of expression data. Furthermore, they should be able to generalize to poses that were not used for training (i.e., poses in between a discrete set of poses used to train the system). Moreover, they should be able to generalize to expression categories that were not seen in certain poses during training but during inference only.

- **How to handle noise and outliers in facial features?** Facial features used for classification of facial expressions depend largely on pre-processing of facial images. For instance, partial occlusions of a face, rapid head movements and illumination changes are just a few factors that can cause the resulting facial features to be contaminated by high levels of noise and/or outliers. This is especially true for data collected in less constrained environments, where subjects move their heads freely. Therefore, when this occurs, a method for pose-invariant facial expression classification needs robust pose-normalization to be able to accurately perform the expression classification.

The goal of the first part of this thesis is to solve the challenges mentioned above. For this, we introduce several models for pose normalization that attain successful decoupling of head pose and expression in the case of large out-of-plane head rotations. This is achieved by means of mapping functions learned at a discrete set of poses, but which can be used to perform pose-normalization of facial expressions with continuous pose changes. Pose-invariant facial expression classification is then accomplished by applying the standard classifiers to the pose-normalized facial expressions. The mapping functions that we propose for pose normalization are based on the Gaussian Process (GP) framework for regression and manifold learning. We use this non-parametric probabilistic framework as a basis for our models because it is particularly suited for learning highly non-linear mapping functions that can generalize from a small amount of training data. To achieve accurate and robust pose normalization,

we incorporate different types of spatial structure of facial expression data into the learning of the mapping functions. This is attained by means of the newly defined priors placed over the GP-based mapping functions. These priors encode dependencies between and within facial features from different poses. Specifically, because of modeling of dependencies between different poses, the learned mapping functions for pose normalization are able to generalize to facial expression categories that were not present in certain poses during training. Then, classification of the target facial expressions is accomplished from the pose-normalized features. Also, classification of facial expressions from underperfroming poses, i.e., poses in which it is more difficult to discriminate between target facial expression categories, is largely improved by modeling dependencies among corresponding facial expressions from different poses. On the other hand, by modeling dependencies in facial features within poses via deformable shape models, we obtain mapping functions that are largely robust to noise and outliers in the data. This all allows us to perform pose normalization of facial expressions from arbitrary poses, while being able to preserve subtle details in facial expressions, and therefore accomplish robust and accurate pose-invariant facial expression classification. In this way, we solved the challenges of the first problem addressed in this thesis.

Analysis of dynamics of facial expressions. As described in Sec.1.2.3, facial expression dynamics are important for successful interpretation of target facial expressions. These dynamics are typically described in terms of timing, speed and duration of the temporal segments (neutral, onset, apex, and offset) and intensity levels of facial expressions. Our goal in the second part of this thesis is to devise a system that can perform temporal segmentation and intensity estimation of facial expressions automatically. For this, we propose novel machine learning algorithms for classification of temporal segments and intensity levels of facial expressions of the six basic emotions, pain and AUs, in image sequences. In order to accomplish this reliably and accurately, there are a number of challenges that the proposed models have to meet. We list them below.

- **How to account for variability in facial expressions of different subjects?** Facial morphology and expressiveness levels can vary significantly among different subjects because contractions of facial muscles are usually combined with the subject's physical characteristics to produce changes in facial appearance. This is especially pronounced in spontaneously displayed facial expressions. Therefore, to be able to successfully generalize to novel subjects, models for automated analysis of temporal segments and intensity of facial expressions and AUs need be able to capture subtle variation in facial appearance caused by changes of temporal segments and intensity levels of facial expressions,

while being largely invariant to the subject-specific variation in facial appearance.

- **How to account for temporal structure in the facial expression data?** Different temporal segments, and also intensity levels, of facial expressions never occur in isolation but vary smoothly in time. Furthermore, temporal segments, and intensity levels, of facial expressions differ in their duration (e.g., the higher intensity levels occur less frequently than the lower levels). Moreover, temporal segments of emotion expression occur in a specific temporal order, i.e., the onset of emotion expression is followed by its apex or offset segment. Accounting for this temporal structure of facial expressions is important for the models to be able, for instance, to discriminate between onset and offset temporal segments of facial expressions. This cannot be accomplished from static images as these two segments are characterized by the same facial appearance.

- **How to account for structure of the facial expression labels?** Intensity levels of AUs and of facial expressions of pain are defined on a monotonically increasing ordinal scale. In other words, facial appearance labeled with the pain intensity level 5 is expected to be more similar to that labeled as the intensity level 6 than 10. The standard classification models do not account for this type of structure in the labels of expression intensity since they treat each intensity level independently. However, accounting for this ordinal structure by models is important because when the misclassification of the intensity levels occur, it is more likely to be between the neighboring intensity levels. For intensity estimation, and temporal segmentation, of facial expressions this is more acceptable than random misclassification (e.g., when intensity level 1 is confused with intensity level 6), as in models that ignore ordinal structure of the intensity labels.

- **How to account for context of facial expressions?** For successful interpretation of facial expressions, it is important to know the context in which the observed expression is displayed [149]. For instance, accounting for the observed subject as a context factor, is expected to result in a model that is robust to the subject differences mentioned above. Knowing where the subject is (e.g., indoors or outdoors) and what he/she is doing (e.g., watching a horror or comedy film) are the context factors that can also influence the meaning of the displayed facial expressions. Therefore, how to account for all six questions from the *W5+* context model [149] is the ultimate challenge in designing models for automated analysis of facial expression dynamics.

- **How to deal with the imbalanced facial expression data?** The lower intensity levels occur much more frequently than the higher intensity levels in spontaneously

displayed facial expressions, and, in particular, AUs, causing the distribution of the intensity levels to be highly skewed to lower intensities. This data imbalance poses a serious challenge for existing models when learning the minority classes (i.e., the higher intensities) as examples of these are scarce. Handling this data imbalance effectively is important for the models to be able to successfully generalize to the higher intensity levels of facial expressions and AUs.

The goal of the second part of this thesis is to solve the challenges mentioned above. For this, we propose models for temporal segmentation and intensity estimation of facial expressions that are based on the Conditional Random Field (CRF) framework for structured learning of image sequences. We base our approach on this framework because it provides a principled way for modeling of different types of data structures. Specifically, in our models we encode the temporal structure of the expression data via the edge potentials in the linear-chain CRF. To encode the ordinal structure in facial expression labels, we define the node potentials of the CRF using the modeling strategy of static ordinal regression models. In this way, we seamlessly integrate the ordinal-temporal structure of the data into our models. To deal with variability in subjects, we propose several strategies. Firstly, we define a non-parametric prior over the parameters of our ordinal CRF model. This prior constrains the parameter space via a low-dimensional manifold that preserves variation due to different emotion categories and their temporal segments, while being largely invariant to differences in subjects. We show benefits of this approach on the tasks of temporal segmentation of six basic emotions, and AUs. Secondly, instead of trying to suppress the subject differences, we allow the facial features of different subjects to influence the model parameters. We achieve this by modeling heteroscedasticity in the parameters via the node potentials of our CRF model. We show on the task of pain intensity estimation from spontaneous facial expressions that this model can better adapt to the subject differences, and thus attain improved intensity estimation, compared to its homoscedastic counterpart, and the traditional models for sequence learning. Finally, we generalize this model by also accounting for context-sensitive variability in the data. We achieve this by modeling the context questions *how*, *when* and *who* via the node and edge potentials of our context-sensitive CRF model. We also address learning of the intensity levels from imbalanced data by formulating a large-margin approach for sequence learning. Compared to existing models, we show that this approach achieves substantially better intensity estimation of pain and AUs from spontaneous facial expressions. By accounting for the effects mentioned above, we solved the challenges of the second problem addressed in this thesis.

1.4.2 List of Contributions

Below we describe in more detail the main contributions of this thesis.

- To address the first problem of this thesis, we propose a method for head-pose-invariant facial expression classification that is based on 2D geometric features, i.e., locations of 39 characteristic facial points. This method achieves pose invariance by warping the facial points from a set of (discrete) non-frontal poses to the frontal pose using the newly introduced Coupled Gaussian Process Regression (CGPR) model. This model attains accurate pose-normalization using a small amount of training data by modeling correlations between different poses. The classification of target expression categories is then performed in the frontal pose using the standard classifiers. In contrast to pose-wise classification of facial expressions, the proposed method can perform classification of expression categories that were not present in certain non-frontal poses during training due to its ability to accomplish pose normalization of novel expression categories. The proposed method is the first that can successfully handle expressive faces with continuous change in pose, ranging from $-45°$ to $+45°$ pan rotation and from $-30°$ to $+30°$ tilt rotation.

- The proposed CGPR model for pose normalization accounts for correlations *between* facial points in different poses, but not *within* poses. Consequently, the pose-normalized facial points are not warrantied to form a valid facial configuration, especially in the case of noise and outliers in data (e.g., due to errors in facial point localization). This, in turn, can adversely affect accuracy of facial expression classification from the pose-normalized facial points. To address this, we propose the Shape-conformed GP (SC-GP) regression model that performs structured learning of the warping functions by combining 2D deformable shape models with the GP regression framework. As a result, the output of the model is conformed to only feasible facial configurations, which makes it largely robust to high levels of noise and outliers in the facial points. Compared to existing GP regression models with structured output, the proposed SC-GP is the first that models geometry-based structure in the output by means of deformable shape models, resulting in its attaining more accurate pose normalization.

- The methods proposed above have two main limitations: they need a canonical pose to be chosen in advance, and their learning of mappings for pose normalization in the case of high-dimensional features (e.g., appearance-based features) is intractable because of the large number of outputs. To address this, we introduce the Discriminative Shared GP

Latent Variable Model (DS-GPLVM) that achieves pose-invariance by simultaneously aligning manifolds of facial expressions from multiple poses to a single low-dimensional shared manifold, where classification of target facial expressions is consequently performed. In this model, we use the notion of Shared GPs [58] to generalize discriminative GPLVMs [198, 227], proposed for a single observation space, to multiple observation spaces. In this way, we preserve on the shared manifold most discriminative information of facial expressions from different poses. Consequently, classification of facial expressions from underperfroming poses, i.e., poses in which it is more difficult to discriminate between target facial expression categories, is largely improved on the shared manifold. In contrast to our methods with explicit pose normalization, DS-GPLVM is directly optimized for classification of target expressions, resulting in its classification performance being less affected by inaccuracies in pose-normalization. As a result, this approach achieves better classification of target facial expressions. It also outperforms several state-of-the-art methods for multi-view learning on the target task.

- To address the second problem of this thesis, we introduce the Multi-output Conditional Ordinal Random Field (MCORF) model for analysis of facial expression dynamics by simultaneous classification and temporal segmentation of facial expressions of the six basic emotions. In this approach, we use the modeling strategy of Hidden Conditional Ordinal Random Fields (HCORF) [101] to define learning of a dynamic ordinal model, where temporal segments of emotion are treated as (latent) ordinal variables describing development of emotion expression. In this way, we seamlessly integrate the temporal and ordinal structure of our data into the model. Moreover, we constrain the parameter space of our model by placing the novel prior over the parameters. This prior is based on the graph Laplacian matrix, and is designed so as to force the parameters to be largely robust to inter- and intra- subject differences. This is the first approach in the field that accomplishes classification of facial expressions and their temporal segmentation simultaneously, resulting in its outperforming traditional sequence learning models on each task.

- The feature mappings in the MCORF model mentioned above are a linear approximation of the otherwise non-linear mapping functions due to the use of the graph Laplacian matrix in the prior. Although such mappings have been shown effective in the task of temporal segmentation of facial expressions of emotions, they are limited by their linear form. This constrains their ability to unravel more complex relationships that may exist between input features and temporal segments, as encoded by ordinal labels. This is

especially true in the case of temporal segments of AUs, the analysis of which often involves detection of subtle changes in local facial appearance, typically described by high dimensional feature vectors. In this case, parameter learning in MCORF becomes difficult, and can easily lead to overfitting. To address this, we introduce the Kernel CORF (KCORF) model that generalizes our linear MCORF model by using implicit feature mappings defined directly in a Reproducing Kernel Hilbert Space (RKHS) [170]. The resulting model can handle high-dimensional input features effectively, and also learn highly complex non-linear data structures by means of a pre-defined kernel function. For this, we propose the Composite Histogram Intersection kernel for automatic selection of facial regions that are most relevant for the target task. Besides being kernel-based and temporal, the KCORF model is the first that exploits ordinal relations (neutral $\prec$ onset, offset $\prec$ apex) between temporal segments of AUs in order to facilitate their classification. We show that this all helps to significantly improve temporal segmentation of AUs attained by traditional sequence learning models, as well as the state-of-the-art models for the target task.

- Spontaneously displayed facial expressions are typically characterized by large variation in facial expressiveness of different subjects, the effects of illumination and pose registration, etc. All this can lead to heteroscedasticity (i.e., the changing variance) in the data we aim to model. For this, the KCORF model is not flexible enough as it assumes constant variance in its feature functions. To address this, we propose the Heteroscedastic KCORF model that relaxes the assumption of having constant variance by allowing the inputs (e.g., appearance-based features) to differently influence its parameters. This, in turn, results in the model being able to easily adapt to the varying levels of facial expressiveness of different subjects. We show this on the problem of intensity estimation of spontaneously displayed facial expressions of pain. Compared to existing works addressing the target problem, this is the first method that performs dynamic modeling of pain intensity. Furthermore, this is the first method for facial expression analysis that accounts for effects of heteroscedasticity in data.

- Finally, we propose the Context-sensitive CORF (cs-CORF) model for intensity estimation of spontaneous AUs, and facial expressions of pain. In this model, we go beyond modeling of the spatio-temporal structure (i.e., ordinal and temporal dependencies) and heteroscedasticity in data, which we modeled in our dynamic models mentioned above. Specifically, we formulate our cs-CORF model in terms of the context questions (*who*, *when*, *what*, *where*, *why* and *how*) from the W5+ [149] context model, describing the

context in which target facial expressions occur. In this way, we obtain a framework that allows us to incorporate the spatio-temporal structure by explicitly answering the context questions *how* (the changes in facial expressions) and *when* (the timing of the facial expression intensity), but also the other effects such as the observed subject by answering the context question *who*. In contrast to our heteroscedastic KCORF model, cs-CORF accounts for subject variability by not only modeling the changing variance but also by allowing the subject-specific biases to influence the model parameters. It also provides a principled way of accounting for the other context questions (*what*, *where* and *why*), resulting in a model that can be used to fully exploit the context of facial expressions, and therefore facilitate their intensity estimation. We show that the proposed model considerably outperforms existing models for intensity estimation of AUs and facial expressions of pain. This is also the first attempt in the field to address fully context-sensitive modeling of facial expression data in a principled manner.

The contributions described above have resulted in the following journal and conference articles:

O. Rudovic, M. Pantic, I. Patras. Coupled Gaussian Processes for Pose-invariant Facial Expression classification. IEEE TPAMI 2013.

O. Rudovic, V. Pavlovic, M. Pantic, I. Patras. Context-sensitive Dynamic Ordinal Regression for Intensity Estimation of Facial Action Units. IEEE TPAMI (under review).

O. Rudovic, V. Pavlovic, M. Pantic. Context-sensitive Conditional Ordinal Random Fields for Facial Action Intensity Estimation. CVPR-W 2013.

S. Eleftheriadis, O. Rudovic, M. Pantic. Shared Gaussian Process Latent Variable Model for Multi-view Facial Expression classification. Advances in Visual Computing, ISVC 2013.

O. Rudovic, V. Pavlovic, M. Pantic. Automatic Pain Intensity Estimation using Heteroscedastic Conditional Ordinal Random Fields. Advances in Visual Computing, ISVC 2013.

O. Rudovic, V. Pavlovic, M. Pantic. Kernel Conditional Ordinal Random Fields for Temporal Segmentation of Facial Action Units. ECCV-W 2012.

O. Rudovic, V. Pavlovic, M. Pantic. Multi-output Laplacian Dynamic Ordinal Regression for Facial Expression classification and Intensity Estimation. CVPR 2012.

O. Rudovic, M. Pantic. Shape-constrained Gaussian Process Regression for Facial-point-based Head-pose Normalization. ICCV 2011.

O. Rudovic, I. Patras, M. Pantic. Coupled Gaussian Process Regression for pose-invariant facial expression classification. ECCV 2010.

O. Rudovic, I. Patras, M. Pantic. Regression-based multi-view facial expression classification. ICPR 2010.

O. Rudovic, I. Patras, M. Pantic. Facial Expression Invariant Head Pose Normalization using Gaussian Process Regression. CVPR-W 2010.

1.5 Thesis Outline

The thesis is structured as follows. We describe in Chapter 2 the general approach to automated analysis of facial expressions, and review the existing works. We pay particular attention to the existing machine learning models that have been proposed for facial expression analysis. The following Chapters are split into two parts.

In the first part, we address the problem of pose-invariant facial expression classification. We begin by Chapter 3, where we introduce the problem, and explain the GP framework that we use as a basis for our approach. Chapter 4 introduces our approach to pose-invariant facial expression classification that is based on the proposed Coupled GP regression model for head-pose normalization. Chapter 5 introduces the Shape-conformed GP regression model for head-pose normalization. In Chapter 6, we introduce the Discriminative Shared GP Latent Variable Model for pose-invariant facial expression classification. We conclude this part of the thesis in Chapter 7.

In the second part, we address the problem of modeling of facial expression dynamics, in terms of their temporal segments and intensity. We begin this part by Chapter 8, where we introduce the problem, and explain the CORF-based models that we use as a basis for our approach. In Chapter 9, we introduce the Multi-output CORF model for simultaneous classification and temporal segmentation of facial expressions of the six basic emotions. Chapter 10 introduces our Kernel CORF model for temporal segmentation of AUs. In Chapter 11, we introduce the Heteroscedastic Kernel CORF model for intensity estimation of facial expressions of pain. Chapter 12 introduces our Context-sensitive CORF model for intensity estimation of facial expressions of pain and AUs. We conclude this part of the thesis in Chapter 13. Finally, in Chapter 14 we conclude the thesis .

Chapter 2

Automated Analysis of Facial Expressions: The State of The Art

Contents

2.1 Facial Expression Analysis: Overview 29

2.2 Pre-processing and Feature Extraction 30

2.3 Machine Analysis of Facial Expressions 36

2.4 Relation to Our Work 48

2.1 Facial Expression Analysis: Overview

Although humans detect and analyze faces and facial expressions in a scene with little or no effort, development of an automated system that accomplishes this task is rather difficult [145]. The general approach to automated facial expression analysis typically consists of three steps (Fig. 1.1). Given an input image or image sequence, the first step consists of pre-processing of the target image(s). This is performed by (i) localizing a face, (ii) detecting a set of facial points, which are then used to perform (iii) face registration. Once the face is registered, the next step is facial feature extraction. Different geometric and/or appearance features can be used, and they are usually chosen depending on the target task. The final step is machine interpretation of facial expressions. To this end, different machine learning techniques have been proposed. In the sections that follow, we describe each of these steps in detail, and give an overview of related works. Since in this thesis we focus on machine learning techniques for intepretation of facial expressions, we place particular attention on relevant machine-learning models (Sec. 2.3). In Sec. 2.4, we relate these models to the techniques that we propose.

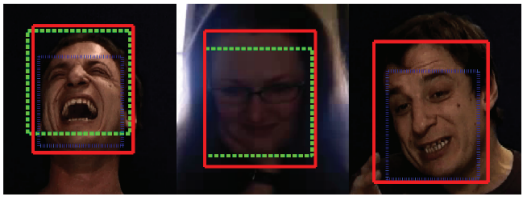


Figure 2.1: The results of the Viola&Jones face detector [204] are shown in green, [141] in red, and [228] in blue (bounding box definition is different for each method). The detector from [141] exhibits the most stable performance.

2.2 Pre-processing and Feature Extraction

2.2.1 Face Detection

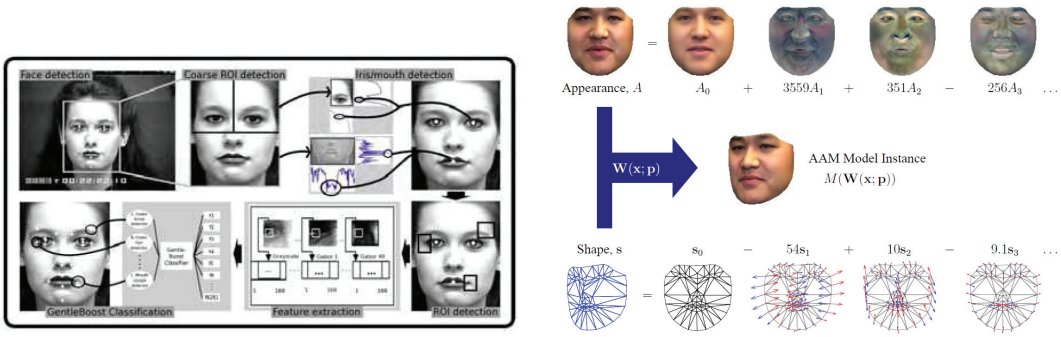
The first step of any face analysis method is to detect a face in an image. This is challenging mainly because of occlusions, variations in head pose and lighting conditions. Furthermore, the presence of non-rigid movements due to facial expression and a high degree of variability in facial size, color, and texture make this problem even more difficult [145]. For near-frontal faces, the Viola&Jones face detector [204] is the most commonly employed. This detector consists of a cascade of classifiers trained by AdaBoost employing Harr-like features. However, for dealing with spontaneous and/or multi-view facial expression data, multi-view face detection is required. For this, [141] recently proposed a method based on Cascade Deformable Part Models, which is capable of performing reliable multi-view face detection within the range of -90 to 90 yaw rotation. An extensive overview of other recent advances in face detection can be found in [222].

2.2.2 Facial Point Detection and Tracking

Once the face is detected, the next step is localization of a set of facial points. Although optional, this step is important as it facilitates face registration (Sec.2.2.3) as well as the extraction of geometric features such as contours of facial components, facial distances, etc. (Sec. 2.2.4). The methods for facial point detection and tracking can be classified as either texture-based methods (modeling local texture around a given facial point) or texture- and shape-based methods (regarding the constellation of all facial points as a shape, which is learned from a set of labeled faces, and trying to fit the shape to any unknown face) [145]. A typical texture-based method is that proposed in [206], while a typical texture- and shape-based method is Active Appearance Model (AAM) [132]. In the following, we briefly describe these and other related methods.

The facial point detector proposed in [206] detects locations of 20 characteristic facial points. Given the localized face in an image, this method models local image regions using Gabor wavelets and builds GentleBoost-based point detectors based on those regions. Specifically, a face image is divided into 20 regions of interest (ROIs), each corresponding to one facial point to be detected. Then, a combination of heuristics, based on the analysis of the vertical and horizontal histograms of the upper and the lower half of the face region image, is used to localize the points (see Fig. 2.2(a)). Further improvements of this method, which constrain the constellation of the facial points to form a valid face shape via graph-based modeling, have been proposed in [201, 131]. However, these methods are designed for near-frontal poses. More recent methods that can deal with pose variation have been proposed in [228] and [216]. These methods are based on a tree-based shape model and cascaded regression strategy, respectively. The method in [228] detects 68 facial points within range of poses $\pm 45^\circ$ yaw, and 39 facial points in profile images, while the method in [216] performs detection of 66 facial points within range of poses $\pm 45^\circ$ yaw, $\pm 90^\circ$ roll and, $\pm 30^\circ$ pitch. Both methods can reliably detect the facial points from expressive faces even in the presence of occlusions and illumination changes.

The methods based on AAMs [132, 4] are very popular popular for the facial point detection task. The facial points are used to define a facial shape, described by a 2D triangulated mesh (see Fig. 2.2(b)). Then, the appearance variation within each triangle in the mesh is modeled using PCA. The 66 facial points are detected by finding the parameter values that minimize the difference between the original image and the image reconstructed by both the AAM shape and appearance parameters. These parameters are typically found through an iterative gradient descent procedure. Implementation speed-up strategies, such as the inverse compositional method [132], allow faces to be matched very efficiently. Due to their representation of the facial shape by a triangulated mesh, they are commonly used for registration of the facial texture (see Sec.2.2.3). Furthermore, AAMs can successfully track the facial points throughout an image sequence, however, their performance largely depends on initialization of the facial mesh. This is usually attained by using some of the facial point detectors described above. Also, the appearance models of AAMs are learned within a narrow range of poses, so they usually do not generalize well in the case of large pose variation. The AAMs belong to the family of Parameterized Appearance Models (PAMs), which also includes many other models that have successfully been used for facial point detection/tracking. Some of the most popular ones are the Lucas-Kanade method [123], Eigentracking [24], Morphable models [56], and Constrained Local Models (CLM) [11]. For an extensive survey of these methods, see [51, 145].

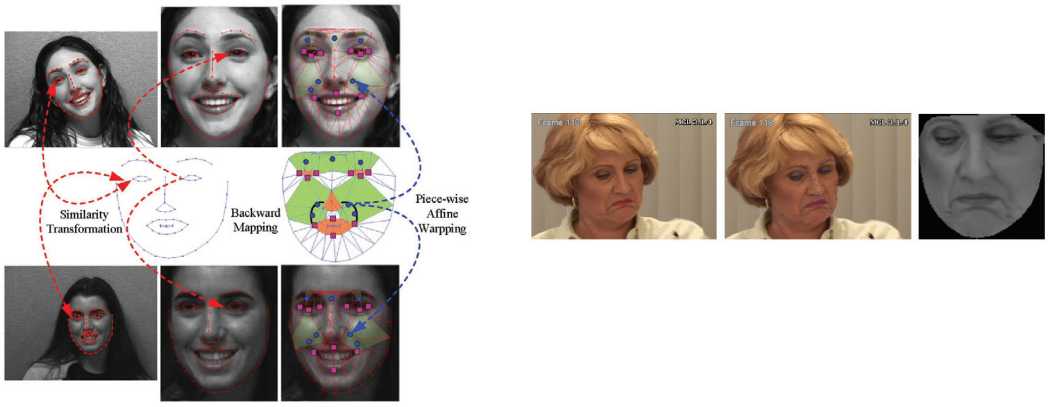


(a) Facial Point detector in [206] (b) AAM [132]

Figure 2.2: Two typical approaches to facial point detection. (a) A texture-based method that models local texture around a given facial point. (b) A texture- and shape-based model that performs fitting of the shape and appearance components through a gradient-descent search.

2.2.3 Face Registration

The goal of face registration is to eliminate rigid motions such as translation, difference in scale, and head rotations, but also to reduce subject variation such as the difference in facial configuration. In general, there are two main 2D approaches to face registration (3D-based methods are explained in Sec.2.3.2). The first performs coarse registration by detecting some inner facial components such as the eyes (e.g. [191, 17]). Then, the distance between the eyes is set equal in all faces, resulting in removal of translation and difference in scale. However, this simple approach is still sensitive to head-rotations and subject variation. This is to some extent addressed by the second approach which uses dense facial points around the eyes and other facial landmarks to register each face to an average (reference) face (e.g., [200, 125, 92]). The registration of the facial points alone can be achieved by learning either an affine transform based on 6 parameters or by applying Procrustes analysis to the facial points. Typically, only the facial points not affected by facial expressions (e.g., corners around the eyes and nose) are used to learn the transform, which is then applied to all the facial points. To register the texture, either global affine transform or piece-wise affine transform are learned from the detected facial points, and then used to warp the facial texture to the reference frame. The global transform is applied to the whole facial texture, while the piece-wise affine warp is applied to the corresponding facial parts. While the former may better preserve facial expression details, the second is better for reducing the subject differences. Fig. 2.3 describes the steps involved in applying the piece-wise affine warp to facial texture. Note that the registration approach presented here assumes small in-plane head rotations, so it is not suitable for dealing with out-of-plane head rotations. In Sec. 2.3.2, we review the registration techniques that are applicable to this case.



(a) Piece-wise affine warp (b) A warped face tracked with the AAM [132]

Figure 2.3: (a) The two-step process for registering the face to the reference face using piece-wise affine warp described in [51]. Purple squares represent detected points and blue dots represent the facial area around them. The dashed blue line shows the mapping between the point in the reference face and the corresponding points on the original image. By applying the piece-wise affine transformation and back-warping with Delaunay triangulation, used to extract appearance representations in areas that have not been explicitly tracked (e.g., nasolabial furrow), the facial texture can be registered better than by applying the global affine transform. (b) Original face (left), AAM tracking result (centre), result of the texture warping (a) to the mean shape (right). Images taken from the UNBC-McMaster shoulder pain database [125], tracking results from [92].

2.2.4 Feature Extraction

After the face registration has been successfully performed, the next step is feature extraction. There are two most commonly employed types of features: geometric and appearance features [145, 51].

Geometric features.. Geometric features encode information about the shape and locations of permanent facial features (e.g., eyes, brows, nose). Approaches that use only geometric features (or their derivatives) mostly rely on detecting sets of fiducial facial points [224, 148, 150, 200], or ASMs [82, 125, 92], or face component shape parametrization [41, 96]. For instance, [224] used locations of 34 fiducial points as a representation of facial expressions of the basic emotions. The works in [148, 150] used a set of 20 facial characteristic points, detected using the facial point detector proposed in [206] (Fig. 2.2(a)), to describe the facial expressions. Similarly, [200] used the same point locations to compute additional features based on distances and angles between them, as well as velocity of the point displacements in images. These features were used to describe temporal development of AUs. A shape model defined by 58 facial points was adopted in [82], where the analysis of the basic expression categories was performed on a manifold of the facial points. The locations of 68 vertices of an ASM, being part of the AAM [132] (Fig. 2.2(b)), were used in [125, 92] to describe the intensity

variation of AUs and facial expressions of pain. [41] adopted a model-based face tracker to track head motion and local deformation of facial features such as the eyebrows, eyelids, and mouth. These features were used for analysis of the basic emotions, which were described as activations of facial regions, and also the direction and velocity of their motion. Infrared eye (pupil) detection and tracking have also been adopted for facial motion representation in [96]. The authors used PCA to recover shape parameters from the eye and eyebrow regions, which were used for representation of AUs from the upper part of the face.

Appearance features. In contrast to geometric features, appearance features encode changes in skin texture such as wrinkles, bulges, and furrows [173]. The appearance features most commonly employed for facial expression analysis include Gabor wavelets [224, 17, 118, 133], Harr-like filters [102, 218], the learned statistical image filters such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Independent Component Analysis (ICA) [17, 120, 84], Discrete Cosine Transform (DCT) [92, 81], Local Binary Patterns (LBP) [136, 92, 173, 133], and gradient-based descriptors [187, 225, 84, 133]. Some of these features (e.g., Gabors and DCT) are better suited to represent global appearance, and are usually extracted from the whole face (holistic features). On the other hand, features such as LBPs or gradient-based descriptors are typically extracted from specific face regions (local features), for instance, around a set of facial points. In the following, we briefly describe the features mentioned above.

Gabor wavelets [114] are obtained by modulating a 2D sine wave with a Gaussian envelope. Representations based on the outputs of Gabor filters at multiple spatial scales, orientations, and locations have proven successful for facial image analysis as they can be sensitive to finer wave-like image structures such as those corresponding to wrinkles and bulges [173]. They are also robust to the misalignment of faces. However, computing Gabor wavelets is expensive as it involves convolution of face images with a set of Gabor filters, and it can also result in a high number of redundant features. Techniques such as PCA or LDA, can be applied to reduce number of the features. The Haar-like filters [211] respond to coarser facial features, and are also robust to alignment errors, while being computationally efficient. However, the Haar filters are not responsive to the fine texture details. The DCT [2] features encode texture variation in the frequency domain, and are usually extracted from the whole image. Although they are not very sensitive to alignment errors, to keep the number of features low, the high frequency coefficients are usually ignored, which can result in the loss of the fine texture details. The gradient-based descriptors such as Histograms of Oriented Gradients (HOG) [48] count the occurrences of gradient orientations in a localized portion of an image.

Similarly, the Scale Invariant Feature Transform (SIFT) [121] descriptors are computed from the gradient vector for each pixel in the neighborhood to build a normalized histogram of gradient directions. These features can capture subtle facial changes, and are particularly robust to illumination and scale variations. The LBP [139] descriptors have recently become popular for facial representation. They are constructed by forming, e.g., an 8-dimensional binary vector that encodes for each pixel in an image whether its intensity is higher from that of the neighboring pixels. A histogram is then computed, where each bin corresponds to one of the binary patterns. The LBPs are typically extracted from image blocks, where the image is divided, for instance, into 10×10 blocks. The LBPs are robust to illumination changes, computationally simple, and can represent the texture details well, even in the presence of spatial shifts [177]. However, compared to the gradient-based descriptors, they are less robust to image rotations.

The performance of the features described above can vary depending on the target task and the processing steps involved in their extraction. For instance, [133] compared Gabors, HOGs and LBPs in the task of AU intensity estimation from spontaneous facial expressions. The features were extracted around a set of facial points obtained using an AAM. They report higher performance by Gabors, compared to that of HOGs and LBPs, which performed similarly. [84] compared HOGs, LBPs and SIFT, extracted around facial points, for multi-view classification of facial expressions of the basic emotions, and in their experiments SIFT outperformed LBPs, which, in turn, outperformed HOGs. In another study on multi-view facial expression classification, [81] compared LBPs, DCTs and SIFT extracted around facial points, and showed that DCT outperform LBPs and SIFT, which performed comparably. Haar-like features were used for global representation of facial images for temporal segmentation of the basic expressions [102], and their intensity estimation [218]. However, no comparison with the other features was reported. LBPs extracted from image blocks, and DCT extracted from the whole image, were used in [92] for intensity estimation of AUs and facial expressions of pain. The LBPs alone outperformed DCTs, but the best results were obtained when the two features were combined. In summary, it is difficult to say which features are better for describing the facial appearance as their performance depends largely on the target task, pre-processing steps involved, datasets and machine-learning algorithms used (see Sec. 2.3). However, the gradient-based descriptors and LBPs should be extracted locally, while the features based on Gabors, Haar-like filters and DCT can be used to extract either holistic or local descriptors.

Geometric vs. Appearance features. Although the geometric features (i.e., locations of the facial points) are commonly used during extraction of local appearance features, when

compared separately, both geometric and appearance features have advantages and disadvantages. Specifically, geometric features are easily interpretable, and extremely computationally efficient, once the facial points have been detected/tracked. Also, the scale, in-plane rotation, and intra- and inter-subject variation, can more easily be removed during the registration step when working with geometric features. The works in [148, 150, 200, 127, 92] showed that analysis of facial expressions of the basic emotions and pain, as well as most of the AUs, can successfully be accomplished using geometric features. These works also showed on different tasks, including analysis of facial expression dynamics, that geometric features can be as effective or even better than appearance-based features. However, not all facial expressions can be described with geometric features (i.e., by using a sparse set of facial points). For instance, AU6 (cheek raise), AU11 (nasolabial furrow deepener), AU14 (mouth corner dimpler), and AU22 (lip funneler, as when pronouncing "flirt") do not produce uniquely identifiable face shape deformations. In particular, to discriminate between AU6 and AU7, capturing differences in appearance (e.g., furrows lateral to the eyes and cheek raising in AU6 but not AU7) is crucial. Similarly, AUs such as AU11 (nasolabial furrow deepener), AU14 (mouth corner dimpler), and AU28 (inward sucking of the lips) can not be detected from the movement of a sparse set of points alone but may be detected from changes in skin texture [51]. In this case, using the appearance-based features, or a combination of both, is the best choice. However, the appearance features may be more suitable for describing the AU intensity variation, but, as mentioned above, normalizing these features for factors such as the subject differences is challenging. The methods that we present in this thesis are based on both geometric and appearance features.

2.3 Machine Analysis of Facial Expressions

After the facial features are obtained using techniques explained in Sec.2.2, different machine-learning models can be designed for analysis of target facial expressions. In what follows, we first review existing approaches for facial expression classification from near frontal-view images, followed by the methods that deal with mutli-view facial images. We then review methods for analysis of facial expression dynamics, i.e., methods for temporal segmentation and intensity estimation of facial expressions. Lastly, we relate those methods to the methods proposed in this thesis.

2.3.1 Frontal-view Classification of Facial Expressions

Different methods have been proposed for classification of facial expressions in near frontal-view facial images. Depending on how these methods perform classification of facial expressions they can be divided into frame-based and sequence-based methods. The frame-based methods for classification of facial expressions of six-basic emotion categories typically employ static multi-class classifiers such as rule-based classifiers [150, 25], Neural Networks (NN) [143, 188], Support Vector Machine (SVM) [18, 177], Bayesian Networks (BN) [41], k Nearest Neighbours (kNN) [128], among others. The aim here is to classify an input image into one of six basic expression categories (sometimes the neutral facial expression is considered as an additional expression category). SVMs and its probabilistic counterpart, Relevance Vector Machine (RVM), have also been used for classification of facial expressions of pain [125, 66]. For instance, [125] addressed the problem of pain detection by applying SVMs either directly to the image features or by applying a two-step approach, where AUs were first detected using SVMs, the outputs of which were then fused using the Logistic Regression (LR) model [23]. Similarly, for static classification of AUs, where the goal is to assign to each frame a binary label indicating the presence of target AUs, the classifiers based on NN [19, 64], Ensemble Learning techniques (such as AdaBoost [217] and GentleBoost [76]), SVM [36, 17, 88], and kNN [54], are commonly employed. Most of these methods perform facial expression classification by directly applying a classifier to the features extracted from static images. Recently, [39] proposed a transductive learning method, named Selective Transfer Machine (STM), which is used to personalize the SVM classifier for AU detection by attenuating person-specific biases. This is accomplished by simultaneously learning the classifier and re-weighting the training samples that are most relevant to the test subject.

The common weakness of the frame-based classification methods is that they ignore dynamics of target facial expressions or AUs. Although some of the frame-based methods (e.g., [88]) use features extracted from several frames in order to encode dynamics of facial expressions, machine learning models for dynamic classification provide a more principled way for doing so. With a few exceptions, most of the dynamic approaches to classification of facial expressions are based on the variants of Dynamic Bayesian Networks (DBN). DBNs are graphical probabilistic models which encode dependencies among sets of random variables evolving in time, which are capable of representing probabilistic relationships among different facial expressions, and of modeling the dynamics in their development [173]. The most commonly employed models for sequence classification, Hidden Markov Models (HMM) [157] and Conditional Random Fields (CRF) [108], are generative and discriminative versions, respectively, of DBNs with the

linear-chain graph structure.

Various approaches based on HMMs have been proposed for dynamic classification of facial expressions [142, 140, 219, 117, 200, 178, 41, 99]. For example, [142, 140, 219, 117] trained independent HMMs using image-sequences of each emotion category, and then performed emotion categorization by comparing the observation likelihoods of the expression-specific HMMs. To better account for variability of subjects [142] modeled observation probability of hidden states in HMMs using mixtures of Gaussians. Furthermore, [219] proposed a two-stage HMM-based approach for classification of expressions of six basic emotions. Firstly, a bank of linear classifiers was applied at frame level, and the output was coalesced to produce a temporal signature for each observation. Secondly, discrete HMMs were used to learn the temporal signatures for each expression category. To model AUs, [117] used HMMs to model image-sequences of each AU independently. [200] used HMMs to perform temporal smoothing of the outputs of AU/emotion-specific SVMs, trained per-frame. The main criticism of these approaches is that they are not fully discriminative, as they perform modeling of facial expression categories (and AUs) independently of each other.

Some attempts toward joint modeling of different facial expression categories using HMMs have been made [178, 41, 99]. For instance [178] used geometric features (i.e., locations of facial points) and a non-parametric estimate of observation probability in the expression-specific HMMs. Discrimination between different expression categories is increased by means of class-membership priors, used to weight the observation probability in each HMM. [41] presented a two-level HMM classifier that performs the expression classification by automatically segmenting an arbitrary long video sequence into the segments corresponding to different emotion categories. This is accomplished by first modeling expression-specific HMMs. Then, transitions between expression categories (including the neutral expression) are modeled using a higher level HMM, which takes as input the predictions of the expression-specific HMMs. Simultaneous classification of AUs using HMMs was addressed by the hybrid HMM-NN model [99]. In this model, temporal development of each AU was first modeled using AU-specific HMMs. Subsequently, the outputs of different HMMs were combined into a NN, to account for dependencies between different AUs.

Discriminative models based on CRFs have also been proposed [202, 85, 34]. In [202], the authors trained one linear-chain CRF per AU, and each frame was associated to a node within the graph. The state of such a node is a binary variable indicating whether the AU is present or not in the current frame. The AU classification is performed per frame by thresholding the state probability for each frame in the test sequence. [85] used a generalization of the linear-

chain CRF model, a variant of Hidden Conditional Random Field (HCRF) [209], where a layer of hidden variables was used to describe the underlying dynamics of facial expressions. The training was performed using image sequences, but the classification of the facial expressions was done per frame by selecting the most likely class (i.e., emotion category) at each time instance. The authors showed that: (i) having the additional layer of hidden variables (set to 5 by a validation procedure) results in the model being more discriminative than the standard linear-chain CRF, and (ii) that modeling the temporal unfolding of the facial shapes is more important for discrimination between different facial expressions than their spatial variation (based on a comparison with the SVM classifier).

The HCRF mentioned above is by definition a sequence classifier, i.e., it infers the target class given the whole sequence. It can be thought of as a model that combines K CRFs, the labels of which are treated as hidden states. Then, a sequence of such states is connected to the top node carrying information about the expression category. The aim of this model, therefore, is to learn to discriminate between different classes at sequence level. [34] proposed a modification of this HCRF, named partially-observed HCRF. The appearance features based on the Gabor wavelets were extracted from the image sequences, and linked to facial expressions of the target emotion category via a combination of hidden and observed variables. The latter were labeled as sets of AU combinations (encoded using the binary information on AU activations in the images). In this way, classification of the emotion categories (sequence-based), and AU combinations (frame-based) was accomplished simultaneously. This method outperformed the standard HCRF, which does not use prior information about AU combinations, forming target expressions. Another variant of the HCRF model, named Hidden Conditional Ordinal Random Field (HCORF) [101], was proposed for sequence-based classification of facial expressions of six basic emotions. The key difference between HCORF and other HCRF-based models is that the former imposes ordinal constraints on its hidden states, implicitly correlating them with different temporal segments of emotion expression. In [101], the authors showed that HCORF exhibits better performance than HCRF with unconstrained hidden states, which, in turn, outperformed the expression-specific HMMs.

Another discriminative approach, named k-segment-SVM, for detecting the starting and ending frames of AUs was proposed in [182]. This model is based on the structured output SVM framework, where AU classification is defined as a problem of detecting temporal events in a time series of visual features from arbitrary long sequences. This is in contrast to most of the methods mentioned above, which deal with pre-segmented sequences. However, the latter methods can be used to segment sequences of different lengths during inference, while

for k-seg-SVM, the number of the AU activations in a sequence has to be specified in advance.

Variants of DBNs with more complex graphical structures have been proposed for dynamic classification of facial expressions [223, 192, 115]. In these models, different spatial relationships between random variables, corresponding to the emotion categories and/or AUs, are first represented by nodes of a static BN. Then, temporal unfolding of the time slices of BNs is modeled by the first-order HMMs. For instance, [192] proposed a DBN to model relationships between the co-occurring AUs. The authors showed that for some specific AUs that are difficult to be recognized, the classification can be significantly improved by modeling temporal as well as spatial dependencies between the AUs, compared to when the SVM classifier is used for each AU independently. To classify the facial expressions of six basic emotions, [223] proposed a hierarchical DBN with three layers. In this model, the hierarchy is modeled using static BNs, where the nodes describing the emotion categories (top layer) were related to the measurements (the bottom layer) via the nodes representing the subsets of the expression-related AUs (intermediate layer). More recently, [115] proposed a DBN with a similar hierarchical structure for simultaneous classification of facial expressions of emotions and AUs, where the relationships between the AUs were also modeled using an intermediate layer. The AU classification with this model outperformed that attained by using the SVM classifier, again mainly due to the modeling of the relationships between AUs. However this improvement did not translate to the classification of the six emotion categories, in the case of which the SVM-based classifier achieved a better performance. The reason for this is that the top nodes, corresponding to the emotion categories, were not directly connected to the measurements, but only via the AU nodes. Consequently, inaccuracies in AU classification adversely affected the classification of the facial expressions of emotions.

Finally, some authors attempted dynamic modeling of facial expressions on an expression manifold (e.g., [82, 176, 112]). These approaches first find either expression-specific or joint manifolds of facial expression data, and then learn classifiers directly in the manifold(s). The main assumption in these methods is that the temporal structure of an expression can be preserved on a manifold, i.e., that an expression sequence becomes a smooth path on the manifold, emanating from the center that corresponds to the neutral expression. [82] used a low dimensional Isomap embeddings to build a manifold of shape variation across different subjects and then used the I-condensation algorithm to simultaneously track and recognize different emotion categories within a dynamic probabilistic framework. A limitation of this approach is that the expression classification was performed on the subject-specific expression manifolds.

On the other hand, learning the subject-invariant expression manifolds is challenging, as most existing manifold techniques cannot successfully separate the expression- and identity-related variation. To this end, multi-linear decomposable generative models have been proposed. For example, [112] used these models to separate the subject's identity from facial expressions on a manifold. The expression embedding parameters were then used to perform the frame- and sequence-based expression classification. With the same aim, [176] proposed a supervised version of the Locality Preserving Projections (S-LPP)[78] method to simultaneously find a manifold of facial expressions of six basic emotions from the LBP-based image features. S-LPP allows us to encode various relationships between input images (using labels and/or features) that we would like to preserve on the manifold (e.g., temporal development of different expression categories), and discard the rest (e.g, the subjects' identity). S-LPP also provides an out-of-sample mapping, which allows us to embed the expressions of novel subjects into the manifold. In their experiments, the authors showed that S-LPP can successfully align data of different subjects on the manifold, while preserving differences caused by their facial expressions. To perform expression classification, a Bayesian temporal model was trained in the manifold. This approach performed better than when the features (without the dimension reduction) were used, showing the importance of reducing both the dimension of the features as well as the subject differences, for the expression classification. Note, however, that in these manifold-based methods, the learning of the manifold and the dynamic classifier is performed separately, which is suboptimal as they try to minimize different loss functions.

2.3.2 Pose-invariant Classification of Facial Expressions

Most of the methods for facial expression classification mentioned so far deal with images (or image sequences) in which the subjects depicted are relatively still and exhibit facial expressions in the nearly frontal view. The performance of these methods is expected to degrade in the case of large out-of-plane head rotations, as commonly encountered in real-world applications that relate to spontaneous human-to-human interactions (e.g., meeting summarization, political debates analysis, etc.). To address this, different approaches for multi-view/pose-invariant facial expression classification have been proposed. These approaches can be divided into face-shape-free models and face-shape-based models. The former are based on static classifiers trained using various appearance-based features extracted directly from facial images captured at multiple-views of facial expressions. In this way, they avoid facial point localization and/or fitting of face-shape models. On the other hand, the face-shape-based models rely on 3D/2D face models, used to perform decoupling of rigid and non-rigid facial motions (typically within a tracking framework) caused by variation in head-pose and

expressions, respectively.

Face-shape-free approaches. Recent release of 3D facial expression datasets (e.g, the BU3FEDB [208] dataset of static images of facial expressions of six basic emotions) has motivated research into multi-view/pose-invariant facial expression classification from static images. Typically, the 3D range data are used to render static expressive images at various views, corresponding to different positions of a camera. Also, the MultiPIE dataset [73] of (posed) facial expressions, recorded with multiple cameras capturing various views of a face, has recently become available for the same purpose. Thus, the availability of the *corresponding* images of expressive faces in multiple views, where the view/pose of an expressive face is known, has instigated the design of methods for multi-view facial expression classification. These methods can be divided into two groups: (i) the *pose-wise* methods, and (ii) the *pose-independent* methods.

In the *pose-wise* methods, facial expression classification is performed in each view independently from other views, by applying static classifiers. For instance, in [136], LBPs (and its variants) are used as input features, extracted from different views, to perform a two-step facial expression classification of facial expression of six basic emotions. The authors considered five different yaw angles (0-90 degrees) using the synthetic images generated from the BU3DFE dataset, and seven yaw angles (0-90 degrees) using facial images from the MultiPIE dataset. In the first step, the closest head-pose to the (discrete) training pose was selected by means of the SVM-based pose classifier. Once the pose was estimated, facial expression classification in that pose was performed using the pose-specific SVM facial expression classifiers. The following methods use synthetic images from the BU3DFE dataset to perform classification of facial expressions of six basic emotions. In [83], the authors investigated the performance of different static classifiers for facial expression classification at five yaw angles (0-90 degrees), using the ground-truth locations of 83 facial points as input features. The SVM classifier showed the best performance in the target task. [84] applied the two-step classification approach, where various appearance-based features (HoG, SIFT and LBP), extracted from synthetic facial images at five yaw angles (0-90 degrees), were used to train the pose-specific k-NN classifiers. The authors showed that the two-step pose-wise classification performs better than when a single classifier is used to discriminate between all possible combinations of views and expressions. They also showed that the classifier-fusion approach, where the outputs of the pose-specific k-NN classifiers trained with different appearance-based features were fused using a probabilistic framework, outperforms the pose-wise classifiers based on a single set of features.

In the second group of methods [225, 187], a single classifier is applied to data from multiple

poses/views. Specifically, [225] used variants of dense SIFT [122] features, extracted from multi-view expressive images in 35 views (sampled uniformly from the range -30 to +30 degrees pitch angles, and -45 to +45 yaw angles). To reduce dimensionality of the features, the authors proposed a reduction technique based on a discriminative Gaussian Mixture Model. The resulting features were then classified using a linear classifier. Likewise, [187] used the Generic Sparse Coding scheme [218] to learn a dictionary that sparsely encodes the SIFT features extracted from facial images in the same 35 views. Again, the linear classifier was used to perform expression classification. Nevertheless, the main limitation of the *pose-wise* methods is that they require a large amount of training data of each facial expression category in training poses, which may not be readily available. Furthermore, these classifiers are trained independently of each other, thus ignoring the data structure shared across the views. On the other hand, the single classifier used in the *pose-independent* methods attempts to simultaneously deal with variation caused by head-pose and facial expressions. When the number of poses and expressions is large, this can result in a too complex classifier, which can easily confuse expression- and pose- related variation.

While the works mentioned above focus on classification of facial expressions of six basic emotions, to our knowledge, only the works in [148] and [193] addressed the problem of multi-view AU classification. These works employed the rule-based and Gentle-boost classification, respectively, to detect AUs based on the displacement of facial points. However, they considered only a very limited number of views (i.e., the frontal- and profile- view).

Face-shape-based approaches. As mentioned above, the face-shape-based methods rely on 3D/2D face-shape models for decoupling of a head-pose and expression. The works in [16, 55, 210, 106] used 3D face-models to perform either pose-normalization of facial features, or to obtain the pose-invariant parameters of the model, which were then used for expression classification. Specifically, [16] first applied facial feature tracking, the output of which was used to estimate the head-pose in 3D from the 2D tracked facial points. Pose normalization was then performed by projecting facial texture onto a 3D face-model, which was then rotated to the frontal view. Facial expression of the pose-normalized facial texture was accomplished by applying the AU-specific SVM classifiers. Likewise, [210] applied an affine transformation to learn back-projection from locations of twenty-one facial points from a face image to a 3D virtual face-model. The rotated facial points were then modeled using the AU-specific HMMs, the outputs of which were used for classification of AUs, and the six basic emotions. With the same aim, [55] proposed an on-line Appearance-based facial tracker based on a 3D Candide model, which was used to find the facial-expression-related model parameters. Classification of

these parameters was accomplished by means of the expression-specific auto-regressive models. [106] used a rigid 3D face-shape-model to extract person-specific facial features, which were then used in a particle filter framework to simultaneously estimate head pose, and facial expressions of six basic emotions.

Decoupling of a head-pose and expression can also be attained by means of the methods that attempt to reconstruct the frontal from non-frontal faces using the 3D face-models (e.g., [26, 210, 230]). [26] proposed a morphable model to reconstruct a 3D face-shape from an input image, based on seven facial points localized in 2D input images. This model was used to generate virtual views (e.g., the frontal view) for face recognition. [230] proposed a normalized singular value decomposition (n-SVD) algorithm to separate head pose from facial expressions through parameterization of a 3D-Point Distribution Model (PDM), based on twenty-eight 2D facial points. However, these methods have not been used so far in the context of facial expression analysis.

Although the methods described above can be used for decoupling of a head pose and expression in 2D images, their performance is bounded by accuracy of head-pose estimation and/or tracking of facial points. This is especially true in the case of naturalistic data where large variation in pose, face morphology and expression is expected [72]. Moreover, because of ambiguities in estimating the 3D face shape from 2D images, some of the facial expression details can easily be lost. This, in turn, can adversely affect the expression classification. There are also methods that use facial geometry to recognize facial expressions from 3D images [184, 87, 208]. However, these methods require a high quality capture of the 3D facial texture, and thus, due to the extensive and complex hardware requirements, are not widely applicable. For facial expression analysis from 3D images, see [167].

A method for *pose-wise* facial expression classification based on a 2D face-shape model was proposed in [81]. In this approach, the authors applied pose-dependent 2D AAMs [45] to automatically localize facial landmarks from synthetic images from the BU3DFE dataset, at yaw angles from -90 to +90 degrees in steps of 15 degrees. The local appearance features (LBP, SIFT and DCT) were then extracted around the facial landmarks, and used to train *pose-wise* SVMs for classification of facial expressions of six basic emotions. When the correspondences between images of facial expression in different poses are known (e.g., as in the BU3DFE and MultiPIE datasets), 2D face models can also be used to perform decoupling of the pose and expression. For instance, the methods in [46, 12], used regression-based methods to perform face warping from the frontal to non-frontal poses. More specifically, [46] used linear regression for mapping the parameters of the frontal 2D PDM, being part of a 2D AAM in the frontal

pose, to the corresponding 2D PDMs in non-frontal poses. These mappings were then used to perform the warping of the facial texture. To obtain more accurate mappings, which is crucial for preserving facial-expression-specific details, [12] applied GP regression to learn mappings between the facial points of 2D PDMs in the frontal pose and non-frontal poses. Although these methods can be employed for decoupling of the pose and expression in presence of large out-of-plane head rotations, they have not been used in the context of multi-view facial expression classification so far.

2.3.3 Temporal Segmentation of Facial Expressions

While most works on facial expression analysis from image sequences have focused on classification of the target expressions or AUs, so far only a few approaches for explicit analysis of dynamics of facial expressions in terms of their temporal segments have been proposed [147, 148, 105, 200, 75, 102]. The important difference between the methods for dynamic classification of facial expressions described in Sec.2.3.1, and the methods for temporal segmentation of facial expressions is that the former model temporal dependences between the neighboring images, while the latter perform classification of different temporal segments of emotion expression. For example, the HMM-based models [178, 41] for facial expression classification, set the number of hidden states so that they correspond to the temporal segments (neutral/onset/apex/offset) of facial expressions. These methods, however, do not classify the sequence into different temporal segments. On the other hand, the methods proposed in [147] and [148] used static rule-based classifiers to classify different temporal segments of AUs based on the movements of the facial points in near frontal and profile view faces, respectively. They also used the temporal pattern (e.g., neutral→onset→apex) to detected AUs in the expression sequences. Similarly, but using dynamic models,[105, 200] encoded AU temporal segments. Specifically, [200] combined SVMs and HMMs in a Hybrid SVM-HMM model, based on the facial points, where the outputs of the temporal-segment-specific SVMs were passed through a sigmoid function to obtain a valid probability distribution for each segment. This distribution was used as the observation probability of hidden states in the AU-specific HMM models. The AU classification was then performed by detecting the temporal pattern, as described above. This approach outperformed the AU-specific HMMs trained without taking into account information about temporal segments of AUs. Similarly, [105] combined outputs of the temporal-segment-specific Gentle-boost classifiers in the HMM-based framework for AU detection.

Recently, [102] proposed a Conditional Ordinal Random Field (CORF) model for temporal

segmentation of the basic emotions, where the Haar-like features, extracted from expressive images, were used as input. This model is based on the linear-chain CRF model, where the node features are set using the modeling strategy of the standard ordinal regression models (e.g., [37]) in order to enforce ordering of the temporal segments (i.e., neutral<onset<apex). The proposed CORF model significantly outperformed the static classifiers for nominal data, such as SVM, and ordinal data, such as Support Vector Ordinal Regression (SVOR) [38], as well as the dynamic classifiers for nominal data such as HMM and CRF. This indicates the importance of imposing the ordinal constraints on the temporal segments, in addition to their temporal constraints, imposed by the transition model of the segments. In their subsequent work [101], the authors proposed the HCORF model for the expression classification (see Sec. 2.3.1). However, this model has not been used for temporal segmentation of facial expressions.

2.3.4 Intensity Estimation of Facial Expressions

Intensity of Emotion Expression. Because there is no established standard for how to code the intensity of facial expressions of emotions, the existing works on intensity estimation of facial expressions of the basic emotions resort to unsupervised approaches to measuring the expression intensity (e.g., [10, 173, 104, 113, 218]). The main idea behind these works is that the image variation due to facial expressions can be represented on expression manifolds, where image sequences are embedded as continuous curves. The distances from the origin of the manifold (corresponding to the embedding of neutral faces) are then used to determine intensity of facial expressions. For instance, [10] used an unsupervised Fuzzy-K-Means algorithm to perform clustering of the Gabor wavelet features, extracted from the expressive images, in a 2D eigenspace defined by the pairs of the features' principal components chosen so that the centroids of the clusters lie on a straight line. The cluster memberships are then mapped to three levels of intensity of a facial expression (e.g. less happy, moderately happy, and very happy). Similarly, [173] first applied a supervised LPP technique [175] to learn a manifold of six-basic expression categories. Subsequently, Fuzzy K-Means was used to cluster the embeddings of each expression category into three fuzzy clusters corresponding to low, moderate and high intensity of the target expression. [104] used a Potential Net model to extract the motion-flow-based features from the images of facial expressions that were used to estimate a 2D eigenspace of the expression intensity.

Continuous estimation of expression intensity was attempted in [113] and [218]. Specifically, [113] used isometric feature mapping (Isomap) to learn a 1D expression-specific-manifold. The distances on the manifold were used to define the expression intensity on a continuous scale.

The mapping of the input features to the expression intensity of three emotion categories (happiness, anger and sadness) was then modeled using either a Cascade NN or Support Vector Regression (SVR). The authors in [218] treated the intensity estimation as a ranking problem. They proposed the RankBoost alghorithm for learning the expression-specific ranking functions that assign different continuous values to each image. These values are assumed to correspond to the expression intensity, and are the result of the pair-wise comparison of the monotonically increasing changes in Haar-like features extracted from temporally neighboring images. The main criticism of all these works is that the expression intensity is obtained as a byproduct of the used learning method (and features), which makes comparison of the different methods difficult.

Intensity of Pain Expression. Recent release of the pain-intensity coded data [125] on a 16-level ordinal scale, based on the intensity of six AUs, has motivated research into automated estimation of pain intensity levels [77, 92, 162]. For example, [77] performed estimation of 4 intensity levels of pain, with the levels greater than 3 on the 16-level scale being grouped together because the distribution of the intensity levels is highly skewed toward the lower intensities. The authors obtained the image features by applying Log-Normal filters to the normalized facial appearance using AAMs, which were then used to train binary SVM classifiers for each pain intensity level. Since this approach uses the binary classifiers, it cannot deal with cases where the outputs of the multiple classifiers are positive. Instead of quantizing the intensity levels for the classification, [92] treated the pain intensity estimation as a regression problem. To this end, the authors proposed a feature-fusion approach based on the Relevance Vector Regression (RVR) [190] model, where the geometric features (facial points) and appearance features (DCT and LBP) are combined. The proposed approach achieved the best results when the combination of the appearance-based features (DCT and LBP) was used. Also, the authors performed a comparison between the pain intensity estimation directly from the image features, and that from the estimated intensities of AUs. They showed that the latter approach performs worse on the target task, which is in part due to the inaccuracies in the AU intensity estimation.

Intensity of AUs. Intensity estimation of AUs is a relatively recent problem within the field, so only a few works have addressed it so far. Based on the modeling approach, these can be divided into the classification-based [130, 133, 160, 52] and regression-based [168, 92, 86] methods. The classification-based methods use SVMs for AU intensity estimation. For example, [130] performed the intensity estimation of AU6 (cheek raiser) and AU12 (lip corner puller) from facial images of infants. The input features were obtained by concatenating the

geometric- and appearance-based features. Due to the excessive size of the feature vectors, the Spectral Regression (SR) [31] was applied to select the low-dimensional features for each AU. The intensity classification was then performed using AU-specific SVMs. The same approach was used in [133] but evaluated using recordings of subjects watching humorous videos. In [160, 52], the authors applied the Locality Learning Embedding (LLE) technique to geometric features extracted from the lower face, to find AU-specific 1D manifolds. Then, they attempted AU intensity estimation on the manifold using SVM. However, this performed poorly since there was a big overlap of the projected features of different intensity levels. As an alternative, the authors proposed the 3-level-intensity model, corresponding to the well separated clusters on the manifold.

The regression-based methods model the intensity of AUs on a continuous scale using either logistic regression [168], Relevance Vector Machine (RVM) regression [92], or Support Vector Regression (SVR) [86]. For instance, [168] used Logistic Regression for the AU intensity estimation, where the input features were selected by applying Ada-boost to the Gabor wavelet magnitudes of 2D luminance and 3D geometry extracted from the target images. The authors showed that the fusion of the 2D and 3D features improves the intensity estimation of most of the AUs addressed. [86] proposed a sparse representation of the facial appearance obtained by applying Non-negative Matrix Factorization (NMF) filters to gray-scale image patches extracted around facial points from the AU-coded images. The image patches were then processed by applying the personal-mean-texture normalization, and used as input to the SVR model for the AU intensity estimation.

Some researchers also tried to use outputs of AU classifiers to estimate the intensity of AUs. Specifically, [17] showed that for some AUs, the margins of AU-specific SVMs are highly correlated with the FACS-defined AU intensity levels. However, this unsupervised approach is unsuitable for the target task because the margins of the SVMs are adjusted for AU classification, and not the intensity estimation. Therefore, they do not necessarily incorporate all of the relevant intensity information [168].

2.4 Relation to Our Work

The machine learning methods for facial expression analysis that we propose are related to the methods reviewed in Sec.2.3. In what follows, we discuss similarities and differences of existing methods to the methods proposed in this thesis. We relate/contrast these methods in the context of the target problems that we address.

Pose-invariant classification of facial expressions. The methods that we propose for multi-view/pose-invariant classification of facial expressions of the six basic emotions from static images, first perform either explicit or implicit pose normalization, and then perform facial expression classification of the pose-normalized facial features. Specifically, the methods with explicit pose normalization, proposed in Chapters 4 and 5, operate *pose-wise* like the *pose-wise* methods for facial expression classification (Sec.2.3.2). However, while the latter learn the expression classifiers in each pose, we propose the GP-based regression models for mapping the features (facial points) from discrete non-frontal poses to the frontal pose, where the facial expression classification is performed subsequently. As we show in Chapter 5, our method can deal with scenarios in which examples of certain facial expression categories are not present in some non-frontal poses during training. In contrast to the *pose-wise* facial expression classifiers, this method can still perform classification of the missing expression categories at the pose in question during inference. This is because the proposed regression model, used in the pose-normalization step of our method, can generalize to unseen facial expression categories. Also, we show that for accurate pose-normalization, and thus facial expression classification, only a small amount of training data is needed to learn the mapping functions for pose normalization. On the other hand, for learning the *pose-wise* facial expression classifiers, much more data is needed in each non-frontal pose in order to achieve comparable performance.

The method for multi-view facial expression classification with implicit pose normalization that we propose in Chapter 6, achieves pose normalization on a discriminative manifold shared among multiple views of a facial expression. Instead of learning independent classifiers, as in the *pose-wise* classification methods, we learn a single classifier in the low-dimensional shared manifold. As we show in our experiments, classification accuracy in underperfoming views can better be attained in the shared manifold than when the *pose-wise* or *pose-independent* classifiers from Sec.2.3.2 are used. This is because the *pose-wise* classifiers ignore the structure shared among different views (some of which are more discriminative for the target task), while the *pose-independent* classifier is not informed about data correspondences, resulting in a more complex and less robust facial expression classifier.

Classification and temporal segmentation of facial expressions. In the methods that we propose for this task, we employ modeling strategy of the CORF and HCORF models (see Sec.8.3), to incorporate ordinal and temporal constraints into the models. These constraints account for spatio-temporal structure in sequences of facial expressions. We also exploit geometric constraints that have been successfully used in the manifold-based models for facial expression classification (Sec.2.3.1). As shown in [176], having these geometric constraints is

important as they aim at reducing the subject differences in input features. In our method for simultaneous classification and temporal segmentation of facial expressions of six basic emotions that we propose in Chapter 9, we model all three types of the constraints. This is in contrast to the HCORF and manifold-based models, as well as the other models proposed for facial expression classification (i.e., HMMs and Bayes nets). We show that by modeling the geometric constraints also, we obtain a more robust and effective model for the target task. Furthermore, while the existing manifold-based models for facial expression classification attempt to learn the dynamic models (e.g., HMMs) independently from the expression manifold, we do so simultaneously. As a consequence, temporal (and ordinal) information is seamlessly integrated into the structure of the expression manifold, facilitating fitting of the HCORF parameters, which become largely invariant to the subject differences. Moreover, existing methods focus on either classification or temporal segmentation of facial expressions of the basic emotions, but none attains both. By contrast, our method achieves this simultaneously.

Temporal segmentation of AUs. The method for temporal segmentation of AUs that we propose in Chapter 10 is based on a kernel extension of the CORF model, proposed for temporal segmentation of the basic emotions (Sec.2.3.3). While the standard CORF model in [102] employs linear mappings in its definition of the ordinal feature functions, our kernel method enjoys all the benefits of the kernel machines [170]. Therefore, this approach can deal with high dimensional feature vectors (e.g., the appearance-based features), in which case learning of the linear CORF model becomes intractable. On the other hand, the hybrid methods for temporal segmentation of expressions, such as hybrid HMM-SVM (Sec.2.3.3), learn the discriminative features for the target task by means of the SVMs independently from dynamic features (i.e., the transition model of HMMs). Our method does this in a principled way as the feature functions are learned jointly with the other parameters of the model. This is another advantage of using the discriminative instead of the generative modeling approach in our method (for discriminative vs. generative modeling, see, e.g., [108]).

Intensity estimation of pain. In the method for intensity estimation of facial expressions of pain, introduced in Chapter 10, we extend the kernel method mentioned above by accounting for heteroscedasticity in the ordinal node potentials. This allows the model to more easily adapt to the varying expressiveness levels of different subjects. The effects of neither heteroscedasticity nor temporal modeling have been addressed before in the domain of facial expression intensity estimation. As we show in Chapter 10, by accounting for both of these effects, we can improve the performance in the target task of the CORF-based models mentioned above, and significantly outperform the commonly employed static classifiers such as SVM. Note that

the methods for intensity estimation of facial expressions of the basic emotions from Sec. 2.3.4 have some similarities to the ordinal model used in the node potentials of our model. While the former estimate expression intensity in an unsupervised manner by learning 1D manifolds, which are assumed to correspond to the target intensity, the latter learn 1D ordinal manifolds, but in a supervised manner since the intensity labels are used. Since we address the supervised learning of facial expression intensity, our approach seems a natural choice.

Intensity estimation of AUs. Finally, in the context-sensitive method for intensity estimation of facial expressions of AUs, introduced in Chapter 12, we further generalize the CORF model by explicitly modeling the subject variability, in addition to the heteroscedastic and temporal modeling of the target task. A distinct feature of this approach is that it is able to personalize the CORF model by allowing subject-specific biases to influence its parameters. As we show in Chapter 12, this model achieves substantially better performance on the target task compared to the CORF-based models, which attempt to attenuate the person biases in the pre-processing step by normalizing the sequence features w.r.t. the first frame in the sequence. Compared to the STM-SVM model for AU detection (Sec.2.3.1), which is also sensitive to the subject variability, our method is dynamic and inductive, while the STM-SVM is static and transductive. Thus, our method can easily generalize to novel subjects, as well as exploit a temporal pattern in sequences of facial expression intensity levels. This has not been addressed before as the existing methods for intensity estimation of facial expressions of pain, and AUs, are static. Moreover, in contrast to these methods, our method is designed to deal with a skewed distribution of the intensity levels, which is commonly encountered in the data of spontaneous facial expressions.

Part I

Pose-invariant Facial Expression Analysis from Static Images

Chapter 3

Gaussian Processes for Pose-invariant Facial Expression Analysis

Contents

3.1 Introduction 55

3.2 Why GPs? 56

3.3 Gaussian Processes 57

3.4 Summary of Proposed Methods 60

3.1 Introduction

Many real-world applications relate to spontaneous interactions (e.g., meeting summarization, gaming, monitoring of patients in hospitals, etc.), resulting in the facial-expression data that appear in multiple views/poses either because of head motion or the camera position. Most of the existing methods deal with images, or image sequences, in which the subjects depicted are relatively still and in a nearly frontal view [221]. While those methods can deal with small in-plane head motion, their performance is expected to drop significantly in the case of large out-of-plane head motion. Thus, achieving accurate decoupling of rigid head motions, from non-rigid facial motions caused by facial expressions, so that the latter can be analyzed independently, is the crux of any method for pose-invariant facial expression analysis. Nonetheless, this remains a significant research challenge, mainly due to the large variation in appearance of facial expressions in different poses and the difficulty in decoupling these two sources of variation. To address this, we propose several methods for head-pose normalization in the case of large out-of-plane head motion or different views, which are based on the Gaussian

Process (GP) [159] framework. These methods are the integral part of our approach for pose-invariant facial expression analysis. In the following sections, we first provide motivation for using GPs in our approach. We then describe the general GP framework and summarize the methods proposed in this part of the thesis.

3.2 Why GPs?

The main goal in our approach to head-pose normalization is to learn high-dimensional mappings between the corresponding facial features in multiple poses, which we pose either as the supervised learning problem (i.e., regression), or as the unsupervised learning problem (i.e., dimensionality reduction). In the former case, we aim to map facial features from non-frontal poses to the corresponding features in the frontal pose (the explicit pose-normalization), while in the latter we aim to find a low-dimensional manifold where the facial features from multiple poses are well aligned (the implicit pose-normalization). In what follows, we outline the key strengths of GPs that make them particularly suitable for the target tasks.

- Due to their non-parametric nature, GPs allow us to specify various types of covariance functions that can capture complex data structures. This is important as we need to be able to preserve facial-expression-specific details during pose-normalization.

- GPs provide a well calibrated uncertainty in their predictions. This uncertainty can be used to design gating functions for combining predictions of different mapping functions learned with GPs. We use this uncertainty to combine the mapping functions learned for pose-normalization from different poses (Chapter 4).

- Prior knowledge can easily be incorporated into the GP models. We use this property of GPs to incorporate two types of priors: (1) the face shape prior, defined using statistical face-shape models. This results in a model with structured output, that we use for head-pose normalization (Chapter 5). (2) The discriminative prior, defined using the notion of graph Laplacian matrix that encodes the class information. We place this prior over a manifold in which we align facial expressions from multiple views, and perform their classification (Chapter 6).

- Different types of information can be combined using the concept of Shared GPs [58]. We use this for alignment of facial expressions from multiple views (Chapter 6).

- GPs can generalize well from a small amount of training data [159]. This is important when learning the mapping functions for pose-normalization as the training data of facial expressions in multiple poses are scarce.

In the following Section, we describe the framework of GPs.

3.3 Gaussian Processes

A GP is a generalization of the Gaussian probability distribution. Whereas a probability distribution describes random variables which are scalars or vectors (for multivariate distributions), a stochastic process governs the properties of functions [159]. Formally:

Definition *A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution.*

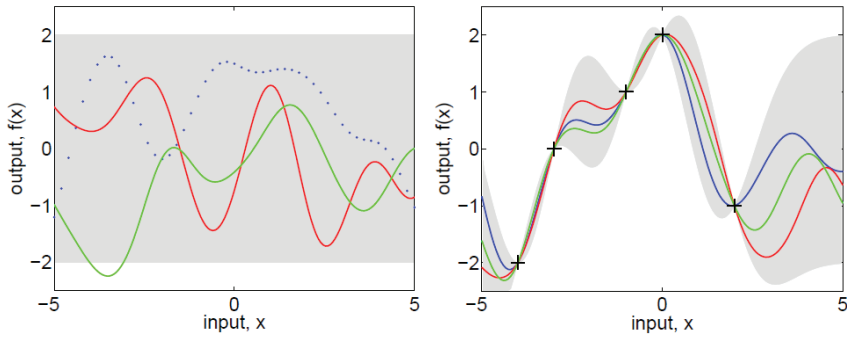


Figure 3.1: The figure on the left shows three functions drawn at random from a GP prior; the dots indicate values of y actually generated. The figure on the right shows three random functions drawn from the posterior, i.e., the prior conditioned on the five noise free observations x indicated. In both plots the shaded area represents the pointwise mean plus and minus two times the standard deviation for each input value (corresponding to the 95% confidence region), for the prior and posterior respectively. (Taken from [159], Fig.2.2.)

For some observed data $\mathbf{x}$, a GP is completely specified by its mean and covariance function. The mean function $m(\mathbf{x})$ and the covariance function $K(\mathbf{x})$ of a real process $f(\mathbf{x})$ are defined as

$$\begin{aligned} m(\mathbf{x}) &= \mathrm{E}[f(\mathbf{x})] \\ K(\mathbf{x}) &= \mathrm{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}^T) - m(\mathbf{x}^T))]. \end{aligned} \tag{3.1}$$

The GP can then be written as $f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), K(\mathbf{x}))$. Assuming a zero mean GP, the posterior distribution of the function values $f(x_*)$ corresponding to the newly observed x_*, is

a multi-variate Gaussian distribution (see [159], Sec.2.2, for details) specified as

$$f(\mathbf{x}_*)|\mathbf{x}_*, \mathbf{x}, f(\mathbf{x}) \sim \mathcal{N}(K(\mathbf{x}_*, \mathbf{x})K(\mathbf{x}, \mathbf{x})f(\mathbf{x}), K(\mathbf{x}_*, \mathbf{x}_*) - K(\mathbf{x}_*, \mathbf{x})K(\mathbf{x}, \mathbf{x})^{-1}K(\mathbf{x}_*, \mathbf{x})). \quad (3.2)$$

Fig. 3.1 shows the functions f sampled from the GP prior, and the posterior distribution given by (3.2), with the covariance function being the Radial Basis Function (RBF). Note that the posterior distribution restricts the function space defined by the prior to only those functions which agree with the observed data. This property of GPs forms the basis for the GP regression model that we describe in the following section.

3.3.1 GP Regression

The goal of GP regression is to learn input-output mappings from empirical data (the training dataset) with continuous outputs. In the context of the target task, i.e., for head-pose normalization, we wish to learn the mapping functions f that can be used to predict the facial features in the frontal view, given the corresponding facial features extracted from non-frontal-view facial images.

The mapping function f is obtained as follows. Given a training set of N input vectors $x = [x_1, \ldots, x_i, \ldots x_N]$ along with the target values $y = [y_1, \ldots, y_i, \ldots y_N]$, the GP regression first defines a smooth mapping $y_i = f(x_i) + \epsilon_i$, where $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ is Gaussian noise with zero mean and variance σ^2. The optimal functional form for f is then found by placing a zero mean GP prior over the functions: $f \sim \mathcal{GP}(0, K)$, where K denotes $N \times N$ covariance matrix, the elements of which are computed by applying a kernel function to data pairs $(x_i, x_j)_{i,j=1..N}$. During inference of a new input x_*, the mean and variance of the predictive distribution in (3.2) are used to obtain y_*, and its uncertainty $V(x_*)$, respectively, as

$$\begin{aligned} y_* &= k_*^T(K + \sigma^2 I)^{-1} y \\ V(x_*) &= k(x_*, x_*) - k_*^T(K + \sigma^2 I)^{-1} k_* \end{aligned} \quad (3.3)$$

Here, $k_* = k(x, x_*)$ is the kernel function computed between the training and query inputs as

$$k(x_i, x_j) = \theta_1^2 \exp(-\frac{||x_i - x_j||^2}{2\theta_2^2}) + \theta_3 x_i x_j^T + \theta_4, \quad (3.4)$$

where (θ_1, θ_2) are the parameters of the RBF, θ_3 corresponds to a parametric model that is a linear function of the input variables, and θ_4 accounts for the model bias. This kernel function is commonly employed, due to its ability to handle both linear and non-linear data structures [23]. The key result of GP regression is that the prediction of y_* is obtained by marginalizing over all possible choices of f, with more weight being put on those functions that agree with

the query input x_*. Because there is not a fixed mapping function f, the model is less prone to overfitting.

The marginal likelihood of the training data (or evidence) is computed as

$$p(y|x) = \int p(y|f,x)p(f|x)df, \tag{3.5}$$

where the likelihood $p(y|f,x)$ is a factorized Gaussian $y|f \sim \mathcal{N}(f, \sigma^2\mathrm{I})$ and $p(f|x)$ is the GP prior. The adaptation of the model parameters $\Omega = \{\sigma, \{\theta_i\}_{i=1}^4\}$ can then be accomplished by maximizing the log marginal likelihood:

$$\log p(y|x) = -\frac{1}{2}y^T(K+\sigma^2\mathrm{I})^{-1}y - \frac{1}{2}\log\left|K+\sigma^2\mathrm{I}\right| - \frac{N}{2}\log 2\pi, \tag{3.6}$$

w.r.t. Ω using conjugate gradient algorithm [159]. The $|\cdot|$ represents the determinant of the matrix.

Note that although here we focused on the application of GPs to regression, GPs can also be used for supervised learning with discrete outputs (i.e., classification), and unsupervised learning (i.e., data dimensionality reduction) (see [159] for details). In the case of GP regression, the computation of predictions is straightforward, as the relevant integrals are Gaussian and can be computed analytically. By contrast, in the classification case, the likelihood is non-Gaussian, which makes the integral in (3.5) analytically intractable. Although the approximation methods are available, in the case of more than two classes, the learning of the GP classifier becomes computationally intense. Because of this, standard classifiers such as SVM are a more practical solution. On the other hand, dimensionality reduction with GPs is typically accomplished by treating the inputs x in (3.5) as low-dimensional latent variables that are estimated together with other parameters of the model (Ω). We detail this in Chapter 6. Lastly, the GP regression model that we introduced here is designed for a single output ($dim(y) = 1$). To deal with more outputs simultaneously, different multi-output GP regression models have been proposed. We mention the most commonly used models for multiple outputs in Chapter 4.

3.3.2 Relation to traditional regression models

GP regression is closely related to traditional regression models such as Linear Regression (LR) [23], Support Vector Regression (SVR) [171], and Relevance Vector Regression (RVR) [190]. All these models can be seen as special cases of GP regression [159]. For instance, LR is a parametric model, the parameters of which are estimated using the sum-of-least squares criterion [23]. In its standard formulation, N training examples is needed to achieve a stable

solution for the design matrix of the model, where $N >> D$, and D is the dimension of the input vectors. This can pose a serious limitation when working with high-dimensional data, and is usually ameliorated by using the rank regularization (e.g., Ridge Regression [23]). This is equivalent to MAP estimation with a GP where the regularization in the Reproducing Kernel Hilbert Space (RKHS) is performed. A commonly employed approach based on the regularization framework is SVR, a sparse kernel technique that selects a small number of training examples, known as support vectors, and uses them for making predictions. While LR provides a closed-form solution for its parameters, the parameters of SVR (error/margin trade-off (C), insensitivity (ϵ) and kernel (θ) parameters) are estimated using a validation procedure, which can be time consuming. Compared to GP regression, where all training examples are used during inference, this deterministic model achieves sparsity by using a different model for the distribution of residuals (when $\epsilon = 0$, it becomes Laplace distribution). RVR is a sparse formulation of GP regression. The basic idea behind RVR is that training examples that are not significantly contributing to explaining the data should be removed. The examples selected by the model are called relevance vectors. Empirically it is often observed that the number of relevance vectors is smaller than the number of support vectors for the same problem [190]. In contrast to LR and SVR, RVR and GP regression are defined in the Bayesian framework, so they provide uncertainty in their predictions. However, RVR has a degenerate type of covariance matrix (see [159]), which can result in the wrong estimation of the uncertainty for query inputs that are not very close to training data.

3.4 Summary of Proposed Methods

Below we summarize the proposed methods.

- In Chapter 4, we propose a method for head-pose-invariant facial expression classification that is based on 2D geometric features, i.e., the locations of 39 characteristic facial points, extracted from images depicting facial expression of different subjects with various head-poses. To achieve head-pose invariance, we propose the Coupled Scaled Gaussian Process Regression (CSGPR) model for head-pose normalization by warping the facial points from (discrete) non-frontal poses to the frontal pose. This model is based on a mixture of GP regression models designed for multiple outputs, where the outputs represent coordinates of the facial points in the frontal pose. Each component in the mixture model learns the mappings between a pair of a non-frontal pose and the frontal pose, and then the predictions from different poses are combined in the frontal pose using the

proposed gating function. This gating function is designed by exploiting the uncertainty of the predictions from different poses. For simultaneous mapping of the facial points, the multi-output is attained by defining a GP covariance matrix that is differently scaled for each output. The resulting model achieves accurate pose-normalization of the facial points, performing similarly or better than several state-of-the-art GP regression models for multiple outputs. It also largely outperforms traditional regression-based approaches to head-pose normalization, 2D and 3D Point Distribution Models (PDMs), and Active Appearance Models (AAMs), especially in the case of imbalanced training data. This performance translates into the classification performance in the frontal pose, where the classification is performed by applying the SVM classifier to the pose-normalized facial points. In contrast to the *pose-wise* facial expression classifiers, commonly used for the target task, our approach performs similarly or better while using much less training data in non-frontal poses, and it can also perform classification of facial expression categories that were not available in certain non-frontal poses during training.

- In Chapter 5, we propose the Shape-conformed GP (SC-GP) regression model for facial-point-based head-pose normalization. This model achieves structured learning of the mapping functions for warping the facial points from non-frontal poses to the frontal pose, which makes it largely robust to high levels of noise and outliers in the facial points (e.g., due to errors in the facial point localization). Specifically, we model the structure in both inputs and outputs of the model by means of a 2D deformable shape model, which we incorporate into the learning of the GP regression model. The structure in the inputs is incorporated via the GP covariance function, while the structure in the output is incorporated via a minimization process that enforces only anatomically feasible facial configurations to arise from the model. Compared to the standard multi-output GP regression models, which attempt to learn structure in the output without taking into account domain knowledge, in SC-GP we use the face-shape models to accomplish this, resulting in a model that is more effective for pose-normalization. We show that SC-GP achieves accurate head-pose normalization in the presence of noise and outliers in the expression data from various poses, outperforming the standard GP regression, a 3D-PDM and AAM, on the target task. Also, in the presence of high-levels of noise and outliers, SC-GP outperforms Twin GP [28], the state-of-the-art regression model for multiple outputs.

- Finally, in Chapter 6, we propose the Discriminative Shared GP Latent Variable Model (DS-GPLVM) for classification of facial expressions from multiple views. Instead of

warping the facial features to a pre-defined view (i.e, frontal), this model achieves pose-invariance by aligning low-dimensional manifolds of facial expressions from multiple views. In DS-GPLVM, we use the framework of Shared GPs to generalize the GP discriminative latent variable models designed for a single observation space. Specifically, in DS-GPLVM we perform discriminative learning of the expression manifold shared across different views by placing a discriminative prior, defined using the notion of the graph Laplacian matrix, on the manifold. Classification of the facial expressions from multiple views is then performed in the shared manifold using a single classifier (e.g., k-NN). The advantage of this approach is that it is not affected by errors in pose-normalization. Moreover, the adverse effect of the high-dimensional noise is reduced as the classification of the target expressions is performed in the shared-manifold of multiple views, instead of directly in the observation space of the canonical view. Also, while performance of the pose-normalization methods mentioned above depends on the choice of the canonical view (e.g., we used frontal), DS-GPLVM automatically selects the shared-space optimal for classification of facial expressions from multiple views. We show that DS-GPLVM outperforms state-of-the-art methods for multi-view facial expression classification, including our approach based on the pose-normalization mentioned above, and several state-of-the-art methods for multi-view learning.

In Chapters 4-6, we describe in detail each of the proposed contributions. The discussion and directions for future work are given in Chapter 7.

Chapter 4

Coupled Gaussian Processes for Pose-invariant Facial Expression Classification

Contents

4.1 Introduction 63

4.2 Methodology 64

4.3 Experiments 70

4.4 Conclusions 80

4.1 Introduction

In this Chapter, we propose a probabilistic approach to pose-invariant facial expression classification that is based on 2D geometric features, i.e., the locations of 39 characteristic facial points, extracted from an expressive face in an arbitrary pose. The proposed approach consists of three steps: (1) pose estimation, (2) pose normalization, and (3) facial expression classification in the frontal pose. To perform the pose estimation, we first project the input facial points onto a low-dimensional manifold obtained by multi-class Linear Discriminant Analysis (LDA) [23]. We then use a Gaussian Mixture Model (GMM) [23], trained on the manifold data, to estimate the likelihood of the input being in a certain pose. In the second step, we perform pose normalization. This is achieved by learning mappings between a discrete set of non-frontal poses and the frontal pose by means of the proposed Coupled Scaled Gaussian Process Regression (CSGPR) model. To enable accurate pose normalization for continuous

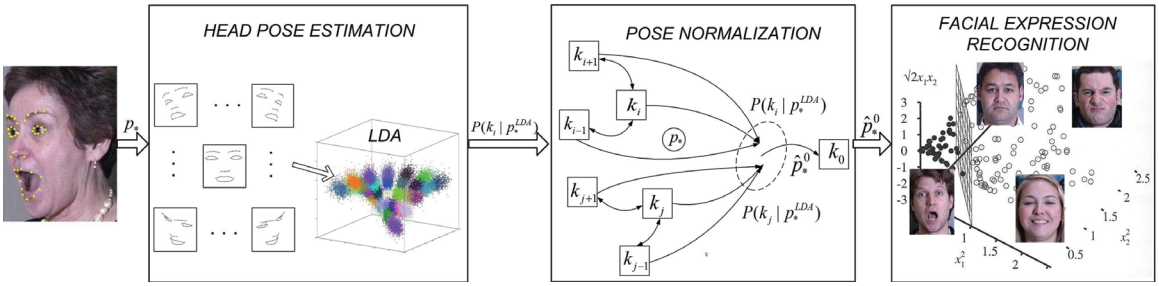


Figure 4.1: The overview of the proposed approach. p_* are the 2D locations of the facial points extracted from the input face image, $P(k_i|p_*^{LDA})$ is the likelihood of p_* being in pose k_i, where k_0 is the frontal pose. The bidirectional lines in the pose normalization step connect the coupled poses, while the directed lines connect the poses for which the CSGPR models are learned. $\hat{p}_*^0$ is the prediction in the frontal pose for the query point p_*, obtained as a combination of predictions by the CSGPR models. The gating function is derived from the pose likelihoods $P(k_i|p_*^{LDA})$. Facial expression classification is performed by applying a multi-class Support Vector Machine (SVM) classifier in the frontal pose to $\hat{p}_*^0$.

change in pose (i.e., for poses that do not belong to a discrete set of poses), we devise a gating function that combines the point predictions made by the CSGPR models trained in discrete poses, and which is based on the pose estimation attained in the first step of the proposed approach. In the final step, we perform facial expression classification by applying a multi-class Support Vector Machine (SVM) classifier to the pose-normalized facial points. The outline of the proposed approach is given in Fig.4.1.

4.2 Methodology

In this section, we describe the proposed approach to pose-invariant facial expression classification. We first explain the proposed model for pose normalization, followed by pose-invariant facial expression classification. In the following, the space of possible poses is divided into P (evenly distributed) discrete poses. The locations of the facial points extracted from an expressive face in pose k, where $k = 0, ..., P-1$, are stored in a vector $p^k \in \mathbf{R}^{2d}$. The training dataset is then denoted by $D = \{D^0, ..., D^k, ..., D^{P-1}\}$, where $D^k = \{p_1^k, ..., p_N^k\}$ is comprised of N training examples in non-frontal pose $k \neq 0$, and N can vary between non-frontal poses. Lastly, $\{D^k, D^0\}_{k=1}^{P-1}$ are the pairs of the corresponding training data in pose k and the frontal pose.

4.2.1 Coupled Scaled Gaussian Process Regression (CSGPR)

In this Section, we describe the proposed CSGPR model for pose normalization. For this, we first learn a set of base functions $\{f^{(1)}(\cdot), ..., f^{(k)}(\cdot), ..., f^{(P-1)}(\cdot)\}$ for mapping the facial points

from non-frontal poses to the corresponding points in the frontal pose. An ensemble of the coupled functions $\{f_C^{(1)}(\cdot), ..., f_C^{(k)}(\cdot), ..., f_C^{(P-1)}(\cdot)\}$ is then inferred by modeling the correlations between the base functions. In this way, we perform knowledge transfer across the poses, which is important for improving the performance of the base mappings in situations where examples of facial expressions of certain emotions are not present in all poses during training.

Scaled Gaussian Process Regression (SGPR). To learn the base mapping functions $f^{(k)}(\cdot)$, we propose Scaled GPR. This model is based on the Scaled Gaussian Process Latent Variable Model (SGPLVM) proposed in [71], originally proposed for data dimensionality reduction. In contrast to standard GPR [159], which is designed for a single output (i.e., each coordinate of each facial point), SGPR achieves simultaneous prediction of multiple outputs (i.e., all coordinates of all facial points). Formally, given a set of N_k training pairs of the facial points in non-frontal pose k and the corresponding points in the frontal pose, $\{D^k, D^0\}$, where each element p_i^k and p_i^0 $(i = 1, ..., N_k)$ in D^k and D^0 is a $2d$-dimensional vector (d is the number of the facial points), the goal is to learn the mapping:

$$p^0 = f^{(k)}(p^k) + \mathbf{1}_{1\times(2d)}\varepsilon_i, \tag{4.1}$$

where $\varepsilon_i \sim \mathcal{N}(0, \sigma_n^2)$ are the error terms with a Gaussian distribution. For standard GPR, the likelihood of the single output m given the inputs is given by

$$P(\{p_{i,m}^0\}|\{p_{i,m}^k\}, \theta) = \frac{1}{\sqrt{(2\pi)^{N_k}\,|K_m|}} \exp(-\frac{1}{2} D_m^k K_m^{-1} (D_m^k)^T), \tag{4.2}$$

where K_m is the data covariance matrix with entries $k_m(x_i, x_j)$. Instead of learning a separate K for each output, SGPR defines a scaling parameter w_m for output dimension m, which results in having the kernel function $k(x_i, x_j)/w_m^2$, where $k(x_i, x_j)$ is shared between the multiple outputs. The joint likelihood of the SGPR model is then obtained as

$$P(\{p_i^0\}|\{p_i^k\}, \theta, W) = \prod_m \frac{w_m^2}{\sqrt{(2\pi)^{N_k}\,|K|}} \exp(-\frac{1}{2} w_m^2 D_m^k K^{-1} (D_m^k)^T), \tag{4.3}$$

where $\theta = \{\sigma_s, S, \sigma_l, \sigma_b, \sigma_n\}$ are the kernel parameters and $W = \{w_1, ..., w_{2d}\}$, and the entries of the covariance matrix K are given by:

$$k(p_i^k, p_j^k) = \sigma_s^2 \exp(-\frac{1}{2}(p_i^k - p_j^k)^T S^{-1} (p_i^k - p_j^k)) + \sigma_l p_i^k p_j^k + \sigma_b, \;\; i, j = 1, ..., N_k \tag{4.4}$$

where σ_s^2 is the variance and $S = diag(s_1^2, .., s_{2d}^2)$ are the length-scales of each input dimension (i.e., each coordinate of each landmark point) of the RBF kernel, σ_l is the scale of the linear kernel, and σ_b is the model bias. We adopt this composite kernel because it can handle both

linear and non-linear data structures [159]. The model parameters θ and W are found by minimizing the negative log-likelihood:

$$-\log P(D^k, \theta, W|D^0) = d \ln |K| + \frac{1}{2} \sum_{m=1}^{2d} w_m^2 (D_m^0)^T (K + \sigma_n^2 I)^{-1} D_m^0 + const. \tag{4.5}$$

This likelihood function is first minimized w.r.t. θ using Scaled Conjugate Gradient algorithm [159]. The scale parameters W are then computed in a closed-form as $w_m = \sqrt{2d/\left((D_m^0)^T K^{-1} D_m^0\right)}$. These two steps are repeated until convergence of the likelihood function. During inference in SGPR, the mean $f^{(k)}(p_*^k)$ and variance $V^{(k)}(p_*^k)$ of the predictive distribution for the query point p_*^k are obtained as

$$f^{(k)}(p_*^k) = k_*^T (K + \sigma_n^2 I)^{-1} D^0, \tag{4.6}$$

$$V^{(k)}(p_*^k) = (k(p_*^k, p_*^k) - k_*^T (K + \sigma_n^2 I)^{-1} k_*)\, diag(W)^{-2}, \tag{4.7}$$

where $k_* = k(D^k, p_*^k)$. The mean $f^{(k)}(p_*^k)$ provides point predictions of the facial points in the frontal pose, and $V^{(k)}(p_*^k)$ their uncertainty .

Learning the coupled functions So far, we have used SGPR to learn a set of the base functions that map the facial points from non-frontal poses to the frontal pose. However, since these functions are learned separately, there is no sharing of knowledge between the poses. This sharing may be valuable when different training data are available across the poses. We accomplish this sharing by learning a set of coupled functions, which take into account the correlations between the base mappings. This is illustrated by an example of coupling a function $f^{(k_2)}(\cdot)$, the base function for pose k_2, to a function $f^{(k_1)}(\cdot)$, the base function for pose k_1. We adopt a parametric approach to learning the correlations between the mapping functions, which are induced through a prior distribution defined as:

$$P(f^{(k_1)}, f^{(k_2)}|k_1) \propto \exp(-\frac{1}{2\sigma_{(k_1,k_2)}^2} ||f^{(k_1)}(p_*^{k_1}) - f^{(k_2)}(p_*^{k_1})||^2), \tag{4.8}$$

where $\sigma_{(k_1,k_2)}^2$ is the variance of coupling that is estimated from training data D^{k_1} and D^{k_2}. Intuitively, it measures the similarity of the predictions made by the function $f^{(k_2)}(\cdot)$ and predictions made by the function $f^{(k_1)}(\cdot)$, when they are evaluated on the training data in pose k_1. It can also be seen as an independent noise component in the predictions obtained by $f^{(k_2)}(\cdot)$, which is learned using training data in pose k_2, when evaluated on training data in pose k_1. Because we assume that this noise is Gaussian and independent of the noise already modeled in $f^{(k_2)}(\cdot)$, these two sources of randomness simply add [159]. Consequently, by

including the coupling variance $\sigma^2_{(k_1,k_2)}$ into the predictive distribution of $f^{(k_2)}(\cdot)$, we obtain the following expressions for the mean and variance of the predictive distribution of the coupled function $f^{(k_1,k_2)}(\cdot)$ as:

$$f^{(k_1,k_2)}(p_*^{k_1}) = k_{k_2*}^T(K_{k_2} + (\sigma^2_{nk_2} + \sigma^2_{(k_1,k_2)})I)^{-1}D^0, \tag{4.9}$$

$$V^{(k_1,k_2)}(p_*^{k_1}) = (k_{k_2}(p_*^{k_1}, p_*^{k_1}) - k_{k_2*}^T(K_{k_2} + (\sigma^2_{nk_2} + \sigma^2_{(k_1,k_2)})I)^{-1}k_{k_2*})\, diag(W_{k_2})^{-2}, \tag{4.10}$$

where the subindex k_2 refers to the model parameters of the base function for pose k_2, and $k_{k_2*} = k(D^{k_2}, p_*^{k_1})$. Here, the sharing of knowledge between poses k_1 and k_2 is achieved through the coupled function $f^{(k_1,k_2)}(\cdot)$, which uses training data from pose k_2 when making predictions from pose k_1. Note also from Eq.(4.10) that the less $f^{k_2}(\cdot)$ is coupled to $f^{k_1}(\cdot)$, which is measured by the coupling variance $\sigma^2_{(k_1,k_2)}$, the higher uncertainty in the outputs obtained by the coupled function $f^{(k_1,k_2)}$. In other words, if the functions are perfectly coupled (i.e., $\sigma^2_{(k_1,k_2)} \to 0$), then $f^{(k_1,k_2)}(\cdot) \to f^{(k_2)}(\cdot)$. Conversely, if they are very different (i.e., $\sigma^2_{k_1,k_2} \to \infty$), then $f^{(k_1,k_2)}(\cdot)$ converges to a GP prior with the zero mean and constant variance. Lastly, the variance in Eq.(4.10) is guaranteed to be positive definite since we add a positive term (i.e., the coupling variance) to its diagonal.

CSGPR: Model. Here we explain how the outputs of the base and coupled functions are combined, resulting in the Coupled SGPR model for pose normalization. Let us consider the base function $f^{(k_2)}(\cdot)$ and the coupled function $f^{(k_1,k_2)}(\cdot)$. During inference, these two functions give their own predictions of the facial points in the frontal pose. We now combine them in order to obtain a single prediction. A straightforward approach is to apply either density-based (DB) weighting, using pose estimation explained in Alg.4.2, or the variance-based (VB) weighting, where the weights are set to inversely proportional values of the uncertainty in GP predictions. In this work, we employ the Covariance Intersection (CI) [91] rule for combining predictions, which is the optimal fusion rule when correlation between the prediction errors of two estimators are unknown [195]. For predictions obtained by the base and coupled functions, this fusion rule yields the mean and the variance of the CSGPR model, given by

$$f_C^{(k_1)}(p_*) = V_C^{k_1}(p_*)(\omega V^{(k_1)}(p_*)^{-1} f^{(k_1)}(p_*) + (1-\omega)V^{(k_1,k_2)}(p_*)^{-1} f^{(k_1,k_2)}(p_*)), \tag{4.11}$$

$$V_C^{(k_1)}(p_*)^{-1} = \omega V^{(k_1)}(p_*)^{-1} + (1-\omega)V^{(k_1,k_2)}(p_*)^{-1}. \tag{4.12}$$

The optimal $\omega \in [0,1]$ is found during inference by minimizing the trace of $V_C^{(k_1)}(p_*)$, used as the uncertainty criterion, w.r.t. ω (see [91] for details).

Algorithm 4.1 Learning and inference with CSGPR

OFFLINE: Learn the base SGPR models and the coupling variances.
1. Learn $P-1$ base SGPR models $\{f^{(1)}(\cdot),..,f^{(P-1)}(\cdot)\}$ for target pairs of poses.
2. Perform coupling of the base SGPR models learned in Step 1.

for k_1=1 to P-1 **do**
  for k_2=1 to P-1 & $k_1 \neq k_2$ **do**
    predict $\sigma_{(k_1,k_2)}$
    if $C_{eff}^{(k_1,k_2)}/C_{eff}^{(k_1,k_1)} > C_{min}$ **then** $\sigma_C^{k_1} = [\sigma_C^{k_1}, \sigma_{(k_1,k_2)}]$ **end if**
  end for
  store $\sigma_C^{k_1}$
end for

ONLINE: Infer the facial points in the frontal pose from the facial points $p_*^{k_1}$ in pose k_1.
B_{k_1}: number of the base functions coupled to $f^{(k_1)}$.
1. Evaluate the base function for pose k_1:
  $Pr(0) = \{f^{(k_1)}(p_*^{k_1}), V^{(k_1)}(p_*^{k_1})\}$
2. Combine the functions coupled to pose k_1.
  for i=1 to B_{k_1} **do**
    $\sigma_{(k_1,i)} = \sigma_C^{k_1}(i)$, $Pr(i^-) = \{f^{(k_1,i)}(p_*^{k_1}), V^{(k_1,i)}(p_*^{k_1})\}$
    $Pr(i^+) = CI(Pr(i-1), Pr(i^-))$
  end for
  $\{f_C^{(k_1)}(p_*^{k_1}), V_C^{(k_1)}(p_*^{k_1})\} = Pr(i)$

The pruning scheme. Drawing inference with all the coupled functions, i.e., $\frac{P(P-1)}{2}$ coupled functions, is computationally intensive. Also, not all the coupled functions contribute to improving the predictions obtained by the base functions. To address this, we propose a pruning criterion, which is based on the number of effective degrees of freedom (EDoF)[194] of a GP, to select the coupled functions that will be used during inference. EDoF of a GP measures how many degrees of freedom are used by the given data, and can be a good indicator of the variability in the training dataset (in terms of facial expressions). Hence, in our pruning scheme, we keep only the coupled functions that have a similar or larger number of EDoF than that of the base functions they are coupled to. In this way, we significantly reduce the computational load of the CSGPR model during inference. We define the number of EDoF of a coupled function $f^{(k_1,k_2)}(\cdot)$ as:

$$C_{eff}^{(k_1,k_2)} = \sum_{i=1}^{N_{k_2}} \frac{\lambda_{k_2}^i}{\lambda_{k_2}^i + \sigma_{nk_2}^2 + \sigma_{(k_1,k_2)}^2} \tag{4.13}$$

where $\lambda_{k_2}^i$ are the eigenvalues of the covariance matrix K_{k_2}, and N_{k_2} is the number of training data used to learn the base function $f^{k_2}(\cdot)$. The number of EDoF is approximately equal to

the number of eigenvalues of the kernel matrix K_{k_2} that are greater than the noise variance. Thus, if $\sigma^2_{(k_1,k_2)}$ is high, then $C_{eff}^{(k_1,k_2)} \rightarrow 0$, and the predictions made by the coupled function $f^{(k_1,k_2)}(\cdot)$ can be ignored. The coupled functions used for inference are selected based on the ratio: $C_{eff}^{(k_1,k_2)}/C_{eff}^{(k_1)}$, where its minimum value (C_{min}) is set using a cross-validation procedure, as explained in the experiments. The number of EDoF of the base functions is computed using Eq.4.13 without the coupling variance term. Note that the coupling variance could also be used as a criterion for pruning. However, the proposed measure is more general since it also tells us how much we can 'rely' on the coupled function in the presence of novel data (e.g., novel facial expression categories) - something that is not encoded by the coupling variance. Finally, learning and inference of CSGPR are summarized in Alg.4.1.

4.2.2 CSGPR for pose-invariant Facial Expression classification

Below we explain each of the three steps used in our approach for pose-invariant facial expression classification.

Head Pose Estimation. We devise a simple but efficient method for pose estimation that is based on multi-class Linear Discriminant Analysis (LDA) [23]. To this end, we first align the facial points in each discrete pose by using generalized Procrustes analysis to remove the effects of scaling and translation. Then, we learn a low-dimensional manifold of poses by means of multi-class LDA [23] using the aligned training data in each discrete pose and the corresponding pose labels. This manifold encodes pose variations while ignoring other sources of variations such as facial expressions and inter-subject variation. We denote the vector of the input facial points projected onto this manifold as p_{lda}. The distribution of such vectors having the same pose is modeled using a single Gaussian. Consequently, the likelihood of a test input p^*_{lda} being in pose k is then given by $P(p^*_{lda}|k) = \mathcal{N}(p^*_{lda}|\mu_k, \Sigma_k)$, where μ_k and Σ_k are mean and covariance of the training data in pose k after being projected onto the pose manifold. By applying Bayes' rule, we obtain $P(k|p^{lda}) \propto P(p^{lda}|k)P(k)$, where a uniform prior over the poses is used.

Head Pose Normalization. The pose normalization is attained by mapping the locations of the facial points from an arbitrary pose to the locations of the corresponding facial points in the frontal pose. To do this, we apply the proposed CSGPR model, which is explained in detail in Section 4.2.1.

Facial Expression Classification in the Frontal Pose. The final step in the proposed approach is the facial expression classification applied to the pose-normalized facial points.

For this, different classification methods can be employed (e.g., see [221, 200]). Here we use the multi-class SVM classifier with the one-vs-all approach [171]. The SVM classifier takes the locations of the facial points in the frontal pose $\hat{p}_*^0$ as the input, and constructs a separating hyperplane that maximizes the margin between the positive and negative training examples for each class.

Algorithm Summary. Given a query point p_*, we first compute the likelihood $P(k|p_*^{lda})$ of it being in non-frontal pose k, where $k = 1, ..., P-1$. The facial points $\hat{p}_*^0$ in the frontal pose are then obtained as a weighted combination of the predictions of the coupled functions $f_C^{(k)}(p_*)$ from non-frontal poses. Note that before $f_C^{(k)}(\cdot)$ is applied to the points p_*, these points are first registered to a reference face in the pose k, which is a standard pre-processing step. This registration is performed by applying an affine transformation learned using three reference points: the nasal spine point and the inner corners of the eyes. These are chosen since they are stable facial points, and are not affected by facial expressions [200]. The facial expression classification is then performed by applying the multi-class SVM classifier to the pose-normalized facial points. Finally, note that the inference time for p_* can be significantly reduced by only considering the most likely poses, i.e., $P(k|p_*^{lda}) > P_{min}$, where P_{min} is chosen so that only the predictions from poses being in the vicinity of the test input p_* are considered. Alg.4.2 summarizes the proposed approach.

Algorithm 4.2 pose-invariant Facial Expression classification

Input: Positions of facial landmarks in an unknown pose (p_*).
Output: Facial expression label (l).
1. Apply the pose estimation (Sec. 4.2.2) to obtain $P(k|p_*^{lda})$, $k = 0, ..., P-1$.
2. Register p_* to poses $k \in \mathcal{K}$ which satisfy $P(k|p_*^{lda}) > P_{min}$, and predict the locations of the facial landmarks in the frontal pose (Sec. 4.2.1) as

$$\hat{p}_*^0 = \frac{1}{\sum_{k\in\mathcal{K}} P(k|p_*^{lda})} \sum_{k\in\mathcal{K}} P(k|p_*^{lda}) f_C^{(k)}(p_*^k).$$

3. Perform the facial expression classification in the frontal pose to obtain l.

4.3 Experiments

4.3.1 Datasets and Experimental Procedure

We evaluate the proposed method using facial images from three publicly available datasets: the BU-3D Facial Expression (BU3DFE) [208], CMU Pose, Illumination and Expression (MultiPie) [73], and Semaine [135] datasets. We also use the Multi-pose Facial Expression (MPFE) dataset that we recorded in our lab. Table 4.1 summarizes the properties of each dataset, and Figs. (4.2),(4.7) show the sample images. The BU3DFE and MPFE datasets contain images

Figure 4.2: Example images from the **BU3DFE** dataset (*top*) and the **MultiPIE** dataset (*bottom*) with synthetic and manually localized facial points, respectively.

.

Table 4.1: Summary of the data used from the datasets employed. We use ∞ to denote the facial expression levels and poses that change continuously.

Dataset	Subjects	Expressions		Poses			Type	
		number	levels	tilt	pan	total	posed	real
BU3DFE	100	7	2	(-30°,+30°)	(-45°,+45°)	247	$\surd$	$\times$
MulitPIE	50	4	1	0°	(-45°,0°)	4	$\surd$	$\surd$
Semaine	10	2	∞	(-30°,+30°)	(-45°,+45°)	∞	$\times$	$\surd$
MPFE	3	7	1	(-30°,+30°)	(-45°,+45°)	∞	$\surd$	$\surd$

depicting facial expressions of Anger(AN), Surprise (SU), Disgust (DI), Joy (JO), Sadness (SA), Fear (FE) and Neutral (NE). From the MultiPIE dataset, we use images of facial expressions of SU, DI, JO and NE, and from the Semaine dataset we use ten image sequences, coded by frame either as Speech or Laughter. The facial expressions in the BU3DFE dataset are posed, at four different levels of intensity, with the highest level corresponding to the apex of the expression. The facial expressions in the MultiPIE and MPFE datasets are also posed and depict only the apex of the expressions, while the expressions from the Semaine dataset are spontaneously displayed. In the case of the BU3DFE dataset, we rendered 2D facial images of 100 subjects at levels 3 and 4 of the expression intensity, and in 247 discrete poses (with 5° increment in pan and tilt angles), using the 3D range data. Images from all 247 poses were used during testing, whereas images from a subset of 35 poses (with 15° increment in pan and tilt angles) were used for training. The images from the MultiPie dataset depict 50 subjects captured at 4 pan angles (i.e., 0°,-15°,-30° and -45°). The MPFE dataset contains expressive images of 3 subjects and the Semaine dataset contains expressive images of 10 subjects, with various poses. All the images were annotated in terms of 39 facial points (e.g., see Fig.4.2). Specifically, the MultiPIE dataset was annotated manually, while for the BU3DFE dataset the locations of the facial points are provided by the dataset creators. The facial images from the MPFE and Semaine datasets were annotated automatically using the AAM tracker [56].

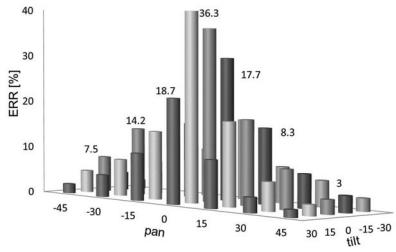


Figure 4.3: The error rate (ERR) per pose attained by the LDA-based pose classification. The subspace of poses was learned using $N = 200$ training data-pairs from each of the 35 training poses from the **BU3DFE** dataset. The average ERR is 9%.

4.3.2 Experiments on Synthetic Data

In this section, we present the experiments conducted on the BU3DFE dataset. The training dataset contained the locations of the facial points in 34 non-frontal poses, and the corresponding facial points in the frontal pose (thus, 35 poses in total). The training points were registered per pose, as described in Alg.4.2. For testing, we used the facial points from the training poses (tp) and the non-training poses (ntp). We measured the performance of the pose normalization using the root mean of the squared error (RMSE) computed between the pose-normalized facial points and the ground truth in the frontal pose. The performance of the facial expression classification was measured using the classification rate (RR) computed by applying the SVM classifier (F-SVM), trained using the training data in the frontal pose, to the pose-normalized facial points. If not stated otherwise, we applied 5-fold cross validation in all our experiments, with each fold containing images of different subjects.

Fig.4.3 shows the error rate for pose classification attained by taking the most likely discrete pose as the predicted class. The likelihood of each pose was obtained by the proposed pose estimation approach, described in Sec.4.2.2. As can be seen, the larger misclassification occurs in near-frontal poses. This is expected as the facial points in near-frontal poses are more alike than those in poses being far from the frontal pose. Note also that the misclassification occurs mostly between neighboring poses, which is a tractable problem for the CSGPR model, due to its definition of the weighting function (see Alg.4.2).

In the experiments for pose-normalization, we also evaluate standard Linear Regression (LR) and Support Vector Regression (SVR)[33], and recently proposed models for multi-output GPR: Twin GPR (TWINGPR)[28] and Multi-task GPR (MTGPR) [29]. As the baseline, we use independent GPRs (IGPRs)[159] for each output (i.e., each coordinate of each facial point). Also, analogously to the coupling of the SGPR models, we performed the coupling

Table 4.2: RMSE and RR attained by the base models for pose normalization and facial expression classification. The models were trained using N data-pairs per pose from the **BU3DFE** dataset. In the case of the regression-based methods, the classification was performed by applying F-SVM classifier to the pose-normalized facial points.

Method	RR (%)						RMSE (in pixel)					
	N=50	N=100	N=150	N=200	N=300	N=500	N=50	N=100	N=150	N=200	N=300	N=500
LR	57.2	59.8	62.3	67.0	67.9	68.4	2.71	2.45	2.11	1.72	1.80	1.83
SVR	64.3	68.4	70.3	71.5	71.9	72.1	1.85	1.42	1.27	1.18	1.15	1.09
TWINGPR	64.0	69.2	70.7	71.4	71.8	72.3	2.18	1.35	1.15	0.90	0.85	**0.80**
MTGPR	61.6	65.4	67.8	69.1	69.8	70.1	2.11	1.74	1.43	1.35	1.28	1.26
IGPR	68.1	70.6	72.0	72.3	72.8	73.2	1.71	1.22	0.98	0.93	0.88	0.83
SGPR	66.8	70.5	71.8	72.1	72.4	72.9	1.72	**1.15**	1.01	0.95	0.89	0.85
CIGPR	**69.2**	**72.2**	**73.1**	73.7	74.0	74.6	**1.68**	1.19	**0.95**	0.92	0.85	0.81
CSGPR	68.7	72.1	72.9	**73.9**	**74.2**	**74.9**	1.70	1.17	0.98	**0.90**	**0.82**	0.82
PW-SVM	60.3	66.4	68.5	70.4	72.7	73.3	-	-	-	-	-	-

of the IGPR models, to obtain the Coupled IGPR (CIGPR) models. We did so since IGPR has the same covariance form as SGPR, and, thus, the coupling of the IGPR models using the proposed framework is straightforward. Apart from TWINGPR, the hyper-parameters of all other GPR-based models were optimized by minimizing the negative log-likelihood of the models. In the case of TWINGPR, SVR and the pose-wise SVMs (PW-SVMs), we cross-validated the model parameters. In all models, we used a composite kernel function that is a sum of a linear term, an isotropic Radial Basis Function (RBF) and the model bias. We also include the results obtained by traditional shape models: the 2D-PDM[44] and 3D-PDM[230], and the appearance-based model AAM[56].

We compare the performance of the models mentioned above w.r.t. the amount of data used for training (when the pose is known). We used N training data for each pair of a non-frontal pose and the frontal pose, sampled uniformly from all the expression classes (at random) and from the 4 folds used for training. The 5th fold was used to test the models. This was repeated for all the folds. The average RMSE for pose normalization attained by different regression models is shown in Table 4.2. We also include the classification results attained by the PW-SVM classifiers. Note that MTGPR, specifically designed for dealing with multiple outputs, fails to outperform the other GP-based regression models in the target task. We noticed from the training and testing performance of this model that, for the given range of N, it was prone to over-fitting. This is possibly due to the large number of the outputs ($d = 78$), resulting in the large number of the parameters of the model to be learned. On the other hand, TWINGPR performs better pose-normalization (in terms of RMSE). However, this does not translate into RR attained by this model, compared to that of IGPR and SGPR, and their coupled counterparts, which outperform the other models on the target task. Finally, note that the PW-SVM classifiers require more training data to achieve RR similar to that

.

Table 4.3: The performance of different methods for pose-invariant facial expression classification trained using balanced (bal.) and imbalanced (imb.) data in 35 training poses (tp) from the **BU3DFE** dataset, and tested in a subject-independent manner using data in 247 test poses (tp and non-tp (ntp)) from the **BU3DFE** dataset, and corrupted by different levels of noise UNIF $\sim [-\sigma, \sigma]$, with $\sigma = 0, 2, 4$ pixels (where 10% of interocular distance for the average registered frontal-pose face in the **BU3DFE** dataset is approximately 5 pixels).

Method	RR ($\sigma = 0$)		RMSE ($\sigma = 0$)		RR ($\sigma = 2$)		RMSE ($\sigma = 2$)		RR ($\sigma = 4$)		RMSE ($\sigma = 4$)	
	tp	ntp	tp	ntp	tp	ntp	tp	ntp	tp	ntp	tp	ntp
Pose-wise												
PW-SVM *(bal.)*	70.5	68.2	-	-	68.9	67.3	-	-	66.7	65.0	-	-
PDM & F-SVM *(bal.)*												
2D-PDM	59.5	59.1	3.18	3.21	57.2	56.7	3.33	3.45	55.2	54.1	3.58	3.62
3D-PDM	62.1	62.3	2.70	2.67	61.8	61.0	2.78	2.89	59.9	59.2	3.12	3.09
Regression *(bal.)* & F-SVM												
LR - DB	64.3	63.1	2.08	2.12	60.1	59.9	2.45	2.52	56.1	54.3	2.85	2.91
SVR - DB	68.7	68.1	1.60	1.63	67.9	67.0	1.92	2.06	66.7	65.2	2.19	2.21
TWINGPR - DB	70.1	68.5	1.18	1.29	68.2	67.8	1.63	1.79	66.4	66.5	2.12	2.27
MTGPR - DB	67.8	66.5	1.36	1.45	66.7	66.1	1.50	1.61	65.2	64.4	2.09	2.13
MTGPR - VB	68.1	67.2	1.32	1.40	67.1	66.8	1.48	1.58	65.4	65.1	1.98	2.01
IGPR - DB	71.9	70.5	1.25	1.29	69.5	68.4	1.50	1.61	67.4	66.9	1.85	1.88
IGPR - VB	72.0	69.4	1.22	1.26	68.3	67.6	1.51	1.75	67.1	66.0	1.87	1.95
SGPR - DB	71.6	70.1	1.30	1.34	69.9	68.8	1.53	1.69	69.0	68.9	1.71	1.82
SGPR - VB	71.8	69.8	1.19	1.26	69.0	68.8	1.59	1.76	68.3	67.2	1.78	1.89
CIGPR	**72.9**	**72.2**	**1.01**	1.15	70.2	69.0	**1.34**	**1.42**	68.1	67.7	1.72	1.80
CSGPR	72.6	71.5	1.05	**1.11**	**70.5**	**69.4**	1.37	1.45	**69.9**	**69.7**	**1.64**	**1.71**
Regression *(imb.)* & F-SVM												
LR - DB	57.6	56.1	2.28	2.43	54.1	53.1	2.79	2.81	52.7	52.2	3.01	3.11
SVR - DB	60.3	60.1	1.85	1.87	59.0	58.2	2.03	2.17	57.1	57.0	2.43	2.55
TWINGPR - DB	63.7	62.5	1.45	1.60	59.0	58.8	2.01	2.11	58.6	57.9	2.33	2.42
MTGPR - DB	63.1	62.6	1.47	1.58	61.7	61.1	2.07	2.19	60.1	59.5	2.73	2.81
MTGPR - VB	63.4	62.9	1.41	1.53	61.9	61.4	1.98	2.18	60.6	60.1	2.67	2.71
IGPR - DB	65.1	64.6	1.52	1.61	62.5	62.0	2.01	2.10	60.0	59.8	2.52	2.58
IGPR - VB	64.9	64.3	1.41	1.57	62.1	61.8	2.01	2.11	60.2	59.2	2.58	2.60
SGPR - DB	64.4	63.9	1.59	1.66	62.7	61.9	1.98	2.07	60.8	60.0	2.40	2.49
SGPR - VB	64.5	64.4	1.52	1.63	63.1	62.1	1.97	2.04	61.2	60.9	2.44	2.50
CIGPR	**71.5**	**70.2**	**1.09**	**1.22**	69.8	67.9	1.51	1.72	68.4	67.7	1.97	2.01
CSGPR	71.1	69.2	1.15	1.31	**70.0**	**68.2**	**1.43**	**1.68**	**69.1**	**68.9**	**1.86**	**1.92**

of the GPR-based methods. Nevertheless, their RR remains lower than that attained by the coupled models.

So far, we evaluated the models using the noiseless data from the 35 training poses. We next test the robustness of the models to missing data and noisy data. To this end, we trained the regression models using balanced and imbalanced data (as explained below) sampled from the 35 training poses, and tested on noiseless and noise-corrupted data (with unknown pose) sampled from all 247 poses. The balanced dataset contained examples sampled (from 4 folds) per pose-pairs (non-frontal poses and the frontal pose) and from all seven facial expressions. The imbalanced dataset was prepared as follows: examples of Neutral facial expressions from 4 folds were used to train 50% of the pose-pairs, which were selected at random. For the rest of

the poses, training examples were selected as in the balanced dataset. The 5th fold, containing examples of all facial expressions, was used to test the models, and this was repeated for all the folds. Furthermore, the test data were corrupted by adding noise to the locations of the facial landmarks, as explained in Table 4.3. For the 2D- and 3D-PDM, we selected 13 and 17 shape bases, respectively. The shape bases were chosen from the balanced dataset so that 95% of the energy was retained. In the case of the 3D-PDM, we used the 3D facial points, and for the 2D-PDM we used the corresponding 2D facial points in the frontal pose. The PW-SVMs were trained using the balanced dataset, as in the previous experiment. In the case of 'non-coupled' regression models, the predictions from different non-frontal poses were combined using either DB or VB weighting, as described in Sec.4.2.1. The latter approach was used only for MTGPR, IGPR and SGPR, since these models provide uncertainty in their predictions. To reduce the computational load of the coupled models, the parameters P_{min} (see Alg. 4.2) and C_{min} (see Alg. 4.1) were set to 0.1 and 0.8, respectively[1]. Also, the number of coupled functions per pose was constrained to three.

Table 4.3 shows the comparative results. The performance of the 2D- and 3D-PDM is inferior to that of the PW-SVMs and the regression-based models. These results indicate that the face-shape-based models employed are unable to accurately recover facial-expression-related changes in the presence of large head movements. This, in turn, results in high RMSE and low RR attained by these two models. PW-SVM classifiers outperform the LR- and SVR-based methods, and perform similarly to the GPR-based methods, when trained on the balanced data and tested on the noiseless data in discrete poses. However, they are less robust to noise and pose changes (i.e., test data from non-training poses). Note that the results for the noiseless case and training poses differ from those shown in Table 4.2. This is caused by inaccuracies of the head pose estimation step. We also observe that TWINGPR is very sensitive to high levels of noise which is reflected in its RMSE and RR. IGPR and SGPR show similar performance, with SGPR performing better in most cases in the classification task. The performance of MTGPR in the target task is lower than that of IGPR, which, again, we attribute to the over-fitting of the model. On the other hand, CIGPR- and CSGPR-based methods outperform the other models. Note also that their performance remains stable in the case of non-training poses. This clearly suggests that these models are able to generalize well in the case of continuous change in poses despite the fact that they were trained on a limited set of data in discrete poses. Note also that using DB or VB weighting of the GPR-based models results in an inferior performance compared to that attained by the proposed coupled models, which use the CI fusion rule for combining the outputs of different mapping functions.

[1] We used a small validation set, containing examples of 5 randomly selected subjects, to set P_{min} and C_{min}.

We also observe that when the test data are corrupted by the noise, there is an expected decline in performance of all the models. However, this is less pronounced in CSGPR than in CIGPR, because in the former the base SGPR model preserves face structure by performing simultaneous prediction of the points. Finally, in the case of the imbalanced dataset, the performance of the 'non-coupled' models is substantially lower compared to that of the CIGPR and CSGPR models. This clearly shows the benefit of using the proposed coupling scheme. Since these two models exhibit similar performance, with CSGPR performing better in the case of the noisy data and being computationally much less intense, in further experiments we evaluate CSGPR and use SGPR (VB) as the baseline model. To determine the optimal number of training data needed to train the CSGPR model, in Fig.4.4 we show the pose normalization performance of this model w.r.t. the number of training data N. We note that CSGPR exhibits stable performance across the poses. In the following, we use $N = 200$ and $N = 500$ data to train the regression models and PW-SVMs, respectively, in order to keep them computationally tractable without significantly affecting their performance.

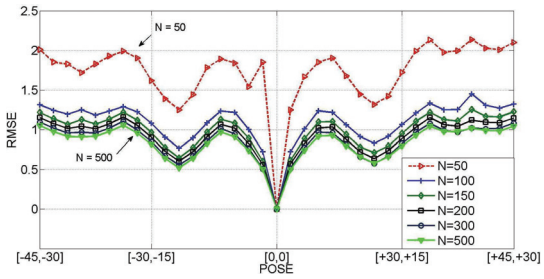


Figure 4.4: RMSE of head pose normalization attained by the CSGPR model trained per pose and by using N data from the **BU3DFE** dataset.

Fig.4.5 shows the confusion matrices for facial expression classification attained by the SGPR- and CSGPR-based methods. In contrast to the CSGPR-based method, RR of the SGPR-based method decreases considerably in the case of the imbalanced data compared to when this model is trained using the balanced data. However, the SGPR-based method outperforms the CSGPR-based method on the Neutral facial expression class (when trained using the imbalanced data). This is because, for some pose-pairs, the SGPR models are trained using data of Neutral facial expression only, and, thus, there is no need for their coupling. Still, the CSGPR-based method shows a better performance on average. Fig.4.6 depicts changes in the RMSE of different models across tested poses. As can be seen, the RMSE of the 3D-PDM increases rapidly in poses being far from frontal, indicating that the used 3D-PDM model is unable to accurately recover the 3D face shape from the 2D points in these poses. On average, the 3D-PDM and the LR-based method show a similar performance, and inferior to

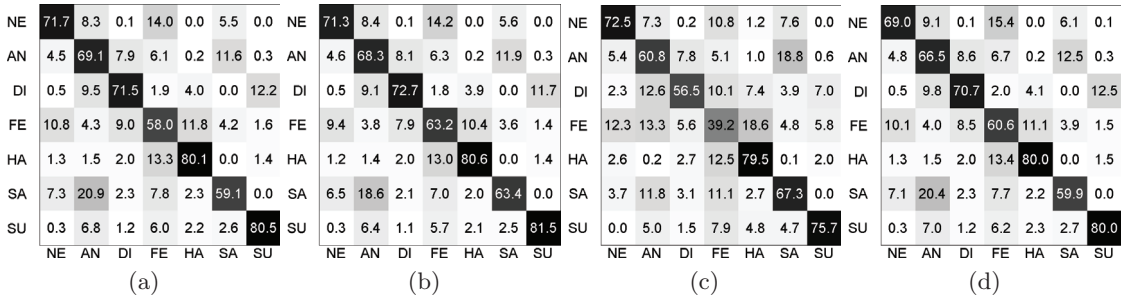


Figure 4.5: Confusion matrices for pose-invariant facial expression classification obtained by (a) SGPR (bal.), RR=70.1%, (b) CSGPR (bal.), RR=71.6%, (c) SGPR (imb.), RR=64.5% and (d) CSGPR (imb.), RR=70.2%. The methods were trained using noiseless data in 35 training poses from the **BU3DFE** dataset.

that obtained by the CSGPR-based method, which generalizes well even in the non-training poses.

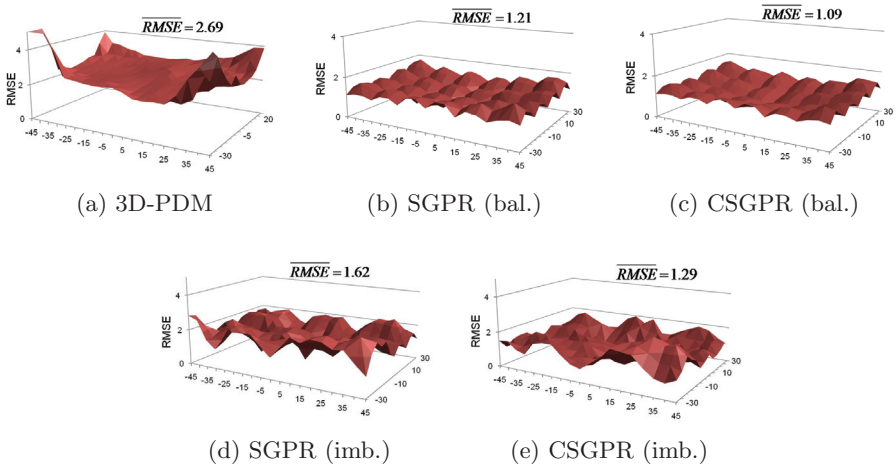


Figure 4.6: RMSE of the pose normalization in 247 tested poses attained by the 3D-PDM, and the SGPR and CSGPR models trained using the noiseless balanced/imbalanced data from the **BU3DFE** dataset.

Table 4.4 gives an overview of the results obtained by the proposed CSGPR-based method and previously proposed methods for pose-invariant facial expression classification on the BU3DFE dataset. When studying the results shown in Table 4.4, the following should be considered. First, the methods proposed in [83, 136, 84, 226] were trained/tested on a small set of discrete poses containing only pan rotations. In other words, they do not deal with large pose changes. Second, the methods proposed in [83, 136, 84, 226, 186] are person-specific since they use the neutral frame in the feature pre-processing step. Therefore, they

Table 4.4: The results of the state-of-the-art methods for pose invariant facial expression classification on the **BU3DFE** dataset.

Method	Classifier	Features	Poses			Expressions		RR
			tilt	pan	total	number	levels	tp(bal.)
Hu et al. [83]	pose-wise svm	41 landmarks	-	(0°,+90°)	5	6	1, 2, 3, 4	66.7%
Moore and Bowden [136]	pose-wise svm	lgbp\lbp	-	(0°,+90°)	5	6	1, 2, 3, 4	71.1%
Hu et al. [84]	single knn	sift+lpp	-	(0°,+90°)	5	6	2, 3, 4	73.8%
Zheng et al. [226]	pose-wise knn	83 landmarks+sift	-	(0°,+90°)	5	6	1, 2, 3, 4	78.5%
Zheng et al. [225]	single linear	sift + bda\gmm	(-30°,+30°)	(-45°,+45°)	35	6	4	68.3%
Tang et al. [186]	pose-wise svm	sift + hmm	(-30°,+30°)	(-45°,+45°)	35	6	4	75.3%
SGPR (lev. 3)	frontal svm	39 landmarks	(-30°,+30°)	(-45°,+45°)	247	7	3	68.2%
CSGPR (lev. 3)	frontal svm	39 landmarks	(-30°,+30°)	(-45°,+45°)	247	7	3	68.7%
SGPR (lev. 4)	frontal svm	39 landmarks	(-30°,+30°)	(-45°,+45°)	247	7	4	75.4%
CSGPR (lev. 4)	frontal svm	39 landmarks	(-30°,+30°)	(-45°,+45°)	247	7	4	76.5%

are inapplicable to real-world scenarios. The method proposed in this paper and the methods proposed in [226, 186] are the only ones that consider the 'full' range of poses, including pan and tilt rotations with a significant part of the face remaining visible. Yet, the methods in [226, 186] were evaluated on a set of discrete poses used for training, so it is not clear how these methods would perform in non-training poses. On the other hand, the proposed CSGPR method and the baseline SGPR method (C/SGPR methods) were evaluated on both training and non-training poses, and using balanced and imbalanced datasets. Furthermore, most of the methods in Table 4.4 were trained pose-wise, and, hence, cannot deal with missing facial expressions (i.e., the imbalanced data), as opposed to the C/SGPR-based methods. For the C/SGPR-based methods, in Table 4.4 we report the results per expression levels 3 and 4 separately so that they can be compared with the results of the other methods, which usually consider only the expression level 4. Note that in Table 4.3 (the noiseless case) we show the average results for levels 3 and 4.

4.3.3 Experiments on Real-image Data

We next run the experiments on the real-image data from the MultiPIE dataset. For this, we prepared the imbalanced datasets as follows: for pose (0°,−30°) and for facial expression of, e.g., Surprise, we removed all examples of this facial expression from the pose in question, and kept the examples of all four facial expressions in the two remaining (non-frontal) poses. This was repeated for each facial expression and non-frontal pose. Such datasets were then used to train the SGPR models for each pair of a non-frontal and the frontal pose, which, in the case of the CSGPR model, were then coupled.

Table 4.5 shows the performance of the C/SGPR-based methods trained using the balanced and imbalanced data from the MulitPIE dataset. In the former case, the testing was done using examples of all facial expressions in all non-frontal poses. The methods trained on the

.

Table 4.5: RMSE and RR (per expression) attained by the SGPR- and CSGPR-based methods, trained/tested using balanced (bal.) and imbalanced (imb.) data from the **MulitPIE** dataset.

	RR (%)				RMSE (in pixel)			
	SGPR (bal.)	CSGPR (bal.)	SGPR (imb.)	CSGPR (imb.)	SGPR (bal.)	CSGPR (bal.)	SGPR (imb.)	CSGPR (imb.)
NE	93.7	93.8	84.4	89.5	1.45	1.39	2.35	1.86
DI	92.0	91.7	75.7	82.1	1.60	1.52	2.91	1.91
JO	93.9	95.6	84.2	90.1	1.59	1.51	2.85	1.88
SU	96.6	98.1	82.8	88.7	1.65	1.59	2.95	2.31
Av.	94.1	**94.8**	81.8	**87.6**	1.57	**1.50**	2.76	**1.99**

imbalanced datasets were tested using only the examples of the missing facial expression in the target pose. As can be seen from Table 4.5, both the methods perform similarly when the balanced datasets are used. This is especially the case for facial expressions of Neutral and Disgust. We attribute this to the fact that, in the case of the perfectly balanced dataset, some of the coupled functions in the CSGPR model add noise to the final prediction in the frontal pose as a consequence of the registration process. In the case of the imbalanced dataset, the CSGPR-based method outperforms the SGPR-based method. Again, this is due to the SGPR-based method being unable to generalize well beyond the data in poses used for training.

.

Table 4.6: RMSE and RR (per expression) attained by the AAM (Candide) and the SGPR- and CSGPR-based methods, trained using balanced (bal.) and imbalanced (imb.) data from the **MPFE** dataset.

	RR (%)					RMSE (in pixel)				
	AAM (Cand.)	SGPR (bal.)	CSGPR (bal.)	SGPR (imb.)	CSGPR (imb.)	AAM (Cand.)	SGPR (bal.)	CSGPR (bal.)	SGPR (imb.)	CSGPR (imb.)
NE	72.2	83.4	85.0	73.2	79.1	3.51	1.85	1.61	2.85	2.55
AN	54.1	72.5	73.2	59.6	64.4	3.24	1.98	1.82	3.11	2.95
DI	58.7	73.0	74.8	62.7	70.1	4.13	2.25	2.11	3.80	3.62
FE	60.4	68.3	69.9	59.5	64.6	3.44	2.20	2.30	3.15	2.83
JO	72.9	87.2	89.1	77.0	83.2	4.21	2.38	2.42	3.45	3.15
SA	57.2	68.9	70.2	60.1	63.7	3.65	2.01	1.90	3.60	3.09
SU	78.1	88.5	91.5	76.7	85.2	4.42	2.61	2.51	3.45	3.11
Av.	64.8	77.4	**79.1**	67.0	**73.0**	3.8	2.18	**2.09**	3.34	**3.04**

We further compare the performance of the C/SGPR-based methods using the MPFE dataset. We also report the results attained using the AAM method from [56] for pose normalization. Specifically, we used the Candide model (being the 3D Active Shape Model part of the AAM) to perform the pose normalization by rotating the Candide model to the frontal pose, where the 2D (pose-normalized) facial points were obtained from the corresponding 3D points (see Fig.4.7). The manual initialization of the Candide model in the frontal pose, and the corresponding 2D points obtained from the initialization step were used as the ground

truth when computing the RMSE, and to train the F-SVM. Table 4.6 summarizes the average results per expression. As can be seen, the CSGPR-based method outperforms the AAM (Candide) in the task of pose normalization. This is because the pose normalization based on the Candide model is more susceptible to tracking errors, since, in contrast to the CSGPR-based method, no training data are used to smooth out the noise in its output. Also, the rotation matrix, used to bring the Candide model to the frontal pose, is learned based on the pose-estimation provided by the AAM [56]. So, the inaccuracy of the pose estimation also degrades the performance of this model. In the case of the imbalanced data, the CSGPR-based method largely outperforms the AAM- and SGPR-based methods. However, there is a decline in performance attained by all the methods. In the case of C/SGPR this is expected since they are trained using not only the imbalanced data but also the data of only two subjects (a three-fold person-independent cross-validation procedure was applied in this experiment).

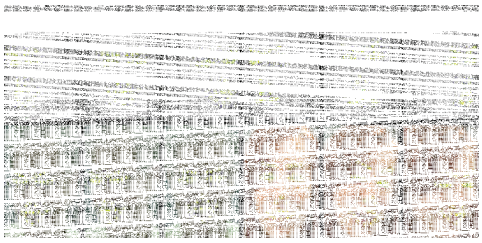


Figure 4.7: Example images from the **MPFE** dataset (*top*) and the **Semaine** dataset (*bottom*) with the facial points automatically localized by the AAM [56].

We also evaluated the proposed method on spontaneously displayed facial expressions from the Semaine dataset [135]. Specifically, we performed cross-database evaluation where the C/SGPR-based methods were trained using the MultiPIE and the MPFE datasets, and tested using the Semaine dataset. Table 4.7 shows that the C/SGPR-based methods generalize well, with CSGPR outperforming the base SGPR, despite the fact that they were trained using a different dataset from the one used for testing. Note also that the C/SGPR-based methods trained on the MPFE dataset perform better than when they are trained on the MultiPIE dataset. This is due to the difference in the localization of the facial points, which, in the case of the MultiPIE dataset, was done manually, and in the case of the MPFE and Semaine datasets was done automatically using the AAM [56].

4.4 Conclusions

In this Chapter, we proposed a method for pose-invariant facial expression classification that is based on 2D geometric features. This approach performs explicit pose normalization by

.

Table 4.7: RR for facial expressions of Laughter and Speech attained by the AAM (Candide), and the SGPR- and CSGPR-based methods, trained using balanced (bal.) and imbalanced (imb.) data from the **MPFE** and **MultiPIE** datasets, and tested on the **Semaine** dataset.

	RR (%)				
	AAM (Semaine)	SGPR (MPFE)	CSGPR (MPFE)	SGPR (MultiPIE)	CSGPR (MultiPIE)
Laughter	64.1	80.4	83.2	69.5	77.1
Speech	93.2	89.8	94.8	85.7	90.3
Av.	78.6	85.1	**89.0**	77.6	**83.7**

means of the proposed CSGPR model. We showed that this approach can deal effectively with expressive faces in poses within the range from $-45°$ to $+45°$ pan rotation and $-30°$ to $+30°$ tilt rotation, outperforming the state-of-the-art regression-based approaches to pose normalization, the 2D- and 3D-PDMs and the online AAM [56]. We also showed that the proposed approach performs accurately for 'continuous' changes in head pose, despite the fact that training was conducted on a limited set of discrete poses. Lastly, in contrast to the existing pose-invariant facial expression classification methods, the proposed method can be used for classification of facial expression categories that were not available in certain non-frontal poses during training, and requires less training data for achieving similar performance to that of the pose-wise classifiers.

Chapter 5

Shape-conformed Gaussian Process Regression for Pose Normalization

Contents

5.1 Introduction 83

5.2 Methodology 84

5.3 Experiments 89

5.4 Conclusions 94

5.1 Introduction

In this Chapter, we propose a model for facial-point-based pose normalization, named Shape-conformed Gaussian Process Regression (SC-GPR). This model achieves structured pose normalization of the facial points by modeling their relationships within the poses using a 2D deformable face-shape model. More specifically, we incorporate a prior knowledge about the facial shapes in non-frontal poses and the frontal pose into the kernel matrix of the standard GP regression. As a result, the output of the proposed SC-GPR model is encouraged to conform to anatomically feasible facial configurations. Note that this structure in the model output has not been accounted for by the CGPR model from Chapter 4, since this model performs structured pose normalization by exploiting relationships between data in different discrete poses, but not within the poses. However, the latter is of great importance when data, i.e., facial points in different poses, are corrupted by high levels of noise and/or outliers (e.g., due to inaccuracies in facial point localization, occlusions, etc.). The SC-GPR model is

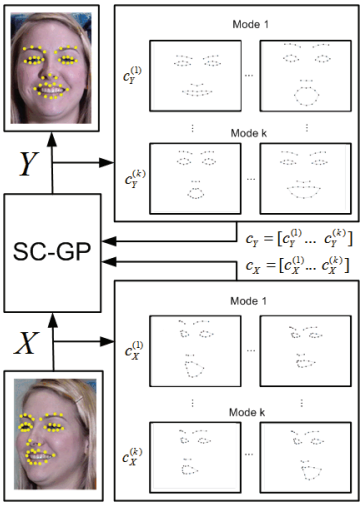


Figure 5.1: **Outline of the method:** The SC-GP regression model is used to map locations of 39 facial landmark points (X) extracted from a facial image in a non-frontal view to the corresponding points (Y) in the frontal pose. SC-GP achieves structured pose normalization by combining GP regression with deformable shape models, learned independently for the input and output faces. The face shapes are described by first k deformable modes of the deformable shape model used.

specifically designed to perform robust pose normalization in such scenarios. The outline of the proposed method is given in Fig 5.1.

5.2 Methodology

In this Section, we first briefly recall the standard GP regression model, and describe the deformable face-shape models. We then present the proposed SC-GP regression, and explain learning and inference in this model.

5.2.1 Gaussian Process Regression

The goal of GP regression is to learn mapping functions that can be used to map the input features onto an output space. Learning and inference in this model is explained in detail in Chapter 3. For notational convenience, we include only the most important results here. Given a training set $\mathcal{D} = \{X, Y\}$[1], containing N multi-dimensional inputs $X = [X_1, \dots, X_i, \dots, X_N] \in \mathcal{R}^{N \times D_X}$ and outputs $Y = [Y_1, \dots, Y_i, \dots, Y_N] \in \mathcal{R}^{N \times D_Y}$, where D_X and D_Y are the dimensions of the input and the output, respectively, the inference in GP regression is carried out by computing the mean and variance of the predictive distribution

[1]The inputs are linearly rescaled to have zero mean and unit variance on the training set.

for a test sample X_* as

$$m(X_*) = \mu_Y + K_*^T K^{-1}(Y - \mathbf{1}_{(N\times 1)}\,\mu_Y), \tag{5.1}$$

$$\sigma^2(X_*) = K_{**} - K_*^T K^{-1} K_*, \tag{5.2}$$

where μ_Y is the mean of all training outputs stored in Y. $K_* = K(X, X_*)$ denotes $N \times 1$ vector of the values of the kernel function $k(\cdot,\cdot)$ computed between the training data X and the test input X_*. Similarly, the entries of $K = K(X, X)$ and $K_{**} = K(X_*, X_*)$ are computed using the kernel function, which is usually defined as

$$k(X_i, X_j) = \exp\left(-\frac{\|X_i - X_j\|^2}{2\theta^2}\right) + \beta\delta_{ij}, \tag{5.3}$$

where δ_{ij} is the Kronecker delta function, which is 1 iff $i = j$, and 0 otherwise. The parameters (θ, β) of the kernel function in (5.3) are estimated by minimizing the negative log-likelihood

$$\mathcal{L} = \frac{D}{2}\ln|K| + \frac{1}{2}tr(K^{-1}YY^T) + const., \tag{5.4}$$

using the conjugate gradient algorithm [159].

5.2.2 Deformable Face-shape Model

In this section, we describe a deformable face-shape model that we use later in Sec.5.2.3 to define the face-shape prior in our model. In general, deformable shape models offer a unique and powerful approach to face representation that is capable of accommodating different sources of variation (e.g., facial expressions, the subject's identity, etc.) [44]. To learn the face deformations, we first collect N training data in the matrix $X = [X_1, \ldots, X_N]$, where $X_i \in \mathcal{R}^D$ is a vector of the coordinates of the facial points extracted from a facial image. We then follow the standard shape representation [44], where X_i is approximated as

$$X_i \approx \mu_X + c_{X_i} B_X^T, \tag{5.5}$$

where μ_X contains the coordinates of the mean face computed from X, and $c_{X_i} = [c_{X_i}^{(1)}, ..., c_{X_i}^{(d)}] \in \mathcal{R}^d$ are the shape parameters corresponding to d ($d < D$) deformable modes, i.e., eigenvectors of $(X - \mu_X)$ that are stored in $B_X = [b_X^{(1)}, ..., b_X^{(d)}] \in \mathcal{R}^{D\times d}$. Thus, the vector X_i can be reconstructed using the deformable shape model with the parameters $S_{X_i} = (\mu_X, B_X, c_{X_i})$. These parameters are learned by means of standard Principal Component Analysis (PCA) [23]. The deformable shape model with parameters obtained in this way is relatively robust to low-levels of Gaussian noise. However, it is highly sensitive to other sources of noise and outliers

in the data (e.g., caused by occlusions, erroneous hand labeling of the facial points, and/or inaccurate automatic facial point localization), due to the least-squares formulation of standard PCA. To deal with this, we employ Robust PCA [50], which estimates the data mean and deformable modes that are robust to outliers. In particular, we use this version of Robust PCA as it can handle intra-sample outliers which, in our case, can effect some, but typically not all of the facial points extracted from a facial image.

5.2.3 Shape-conformed GP (SC-GP) Regression

In this section, we describe the proposed SC-GP regression for pose normalization. We illustrate the method in the task of learning the mapping between 2D locations of the facial points (see Fig.5.1) from a non-frontal pose, denoted by X, and the corresponding points in the frontal pose, denoted by Y. In what follows, we first describe the face-shape prior that we use in our formulation of GP regression to conform its output to anatomically possible facial configurations. We then describe the optimization procedures for training and inference in the proposed method.

Face-shape Prior. In standard GP regression, the multiple output dimensions are assumed to be independent. However, modeling internal dependences within outputs, as well as inputs, helps to preserve the structure in the estimated output [7, 28, 30]. In the context of the target task, modeling the spatial correlations between the positions of the facial points is important for preserving anatomically feasible facial configurations in the model output. This can be achieved by including information about the face geometry, encoded by the deformable face-shape model explained in Sec.5.2.2, into GP regression. Formally, this is attained by defining a face-shape prior as

$$\alpha(S_i, S_j) = p(S_{X_i}, S_{X_j})p(S_{Y_i}, S_{Y_j}), \tag{5.6}$$

where the prior $\alpha(S_i, S_j)$ is data-driven and it measures similarity of the input-output data pairs (i, j), based on the corresponding facial shapes (S_i, S_j), defined in Sec.5.2.2. The goal of the face-shape prior is to enforce the training data pairs (i, j) with similar input face-shapes, (S_{X_i}, S_{X_j}), estimated from (X_i, X_j), to have similar output face-shapes,(S_{Y_i}, S_{Y_j}), estimated from (Y_i, Y_j). The similarity measures in (5.6) are defined as

$$p(S_{X_i}, S_{X_j}) = \exp(-\tfrac{1}{2}(c_{X_i} - c_{X_j})T_X(c_{X_i} - c_{X_j})^T), \tag{5.7}$$

$$p(S_{Y_i}, S_{Y_j}) = \exp(-\tfrac{1}{2}(c_{Y_i} - c_{Y_j})T_Y(c_{Y_i} - c_{Y_j})^T), \tag{5.8}$$

where the scaling matrices are defined as $T_X = \lambda_X^2 \cdot diag(\tau_X^1, ..., \tau_X^{d_X})$ and $T_Y = \lambda_Y^2 \cdot diag(\tau_Y^1, ..., \tau_Y^{d_Y})$. Here, $(\tau_X^1, ..., \tau_X^{d_X})$ and $(\tau_Y^1, ..., \tau_Y^{d_Y})$ are the (positive) eigenvalues corresponding to the deformable modes of X and Y, respectively, sorted in decreasing order. Therefore, in (5.7) and (5.8), we more heavily penalize the difference in the modes that contribute more to the reconstruction of the data, i.e., X and Y. The scaling parameters λ_X and λ_Y control the overall influence of the prior, as explained below.

The face-shape prior defined in (5.6) satisfies the properties of a semi-positive kernel function, so it can be incorporated into the kernel function of the GP regression model. By doing so, we obtain the kernel function of the SC-GP model as

$$K_{ij} = \alpha(S_i, S_j) + k(X_i, X_j) + \beta\delta_{ij}, \tag{5.9}$$

where the kernel function $k(\cdot, \cdot)$ and noise β are defined in Sec.5.2.1. This covariance function is positive-definite and it ensures that, in the case of test data corrupted by high levels of noise (or outliers), the model relies more on the face-shape prior than on the noisy inputs.

SC-GP: Training. The training of the SC-GP regression model is carried out as follows. First, we learn the deformable models independently for the inputs X and outputs Y, using either standard PCA or Robust PCA. The number of deformable modes d_X and d_Y is selected so that $\max(\|X - X^{pca}\|) < \eta_X$ and $\max(\|Y - Y^{pca}\|) < \eta_Y$, where η_X and η_Y are set so as to preserve 97% of the energy in X and Y, respectively. Second, we use the deformable models, $S_X = \{B_X, C_X, \mu_X, \tau_X^1, ..., \tau_X^{d_X}\}$ and $S_Y = \{B_Y, C_Y, \mu_Y, \tau_Y^1, ..., \tau_Y^{d_Y}\}$, and training data X and Y, to learn the hyper-parameters $hp = (\lambda_X, \lambda_Y, \theta, \beta)$ of the SC-GP model. This is performed by minimizing the negative log-likelihood $\mathcal{L}(X, Y, S_X, S_Y, hp)$, defined as in (5.4), w.r.t. hp using the Conjugate Gradients method [159]. The learned parameters $\{S_X, S_Y, X, Y, hp\}$ are then used during inference.

SC-GP: Inference. During inference, given the test input X_*, the goal is to estimate the output Y_*, that is

$$Y_* = \mu_Y + K_*(X, S_X, S_Y, X_*, c_{X_*}, c_{Y_*})^T K^{-1} Y, \tag{5.10}$$

For this, we need both the shape parameters, c_{X_*} and c_{Y_*}. While the parameters c_{X_*} can be obtained from the test input X_*, as explained in Sec.5.2.2, c_{Y_*} are unknown as they depend on the output Y_*. Thus, we have a chicken-and-egg problem: to estimate the output Y_* we need the shape parameters c_{Y_*}, and the other way round. We approach this problem by using either of the following two strategies:

1) **Direct approach.** During training, we also learn independently a set of linear ridge regressors (LRRs) for each output dimension. For a test input X_*, we first obtain an estimate of the output, $\widehat{Y}_*$, using LRRs. Then, we estimate the shape parameters $\widehat{c}_{Y_*}$ from the initial guess of $\widehat{Y}_*$, either by standard PCA or Robust PCA. The final output Y_* is obtained by evaluating (5.10) using X_*, c_{X_*} and $\widehat{c}_{Y_*}$.

2) **Iterative approach.** As in the direct approach, we first apply LRRs to obtain an initial estimate of the output, $\widehat{Y}_*^0$, and the corresponding shape parameters, $\widehat{c}_{Y_*}^0$. We then continue searching for the optimal parameters $\widehat{c}_{Y_*}$ of the output shape so that the output of the SC-GP regression model, given by (5.10), and $Y_*^{pca} - \mu_Y = \widehat{c}_{Y_*} B_Y^T$ are as close as possible. In this way, we iteratively examine the best candidate output shapes until convergence and, based on that, we update the SC-GP-predicted facial landmarks in the frontal view. Formally, we minimize the Euclidean norm of the difference[2]

$$L(c_{Y_*}) = K_*(c_{Y_*})^T K^{-1} Y^T - \widehat{c}_{Y_*} B_Y^T, \tag{5.11}$$

w.r.t. the unknown shape parameters

$$c_{Y_*} = \underset{c_{Y_*}^{(i)},\, i=1,\ldots,d_Y}{\arg\min} \; ||L(c_{Y_*}^{(i)})||_2. \tag{5.12}$$

This non-linear optimization problem is solved using a second order quasi-Newton optimizer with cubic polynomial line search for optimal step size selection, which uses the gradient of the objective function at $c_{Y_*}^{(i)}$, given by

$$\frac{\partial L(c_{Y_*})}{\partial c_{Y_*}^{(i)}} = \frac{L(c_{Y_*})}{\|L(c_{Y_*})\|} \cdot \left(\frac{\partial K_*^T(c_{Y_*})}{\partial c_{Y_*}^{(i)}} K^{-1} Y^T - \mathbf{e}_i B_Y^T \right), \tag{5.13}$$

where $\mathbf{e}_i$ is the i-th unit vector, i.e., the vector which is zero in all entries except the i-th at which it is 1. The gradient of the test covariance K_* at $c_{Y_*}^{(i)}$ is given by

$$\frac{\partial K_*(c_{Y_*})}{\partial c_{Y_*}^{(i)}} = \begin{bmatrix} -\lambda_Y^2 \tau_2^{(i)} (c_{Y_*}^{(i)} - c_{Y_1}^{(i)}) \alpha(S_*, S_1) \\ \vdots \\ -\lambda_Y^2 \tau_2^{(i)} (c_{Y_*}^{(i)} - c_{Y_N}^{(i)}) \alpha(S_*, S_N) \end{bmatrix}, \tag{5.14}$$

where $\alpha(S_*, S_i)$, $i = 1, \ldots, N$, is computed as in (5.6), between the test shapes and all the training shapes.

SC-GP vs. GP. We briefly comment here on the way the SC-GP and standard GP regression account for the structure in their output. In [166], the authors showed that if the training data

[2]For notational simplicity, in K_* we drop dependence on X, S_X, S_Y, X_* and c_{X_*}.

satisfy a set of linear constraints, the mean prediction of the standard GP regression implicitly satisfies these constraints. In other words, if the facial geometry can be learned accurately from training outputs Y (using a linear model as in Eq.5.5), then this geometry will also be preserved in the model output during inference. Note, however, that these constraints may not accurately represent the face geometry in the case of noise and/or outliers, both of which are expected in real-world data. This, in turn, may result in the output of the standard GP model, given by Eq.5.1, being under-constrained (or inadequately constrained) during inference of new facial points from non-frontal poses. On the other hand, during inference in the proposed SC-GP model, we minimize the quadratic cost in Eq.5.12, which conforms the output of GP regression to that generated by the (robust) deformable face-shape model. Since the employed deformable model (learned via either standard PCA or robust PCA) relies on the reconstruction bases that preserve face geometry while reducing the effects of noise and/or outliers, our model achieves more robust pose normalization, as evidenced by experiments presented below.

5.3 Experiments

We evaluated our approach using synthetic data from the BU-3D Facial Expression (BU3DFE) dataset [208], and two real-image datasets: the CMU Pose, Illumination and Expression dataset (MultiPie) [73], and multi-pose facial expression (MPFE) dataset recorded in our lab. These datasets are described in Chapter 4. All data were first registered per pose by applying an affine transformation learned using the three facial points: the nasal spine point and the inner corners of the eyes which were chosen since they are stable facial points, and are not affected by facial expressions. The registered data were then used to learn the regression models independently for each target pair of poses (a non-frontal and the frontal pose). The accuracy of the pose-normalization was measured using the Root-Mean-Square-Error (RMSE) defined as $\sqrt{\frac{1}{d}\|\Delta p\|^2}$, where Δp is the difference between the predicted pixel position of the facial landmarks in the frontal pose and the ground truth (the manually annotated landmarks in the frontal pose images). If not stated otherwise, the datasets were partitioned in a subject-independent manner and used in a 5-fold cross validation procedure.

In the experiments that follow, we compared the performance of the proposed SC-GP regression to that obtained by the standard GP regression and the state-of-the-art Twin GP [3] [28] regression model for structured-outputs. We also compared the SC-GP method to: (1) the

[3]The implementation of Twin GP regression has been obtained from the authors' webpage: http://ttic.uchicago.edu/ blf0218/software/TGP.htm

Table 5.1: RMSE (per expression) of head pose normalization attained by GP, direct SC-GP, iterative SC-GP, Twin GP, and 3D-PDM, trained on the **BU3DFE** data in 12 training poses and tested on the **BU3DFE** data in 70 test poses

Method	Expression							Av.
	Neutral	Surprise	Disgust	Joy	Anger	Fear	Sadness	
GP	1.45	2.51	2.31	2.31	2.12	1.97	1.73	2.04
SC-GP (dir.)	1.38	2.22	2.03	1.82	1.64	1.52	1.61	1.75
SC-GP (iter.)	1.32	**2.03**	**1.86**	1.71	1.48	1.40	1.62	1.64
Twin-GP	1.13	**2.15**	**1.97**	1.57	1.27	1.20	1.40	1.52
3D-PDM	2.12	2.83	2.58	2.55	2.25	2.39	2.07	2.40

nonlinear 3D Point Distribution Model (3D-PDM)[230], and (2) the Candide model, being the ASM part of the online AAM [56] that we used to automatically localize the facial landmarks in real images (see Sec.5.3.2).

5.3.1 Performance on Synthetic Data

For experiments on synthetic data, we rendered 2D multi-view expressive images from the BU3DFE dataset at pan angles from $0°$ to $-45°$, and tilt angles from $0°$ to $30°$ with a step of $5°$, which resulted in 70 poses in total. Only 12 poses (i.e., the poses sampled with a step of $15°$), were used to train the models, while all the 70 poses were used for testing. The data in each pose included expressive images of 50 subjects, showing facial expressions of six basic emotions (joy, sadness, anger, surprise, fear, and disgust, sampled at four different levels of intensity) plus neutral, which resulted in 1250 images per pose. For each image, we used 2D locations of 39 facial landmarks, illustrated in Fig.5.1, which were obtained from the 3D facial points provided by the dataset creators. These 2D facial points were further used as the features in our experiments. For SC-GP regression, we used the first 16 principal components (deformable modes) computed using standard PCA. In the case of the 3D-PDM, we selected 18 deformable modes. Note that the evaluated regression models were trained independently for each pair of a non-frontal pose and the frontal pose, and tested on the images from the corresponding non-frontal poses. Table 5.1 shows the evaluation results on the noise-free data. As can be seen, SC-GP regression outperformed standard GP regression and 3D-PDM. Twin GP outperformed SC-GP regression on average, although iterative SC-GP outperformed Twin-GP in the cases of facial expressions of Surprise and Disgust (the error values shown in bold). These two expressions are more challenging to normalize than the other expressions due to high variation in the corresponding facial landmarks. We show these results to demonstrate the performance when ideal training/test data are used. Note, however, that in real-world applications, where automatic point detectors and trackers are applied, the facial landmarks are usually noisy and/or contain outliers.

Table 5.2: The influence of different levels of noise and outliers on the pose normalization (in terms of RMSE) attained by GP, direct SC-GP, iterative SC-GP, Twin GP, and 3D-PDM. The regression models were trained on the **BU3DFE** noise-free data in 12 training poses, and tested on the **BU3DFE** data in 70 test poses corrupted by different levels of uniformly distributed noise UNIF $\sim [-\alpha, \alpha]$, with $\alpha = 0..5$ pixels ($\alpha = 5$ is 10% of interoccular distance for the registered average frontal-pose face in the **BU3DFE** dataset), and by different levels of bias, $\beta = 0, \ldots, 25$ pixels, added to the locations of 3–5 randomly selected facial points

Method	α						β					
	0	1	2	3	4	5	0	5	10	15	20	25
GP	2.04	2.10	2.21	2.31	2.72	3.10	2.09	2.36	2.88	3.56	3.90	3.99
SC-GP (dir.)	1.75	1.80	1.92	2.04	2.22	2.35	1.84	2.12	2.41	2.63	2.81	2.95
SC-GP (iter.)	1.64	1.70	**1.82**	1.92	2.05	2.14	1.73	**1.99**	2.35	2.51	2.68	2.79
Twin-GP	1.52	1.53	**1.85**	2.22	2.60	3.09	1.55	**2.11**	2.70	3.20	3.38	3.62
3D-PDM	2.40	2.70	2.81	2.93	2.97	2.99	2.45	2.61	2.78	2.95	3.20	3.37

In order to investigate the robustness of SC-GP regression to noise and outliers in test data, we ran two sets of experiments using: noisy data and data containing outliers. In both cases, the training/test data were pre-processed by standard PCA, in the case of noise, and Robust PCA, in the case of outliers. We did this to have a fair comparison of the models. We first evaluated their performance in the presence of noise in the BU3DFE data corrupted by adding five levels of uniformly distributed noise. As can be seen from Table 5.2 (RMSE values for α), SC-GP regression clearly outperforms GP regression and 3D-PDM. Although Twin GP performs better than SC-GP in the case of low noise levels ($\alpha < 2$), the results clearly suggest that SC-GP is more robust to higher levels of noise. The robustness of SC-GP regression to noise comes from the shape regularization attained by the face shape prior, resulting in effective recovery of shape details from very noisy observations. In addition, SC-GP regression with the iterative inference outperformed SC-GP with the direct inference method. We attribute this to the ability of the former inference approach to refine the initial estimate of the shape parameters by the minimization process in (5.12).

In real-world applications, the input data may contain undesirable artifacts due to occlusions, changes in illumination, or inaccurate face/facial point detection/tracking, resulting in outliers, i.e., observations deviating markedly from the majority of the training samples. To evaluate the performance of the models in the presence of outliers, the BU3DFE data were corrupted by adding different levels of bias to the locations of 3–5 randomly selected facial points. In this experiment, for SC-GP regression we used the first 20 deformable modes computed by Robust PCA. As can be seen from Table 5.2 (RMSE values for β), standard GP regression, Twin GP regression, and 3D-PDM were all outperformed by SC-GP. As before, and for the same reasons, the iterative SC-GP model outperformed its counterpart with the direct inference.

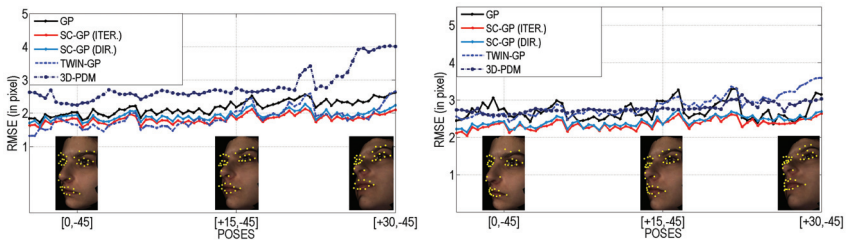


Figure 5.2: RMSE (per pose) obtained by the regression models trained on the **BU3DFE** data in 12 training poses, and tested on the **BU3DFE** data in 70 test poses corrupted by the noise level $\alpha = 2$ (*left*) and the bias level $\beta = 10$ (*right*). Note the difference between the facial points in the corresponding images.

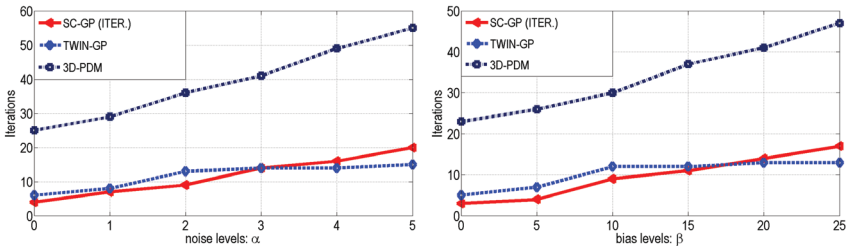


Figure 5.3: The number of iterations required for iterative SC-GP, Twin-GP and 3D-PDM, trained on the **BU3DFE** noise-free data in 12 training poses, to converge when tested on the **BU3DFE** data in 70 test poses, corrupted by noise levels α (*left*) and bias levels β (*right*).

Fig.5.2 shows the generalization ability of the models across 70 poses (only 12 of which were used for training) in the presence of the intermediate noise level ($\alpha = 2$), and outliers ($\beta = 10$). In the case of noise (Fig.5.2 (left)), all the models except 3D-PDM were able to generalize well across the poses. More specifically, SC-GP and Twin-GP regression perform comparably (with SC-GP outperforming Twin-GP in poses being further away from the frontal), while both outperform the head pose normalization attained by standard GP regression. The poor performance of 3D-PDM in poses towards $(+30, -45)$ is due to the occlusions of certain facial points in 2D face-images that occur in those poses. From Fig.5.2 (right), we see that Twin-GP is particularly sensitive to outliers. This is because the KL distance minimized in Twin-GP regression is not robust to non-Gaussian data. Fig.5.3 shows the performance of SC-GP regression, Twin-GP regression, and 3D-PDM in terms of the number of iterations required by these models to converge when tested on noisy/outlier data. Both iterative SC-GP and Twin-GP regression converged considerably faster than 3D-PDM. On average, these models converged in 9.6, 10.3 and 34 iterations, in the case of the noisy data, and 11.6, 9.5 and 39 iterations, in the case of the data corrupted with outliers, respectively.

Table 5.3: RMSE (per expression) of head pose normalization attained by GP, direct SC-GP, iterative SC-GP, and Twin GP, trained/tested using the **MultiPie** data in the four discrete poses

Method	Expression				Av.
	Neutral	Surprise	Disgust	Joy	
GP	1.84	2.47	2.25	2.23	2.20
SC-GP (dir.)	1.41	1.91	1.74	1.57	1.67
SC-GP (iter.)	1.45	1.80	1.66	1.52	**1.61**
Twin-GP	1.52	2.08	1.92	1.76	1.82

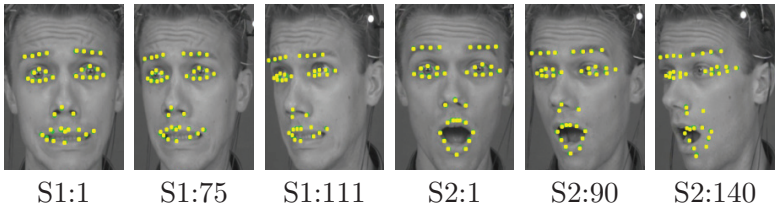


Figure 5.4: **The MPFE dataset:** Sample facial images (with automatically tracked facial points using [56]) from two sequences (S1 and S2) depicting Fear (*top*) and Surprise (*bottom*), while the pose is changing from $(0°, 0°)$ to $(0°, -45°)$. The corresponding frame numbers are given below each image.

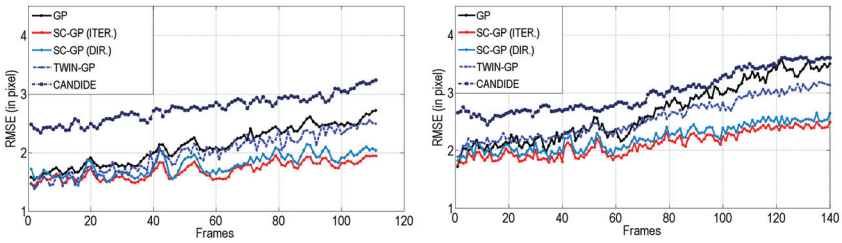


Figure 5.5: RMSE (per frame) of head pose normalization for two image sequences (Fear – *left* and Surprise – *right*), of the subject in Fig.5.4, attained by GP, direct SC-GP, iterative SC-GP, Twin-GP, and the Candide model of the tracker [56]. The models were trained using data of the other two subjects from the **MPFE** dataset.

5.3.2 Performance on Real Data

In the experiments on real image data, we used the MultiPie dataset: images of 50 subjects displaying 4 facial expressions of neutral, disgust, surprise, and joy, captured at 4 pan angles ($0°$,$-15°$,$-30°$ and $-45°$), resulting in 200 images per pose. These images were annotated in terms of 39 hand-labeled landmark points. For SC-GP regression, we used the first 7 deformable modes computed by standard PCA. As can be seen from Table 5.3, in the case of real image data, both SC-GP and Twin-GP regression clearly improve standard GP regression, while SC-GP outperforms Twin GP. Although the facial landmarks were manually annotated, this does not guarantee a 'perfect' annotation, especially in cases where some of the points are not clearly visible in the image, due to the head pose. So, the annotation errors must be expected, introducing additional non-linearities in the mapping to be learned. This, evidently, cannot be handled well by standard GP nor Twin GP regression.

We also performed experiments on real image sequences from the MPFE dataset. The locations of the 39 facial landmark points were obtained by applying the online AAM [56] (see Fig.5.4). The Candide model from this AAM was also used to attain head pose normalization by rotating it to the frontal pose in order to obtain the 2D-image coordinates. The regression models were trained/tested as explained above, and, for each sequence, the Candide model was manually fitted in the first frame, and the corresponding 2D points obtained from this model were used as the ground truth when computing the RMSE. Table 5.4 summarizes the average RMSE per expression, computed for all the image sequences. As can be seen, SC-GP outperforms Twin GP regression for all facial expressions.

Table 5.4: RMSE (per expression) of head pose normalization attained by GP, direct SC-GP, iterative SC-GP, Twin GP, and Candide model, trained/tested in the subject independent manner using the data from the **MPFE** dataset.

Method	Expression							Av.
	Neutral	Surprise	Disgust	Joy	Anger	Fear	Sadness	
GP	2.38	4.44	3.80	3.48	3.23	2.85	3.26	3.35
SC-GP (dir.)	2.04	3.22	3.11	2.46	2.34	2.15	2.47	2.60
SC-GP (iter.)	2.00	3.07	2.83	2.59	2.24	2.12	2.49	**2.48**
Twin-GP	2.40	3.47	3.26	3.07	2.59	2.70	2.91	2.91
Candide	3.28	4.36	4.00	4.18	3.45	3.52	3.38	3.74

Fig.5.5 summarizes the average RMSE per frame for two image sequences shown in Fig.5.4. Note that in poses being far from frontal, the tracker [56] employed estimates the locations of the facial points less accurately than in near-frontal poses. This, in turn, resulted in GP and Twin-GP being outperformed by SC-GP regression. The improved performance by SC-GP is due to its use of the deformable face-shape model, which helped to reduce the adverse effects of tracking errors. Note also that all the regression models achieved better results than those obtained by the Candide model. This is because this model could not recover the 3D face shape accurately, resulting in a significant loss of the facial-expression-specific details.

5.4 Conclusions

In this Chapter, we proposed the Shape-conformed GP regression model for facial-point-based pose normalization. We showed that the proposed model outperforms the standard GP regression, 3D-PDM and AAM, on the target task, especially in the case of the expression data corrupted by high levels of noise and outliers. This is mainly due to its ability to efficiently exploit the information about the face geometry via its kernel function, and the cost function used during inference. Consequently, it forces the model output to conform to anatomically feasible facial configurations. Moreover, the proposed model performs similarly or better than

the state-of-the-art Twin GP regression model with structured output. We attribute this to the fact that TwinGP, like other GP-based regression models with structured output (e.g., [30, 7, 213]), attempts to learn correlations between the outputs without taking into account domain knowledge (i.e., the face geometry). By contrast, in the proposed model this is incorporated by means of a deformable face-shape model. For this reason, our model performs better than Twin GP in the case of high levels of noise and outliers in the synthetic data, and also in the case of real-image data, where automatically localized facial points are used. Finally, note that the proposed SC-GP regression model for pose normalization can be used as an integral part of our approach to pose-invariant facial expression classification, introduced in Chapter 4. In the experiments presented in the following Chapter, we show performance of the SC-GP-based facial expression classification.

Chapter 6

Discriminative Shared Gaussian Process Latent Variable Model for Multi-view Facial Expression Classification

Contents

6.1 Introduction 97
6.2 Methodology 99
6.3 Experiments 104
6.4 Conclusions 107

6.1 Introduction

The method for pose-invariant facial expression classification proposed in Chapter 4, performs classification of the target expressions directly in the observation space of the pose-normalized facial points. This method has a number of limitations: (i) learning of the mappings for pose-normalization, and the expression classifier in the frontal pose is performed independently. Consequently, high-dimensional noise on the pose-normalized facial points can adversely affect the model's classification performance. (ii) The canonical view for expression classification has to be selected in advance. While in Chapter 4 we assumed that it is frontal, this may not hold for all expression categories/features, as shown in [136]. (iii) It models pair-wise dependencies between non-frontal poses and the frontal pose. Yet, modeling dependencies

among multiple poses simultaneously is expected to result in more robust facial expression classification. This is because the discriminative information from certain poses can be used to augment the classification in underperfoming views, i.e., the views where it is more difficult to differentiate between target expression categories. (iv) Handling appearance-based facial features is not trivial with this method because of the excessive number of features to be pose-normalized (typically > 5000). Learning the pose normalization mappings with such the number of outputs can easily lead to overfitting of the model parameters, and, therefore, limit its performance during inference.

To address the limitations mentioned above, in this Chapter we introduce the Discriminative Shared Gaussian Process Latent Variable Model (DS-GPLVM) for multi-view facial expression classification.[1] In this model, we exploit the fact that multi-view facial expression data are different manifestations of the same latent content, which can be represented on a common expression manifold. We model this manifold using the notion of Shared GPs (S-GPs) [58]. Furthermore, since our ultimate goal is expression classification, we place a discriminative prior informed by the expression labels, over the manifold. The classification of the target expressions is then performed on the learned manifold. The introduced DS-GPLVM is a generalization of the discriminative GP Latent Variable Models (D-GPLVM) [198], proposed for classification of data from a single observation space on a non-linear data manifold. More specifically, we combine the modeling strategy of S-GPs and D-GPLVM to perform classification of data from multiple observation spaces (i.e., different views of facial expressions) in a shared manifold. Methodologically, the proposed combination is novel as existing models based on S-GPs are devised for unsupervised dimensionality reduction but not discriminative subspace learning. On the other hand, D-GPLVM models perform discriminative subspace modeling of a single observation space, but not multiple. By contrast, DS-GPLVM accomplishes discriminative subspace learning from multiple observation spaces, and does so simultaneously. Also, in contrast to our GP-based models based on explicit pose-normalization, the resulting model avoids the need for a canonical view. Furthermore, it is a kernel-based model, and therefore it can easily deal with high-dimensional input features as well as complex non-linear data structures. The outline of the proposed model is given in Fig.6.1.

[1] We use the terms pose-invariant and multi-view interchangeably. The latter, thus, does not refer to the traditional definition of multi-view models where data from multiple views are all used during inference.

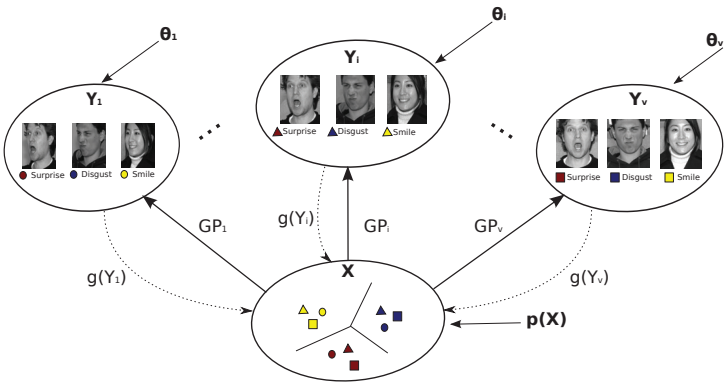


Figure 6.1: In the proposed DS-GPLVM, we first learn the discriminative shared manifold $\mathbf{X}$ of facial images from V different views ($\mathbf{Y}_i$, $i = 1 \dots V$). To enforce the class separation, we place the newly introduced discriminative prior $p(\mathbf{X})$, informed by the emotion labels, over the manifold. Note that a GP for each view is defined by a view-specific covariance matrix computed from the latent variables $\mathbf{X}$ that are shared among all the views. We also incorporate learning of the mapping functions ($g(\mathbf{Y}_i), i = 1 \dots V$), which are used to project data from each view onto the manifold. During inference, the query facial image from view i is projected onto the shared manifold by using the mapping $g(\mathbf{Y}_i)$, followed by classification of the projected image by means of the k-NN classifier learned directly in the shared manifold.

6.2 Methodology

In this Section, we first give a brief overview of the GP Latent Variable Model (GPLVM) [110] for learning a low-dimensional manifold of a single observation space (e.g., data of facial expressions in a single view). We then describe the proposed Discriminative Shared GPLVM (DS-GPLVM) [58] for learning a low-dimensional discriminative manifold, shared among multiple observation spaces, which is then used for expression classification.

6.2.1 Gaussian Process Latent Variable Model (GPLVM)

The GPLVM [110] is a probabilistic model for non-linear dimensionality reduction. It is devised for learning of a low dimensional latent space $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N]^T \in \mathcal{R}^{N \times q}$, with $q \ll D$, corresponding to the high dimensional observation space $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_N]^T \in \mathcal{R}^{N \times D}$. The key difference between GPLVM and standard GP regression is that the inputs in GPLVM are treated as unknown latent processes. Specifically, by using the covariance function $k(\mathbf{x}_i, \mathbf{x}_j)$, the likelihood of the observed data, given the latent coordinates, is

$$p(\mathbf{Y}|\mathbf{X}, \theta) = \frac{1}{\sqrt{(2\pi)^{ND}|\mathbf{K}|^D}} \exp(-\frac{1}{2}\mathrm{tr}(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^T)), \tag{6.1}$$

where $\mathbf{K}$ is the kernel matrix with the elements given by $k(\mathbf{x}_i, \mathbf{x}_j)$. This covariance function is usually chosen as the sum of the Radial Basis Function (RBF) kernel, and the bias and

noise terms

$$k(\mathbf{x}_i, \mathbf{x}_j) = \theta_1 \exp(-\frac{\theta_2}{2}\|\mathbf{x}_i - \mathbf{x}_j\|^2) + \theta_3 + \theta_4 \delta_{i,j}, \tag{6.2}$$

where $\delta_{i,j}$ is the Kronecker delta function and $\theta = (\theta_1, \theta_2, \theta_3, \theta_4)$ are the kernel parameters. The latent space positions X are obtained by minimizing the negative log likelihood $-log(p(\mathbf{Y}|\mathbf{X}, \theta))$ w.r.t. $(\mathbf{X}, \theta)$, which is given by

$$\mathcal{L} = \frac{D}{2} \ln |\mathbf{K}| + \frac{1}{2} \text{tr}(\mathbf{K}^{-1}\mathbf{Y}\mathbf{Y}^T) + const. \tag{6.3}$$

Note that this minimization is similar to that in standard GP, with the main difference being that now we treat X as latent variables.

6.2.2 Discriminative Shared GPLVM (DS-GPLVM)

In this Section, we introduce the DS-GPLVM for multi-view facial expression classification. DS-GPLVM uses the notion of the Shared-GPLVM [58] for learning latent variables $\mathbf{X}$ shared among V observation spaces $\mathbf{Y} = \{\mathbf{Y}_1, \ldots, \mathbf{Y}_V\}$ (i.e., different views). This is achieved by modeling V GPs, where each GP generates one observation space from the shared latent space. Note that a GP for each view is defined by a view-specific covariance matrix computed from the latent variables X that are shared among all the views. Formally, the joint marginal likelihood of a set of the observation spaces is given by

$$p(\mathbf{Y}_1, \ldots, \mathbf{Y}_V|\mathbf{X}, \theta_s) = p(\mathbf{Y}_1|\mathbf{X}, \theta_{Y_1}) \ldots p(\mathbf{Y}_V|\mathbf{X}, \theta_{Y_V}), \tag{6.4}$$

where $\theta_s = \{\theta_{Y_1}, \ldots, \theta_{Y_V}\}$ are the kernel parameters for each observation space, and the kernel function is defined as in (6.2). The shared latent space $\mathbf{X}$ is optimal for the reconstruction of multiple observation spaces, but not for their classification. For this, we use the maximum a posteriori (MAP) learning strategy, which allows us to define an arbitrary prior $p(X)$ over the shared space. This is expressed by

$$p(\mathbf{X}|\mathbf{Y}_1, \ldots, \mathbf{Y}_V, \theta_s) \propto p(\mathbf{Y}_1, \ldots, \mathbf{Y}_V|X, \theta_s)p(\mathbf{X}), \tag{6.5}$$

where $p(\mathbf{Y}_1, \ldots, \mathbf{Y}_V|X, \theta_s)$ is defined as in (6.4). In the following, we define a discriminative prior that enforces the latent positions in $\mathbf{X}$ to be separated in the shared space based on their class labels.

Discriminative Shared-space Prior. To define a discriminative shared space prior, we adopt the modeling approach of the discriminative GPLVM models for a single observation space proposed in [198, 227]. Specifically, in the Discriminative GPLVM (D-GPLVM) [198], the authors define a parametric prior based on Linear Discriminant Analysis (LDA) [23], which

tries to maximize between-class separability and minimize within-class variability in the latent space. On the other hand, in the GP Latent Random Field (GPLRF) [227] model, the authors define a non-parametric prior using the notion of the graph Laplacian matrix. We follow the latter approach in our definition of the shared-space prior, as it also allows us to include the observed data $\mathbf{Y}$ in the prior. This is important because in this way we can ensure that similarity between the observed data $\mathbf{Y}$ is preserved in the latent space $\mathbf{X}$, and, thus, avoid diverging solutions (see [198] for details).

The construction of a prior using the graph Laplacian matrix is explained in detail in Sec.9.2.2. Here, we describe it briefly. The graph Laplacian matrix [40] is defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where W is a weight matrix with elements encoding similarity between two training examples. $\mathbf{D}$ is a diagonal matrix with elements $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}$. For the target task, we first construct the weight matrix $\mathbf{W}^{(v)}$ for each view $v = 1, \ldots, V$ using data-dependent weights, defined as

$$\mathbf{W}_{ij}^{(v)} = \begin{cases} \exp\left(-t^{(v)} \|\mathbf{y}_i^{(v)} - \mathbf{y}_j^{(v)}\|^2\right) & \text{if } \mathbf{y}_i \text{ and } \mathbf{y}_j,\ i \neq j, \text{ belong to the same class} \\ 0 & \text{otherwise.} \end{cases} \tag{6.6}$$

where $\mathbf{y}_i^{(v)}$ is the i-th training example from the v-th view $\mathbf{Y}_v$ and $t^{(v)} > 0$ is the scale of the RBF kernel. The graph Laplacian of view v is then obtained as $\mathbf{L}^{(v)} = \mathbf{D}^{(v)} - \mathbf{W}^{(v)}$. Since the graph Laplacians for different views vary in their scale, we use the normalized graph Laplacian defined as

$$\mathbf{L}_N^{(v)} = \mathbf{D}_v^{-1/2} \mathbf{L}^{(v)} \mathbf{D}_v^{-1/2}. \tag{6.7}$$

Subsequently, we define the (regularized) joint graph Laplacian as

$$\tilde{\mathbf{L}} = \mathbf{L}_N^{(1)} + \mathbf{L}_N^{(2)} + \ldots + \mathbf{L}_N^{(V)} + \mu \mathbf{I} = \sum_v \mathbf{L}_N^{(v)} + \mu \mathbf{I}, \tag{6.8}$$

where $\mathbf{I}$ is the identity matrix, and μ is a regularization parameter. This regularization is required to ensure that $\tilde{\mathbf{L}}$ is positive-definite [229]. This, in turn, allows us to define the discriminative shared-space prior as

$$p(\mathbf{X}) = \prod_{v=1}^{V} p(\mathbf{X}|\mathbf{Y}_v)^{\frac{1}{V}} = \frac{1}{V \cdot Z_q} \exp\left[-\frac{\beta}{2} \mathrm{tr}(\mathbf{X}^T \tilde{\mathbf{L}} \mathbf{X})\right]. \tag{6.9}$$

Here, Z_q is a normalization constant and $\beta > 0$ is a scaling parameter. The discriminative shared-space prior in (6.9) aims at maximizing the class separation in the manifold learned from data from all the views.

DS-GPLVM: Learning. By plugging the discriminative shared-space prior into the negative log of the likelihood (6.5), we arrive at the following minimization problem

$$\min_{\mathbf{X},\theta_s} \sum_v \mathcal{L}^{(v)} + \frac{\beta}{2}\text{tr}(\mathbf{X}^T\tilde{\mathbf{L}}\mathbf{X}), \tag{6.10}$$

where $\mathcal{L}^{(v)}$ is given by (6.3) for each view. The penalty incurred by the prior is controlled with the parameter β. The minimization is carried out using the conjugate-gradients algorithm [159], where the gradients of (6.10) w.r.t. the latent positions $\mathbf{X}$ are given by

$$\frac{\partial L_s}{\partial \mathbf{X}} = \sum_v \frac{\partial L^{(v)}}{\partial \mathbf{X}} + \beta\tilde{\mathbf{L}}\mathbf{X}, \tag{6.11}$$

where we apply the chain rule to the log-likelihood of each view, i.e., $\frac{\partial L^{(v)}}{\partial \mathbf{X}} = \frac{\partial L^{(v)}}{\partial \mathbf{K}_v}\frac{\partial \mathbf{K}_v}{\partial x_{ij}}$, and

$$\frac{\partial L^{(v)}}{\partial \mathbf{K}_v} = \frac{D}{2}\mathbf{K}_v^{-1} - \frac{1}{2}\mathbf{K}_v^{-1}\mathbf{Y}_v\mathbf{Y}_v^T\mathbf{K}_v^{-1}. \tag{6.12}$$

The gradients of (6.10) w.r.t. the kernel parameters θ_s are derived as in the standard GP regression model [159]. The kernel parameters $t^{(v)}$ of the kernel in the weight matrices $W^{(v)}$ are set as explained in Sec.6.3. Finally, the penalty parameter β and the regularization parameter μ in the joint Laplacian matrix are set using a cross-validation procedure designed to optimize the classification performance of the classifier learned in the shared manifold.

DS-GPLVM: Inference. To draw inference of a test point from view $v = 1 \dots V$, $\mathbf{y}_i^{(v)}$, we need first to learn the inverse mappings from the observation space $\mathbf{Y}_v$ to the shared space $\mathbf{X}$ [111]. This is attained the inverse mapping functions by learning for each view separately as

$$x_{ij} = g_j^{(v)}(\mathbf{y}_i^{(v)}; \mathbf{a}) = \sum_{m=1}^{N} a_{jm}^{(v)} k_{bc}^{(v)}(\mathbf{y}_i^{(v)} - \mathbf{y}_m^{(v)}), \tag{6.13}$$

where x_{ij} is the j-th dimension of $\mathbf{x}_i$, and $g_j^{(v)}$ is modeled using kernel ridge regression [23]. To obtain smooth inverse mapping, we apply the RBF kernel to each dimension of the training data as

$$k_{bc}^{(v)}(\mathbf{y}_i^{(v)} - \mathbf{y}_m^{(v)}) = \exp(-\frac{\gamma_v}{2}\|\mathbf{y}_i^{(v)} - \mathbf{y}_m^{(v)}\|^2), \tag{6.14}$$

where γ_v are the kernel inverse width parameters for each observation space v, which are set in the same way as in the joint Laplacian matrix. The weight parameters $\mathbf{A}^{(v)}$ of the kernel ridge regression are found in the closed form as

$$\mathbf{A}^{(v)} = \mathbf{X}^T(\mathbf{K}_{bc}^{(v)} + \lambda\mathbf{I})^{-1}, \;\; v = 1 \dots V, \tag{6.15}$$

where $\mathbf{K}_{bc}^{(v)}$ is the kernel matrix computed over the training data from view v. The regularization term $\lambda\mathbf{I}$ helps to stabilize the inverse numerically by bounding the smallest eigenvalues of the kernel matrix away from zero. Once the test sample is projected onto the shared manifold using the learned inverse mappings, the classification can be accomplished by using any classifier trained on the shared manifold. In this paper, we employ the linear k-NN classifier.

Relation to multi-task learning methods. A common approach in multi-task learning methods is to exploit relationships between different tasks in order to facilitate their learning. In the context of the target task, existing methods typically consider different views of faces [93, 179] or human actions [129], as tasks to be modeled. The goal here is to learn view-specific projections onto a common latent space, followed by their classification. For instance, in [93], the authors extend Linear Discriminant Analysis (LDA) to the multiview case by maximizing between-class and minimizing within-class variation across different views, on the common subspace. Generalized Multiview Analysis (GMA) [179] has been used to extend several of the above mentioned techniques to the multi-view setting. The LDA instance of GMA (GMLDA), finds projections for each view, which aim at separating the content of different classes and aligning different views of the same class. Another example of GMA is the GM Locality Preserving Projections (GMLPP), which extends the LPP [78] to multiple views. While these methods have been applied to multi-view face recognition, a large margin learning approach, named Latent Multi-task Learning Model (LMLM)[129], has recently been proposed for multi-view human action recognition. In order to lower the assumption that all the features from all the views are correlated, LMLM learns correlations between subsets of features across multiple views. This is important when some views contain, for instance, occluded body parts. Nevertheless, while these models are strictly discriminative in their formulation, our DS-GPLVM enjoys the advantages of both discriminative and generative models. Namely, while its learning aims at finding a discriminative shared manifold, due to the prior placed over the manifold, it also penalizes the manifold structures that cannot reconstruct the data in different views accurately. The latter is not accomplished by the multi-task models, which can easily lead to overfitting of their projections from different views. Furthermore, DS-GPLVM is a kernel-based method, thus being able to learn complex non-linear mappings to the shared manifold as well as deal with high-dimensional input features. This is in contrast to the multi-task models mentioned above as they learn linear projections for each task (i.e., the view).

6.3 Experiments

We evaluate the performance of the proposed DS-GPLVM on real-world images from the MultiPIE [73] dataset. We use facial images of 270 subjects displaying facial expressions of Neutral (NE), Disgust (DI), Surprise (SU), Smile (SM), Scream (SC) and Squint (SQ) captured at pan angles $-30°$, $-15°$, $0°$, $15°$ and $30°$, resulting in 1500 images per view. For each view, we chose the flash from the corresponding camera in order to have consistent illumination. The images were cropped to have an equal size of 140×150 pixels, and annotations of the locations of 68 facial landmark points, provided by [164], were used to align facial images within each pose. We evaluated the methods using three sets of features: (I) facial landmarks (the 68 landmark points), (II) full appearance features (Local Binary Patterns (LBPs) [3] extracted from the whole face image), and (III) part-based appearance features (LBPs extracted from the facial patches (of size 15×15), extracted around the facial landmarks. We used LBPs because they have been shown to perform well in the facial expression classification tasks [83]. From each aligned facial image (II) or the region (II), we extracted LBPs with radius 2, resulting in 59 bins.

We applied PCA to reduce the dimensionality of the input features, giving 20-D and 70-D input features for the sets (I) and (II)-(III), respectively. Throughout the experiments, we fix the size of the latent space of the models to five. We compared the DS-GPLVM to the state-of-the-art single-view and multi-view methods.[2] The baseline single-view methods include: 1-nearest neighbor (1-NN) classifier trained/tested on the original feature space, LDA [23], supervised Locality Preserving Projections (LPP) [78], and their kernel counterparts, the D-GPLVM [198] (with the Generalized Discriminant Analysis (GDA)-based prior) and GPLRF [227]. These are well-known methods for supervised dimensionality reduction applicable to single observation space. We also compared DS-GPLVM to the state-of-the-art methods for multi-view learning, the multi-view extensions of LDA (GMLDA), and LPP (GMLPP) [179]. For the kernel methods, we used the RBF kernel with the width parameter set using a validation procedure, as done in [198]. To report the accuracy of facial expression classification, we use classification rate, where the classification is performed using k-NN classifier ($k = 1$) for all the methods. In all our experiments, we applied the 5-fold subject-independent cross-validation procedure.

The evaluation of the models is conducted using the data from all poses for training, while testing is performed 'pose-wise', i.e., by using the data from each pose separately. The same strategy was used for evaluation of the multi-view techniques, i.e., GMLDA and GMLPP.

[2] By *single-view* we refer to the setting where only the data from a single view are used for learning the target classifier, while in the *multi-view* setting the data from all views were used to learn the classifiers.

Table 6.1: **Pose-wise Facial Expression classification.** The average classification accuracy across the views from the MultiPIE database, when three different types of features are used. The reported standard deviation is computed from the average results for each view.

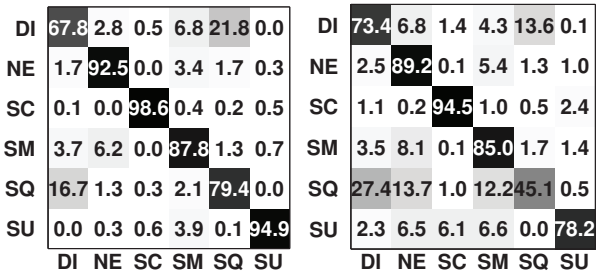
Methods	Features		
	I	II	III
kNN	77.22 ± 5.18	61.46 ± 4.09	81.25 ± 2.62
LDA	88.47 ± 8.38	72.28 ± 3.99	85.47 ± 3.07
LPP	88.40 ± 7.99	71.94 ± 4.21	85.51 ± 3.04
D-GPLVM	84.98 ± 5.48	73.64 ± 4.90	84.27 ± 2.43
GPLRF	87.58 ± 5.02	76.89 ± 4.26	86.91 ± 2.81
GMLDA	83.25 ± 6.64	70.89 ± 5.25	84.73 ± 3.09
GMLPP	80.07 ± 3.89	66.28 ± 3.62	82.03 ± 2.45
DS-GPLVM	**88.83 ± 5.30**	**77.32 ± 3.42**	**87.51 ± 2.02**

Table 6.1 summarizes the results for the three sets of features, averaged across the poses. Interestingly, LDA and LPP achieve high performance on the feature set (I). We attribute this to the fact that when points are used as the inputs, sufficiently discriminative pose-wise manifolds can be learned using the linear models. This is because the facial points of different subjects are well aligned, and subject-specific factors, that are present in the texture features, are filtered out. Furthermore, these models outperform (on average) their kernel counterparts (D-GPLVM and GPLRF), and their multi-view extensions (GMLDA and GMLPP), possibly due to the overfitting of these models. Yet, the proposed DS-GPLVM outperforms its 'single-view' counterpart (i.e., GPLRF). We ascribe this to the simultaneous learning of the shared manifold using data from all the poses, some of which may be more discriminative for the target task. This, evidently, enhances the classification performance of the DS-GPLVM across all poses. Also, DS-GPLVM performs similarly to the linear models on the feature set (I) but with significantly lower standard deviation, meaning that it achieves more consistent classification across views. When appearance-based features are used, learning of the discriminative low-dimensional manifolds is more challenging. However, the proposed DS-GPLVM achieves similar or better accuracy compared to other single- and multi-view methods due to its successful unraveling of the non-linear manifold shared across different views. Although the results of DS-GPLVM for the appearance-based features are slightly better than those obtained by GPLRF, the latter learns separate classifiers for each view, in contrast to the DS-GPLVM that uses a single classifier. Note also that the DS-GPLVM retains relatively low variance across views and the feature sets. This indicates its achieving more consistent predictions across the views.

From Table 6.1, the methods evaluated on the feature set (I) achieve slightly better results than when the feature set (III) is used. However, in the case of the feature set (I), the results obtained have significantly higher standard deviation. In the following experiments, we use the feature set (III). Table 6.2 shows the performance of the models across all poses. It is evident that in this case, the proposed DS-GPLVM performs consistently better than the

Table 6.2: **Pose-wise Facial Expression classification.** The classification accuracy for the MultiPIE dataset across the views, using the feature set (III). The reported standard deviation is computed from the results obtained on 5 folds.

Methods	Poses				
	$-30°$	$-15°$	$0°$	$15°$	$30°$
kNN	82.82 ± 0.019	82.43 ± 0.017	76.59 ± 0.034	82.06 ± 0.017	82.37 ± 0.017
LDA	86.62 ± 0.014	87.42 ± 0.015	80.03 ± 0.014	87.11 ± 0.015	86.17 ± 0.012
LPP	86.81 ± 0.014	87.35 ± 0.013	80.09 ± 0.018	86.86 ± 0.017	86.43 ± 0.011
D-GPLVM	84.67 ± 0.017	86.61 ± 0.020	80.36 ± 0.017	85.89 ± 0.019	83.86 ± 0.017
GPLRF	**87.73 ± 0.026**	88.87 ± 0.020	81.94 ± 0.025	88.16 ± 0.022	**87.83 ± 0.025**
GMLDA	86.03 ± 0.019	86.57 ± 0.016	79.23 ± 0.021	86.16 ± 0.011	85.68 ± 0.018
GMLPP	81.65 ± 0.036	84.61 ± 0.038	78.52 ± 0.034	84.14 ± 0.034	81.25 ± 0.029
DS-GPLVM	87.58 ± 0.008	**89.34 ± 0.007**	**84.12 ± 0.013**	**89.07 ± 0.006**	87.65 ± 0.009



(a) DS-GPLVM (c) Sparse

Figure 6.2: Confusion matrices for facial expression classification over three views, achieved by the (a) DS-GPLVM, (b) Sparse methods.

other models across most of the views. Note also that although GPLRF slightly outperforms DS-GPLVM in $\pm 30°$ pose, the DS-GPLVM significantly outperforms the GPLRF model in the frontal pose, which turns out to be more challenging for the expression classification than the non-frontal poses. We attribute the better performance of DS-GPLVM in this case to its ability to augment the classification in the frontal pose by using the (shared) information learned from the other views.

We next compare DS-GPLVM to the sate-of-the-art methods for multi-view facial expression classification. The results of the LGBP-based method are obtained from [136]. To compare our method with [187], we extracted dense SIFT features from the same images we used from

Table 6.3: Comparisons on the MultiPIE database. The reported standard deviation is computed from the results obtained on 5 folds.

Methods	Poses		
	$0°$	$15°$	$30°$
LGBP [136]	82.1	87.3	75.6
Sparse [187]	81.14 ± 0.009	79.25 ± 0.016	77.14 ± 0.019
CSGPR	80.44 ± 0.017	86.41 ± 0.013	83.73 ± 0.019
SC-GPR	82.79 ± 0.012	87.55 ± 0.010	85.20 ± 0.015
DS-GPLVM	**84.12 ± 0.013**	**89.07 ± 0.006**	**87.65 ± 0.009**

MultiPIE. The resulting features were then fed into the SVM classifier, as done in [187]. We also compared DS-GPLVM with the CSGPR and SC-GPR models, introduced in the previous Chapters, which perform explicit pose normalization. These models are used to perform the pose normalization of the facial points by projecting them to the best view (15°) for expression classification, followed by classification using SVMs. Table 6.3 shows comparative results. Note that the methods in [136] and [187] both fail to model correlations between different views, which results either in a huge gap between the accuracy across poses (e.g., [136]) or in a performance bounded by the one achieved in the frontal pose (e.g., [187]). The CSGPR method accounts for the relations between the poses, while SC-GP accounts for relationships in the facial shapes, through the head-pose normalization process. We see that SC-GP outperformed CSGPR in all the poses. We attribute this to the fact that the training/testing is conducted using the automatically localized facial points, in which case SC-GP was able to deal better with noise/localization errors. This, in turn, resulted in its better performance on the classification task. However, the proposed DS-GPLVM shows the performance that is similar or better than that of the rest of the methods across all the views. Also, it outperforms our approach based on explicit pose normalization. Again, we attribute this to its modeling of the shared manifold, which helps to improve expression classification in under-performing views (mostly the frontal view).

Finally, in Fig.6.2, we show the confusion matrices for different models tested using the feature set (III). The main source of confusion is caused by erroneous predictions of facial expressions of *Disgust* and *Squint*. This is because these expressions are characterized by similar changes in facial appearance around the eye region. In this case, DS-GPLVM is slightly outperformed by the Sparse method on facial expressions of *Disgust*. However, the latter method is largely outperformed by DS-GPLVM on facial expressions of *Squint*. We inspected the projections of these two expressions in the shared manifold, and found that there was an overlap between them, which is the main reason for confusion of these two expression categories by DS-GPLVM.

6.4 Conclusions

In this Chapter, we proposed the DS-GPLVM model for learning a discriminative shared manifold optimized for classification of facial expressions from multiple views. This model is a generalization of the discriminative latent variable models for single observation spaces [227, 198]. Compared to the multi-view facial expression classification methods that are based on explicit pose normalization (via the CSGPR or SC-GPR models from Chapters 4 and 5, re-

spectively), the proposed DS-GPLVM aims at classification of facial expressions from multiple views, while performing pose normalization implicitly via the shared manifold. The advantage of this approach is that it is not directly affected by inaccuracies in pose-normalization. Moreover, the adverse effects of high-dimensional noise on classification of the target expressions are reduced in the shared manifold. As evidenced by our experiments on real data from the MultiPIE dataset, discriminative modeling of the shared manifold improves the 'per-view' classification of the facial expressions, which does not take into account correlations between different views. Also, the proposed approach outperforms several state-of-the-art methods for supervised multi-view learning, as well as the pose-invariant facial expression classification methods based on pose normalization we proposed in Chapters 4 and 5.

Chapter 7

Pose-invariant Facial Expression Analysis: Conclusions and Future Work

In this part of the thesis, we addressed the problem of pose-invariant facial expression classification. The main challenge here is how to efficiently encode relationships between corresponding facial displays in multiple poses, so as to increase robustness of facial expression classification in the presence of large variation in poses, subjects, as well as of errors in facial feature extraction. To this end, we have proposed two different approaches for pose-invariant classification of facial expressions of the six basic emotions from static images. In these approaches, we achieved pose invariance using the proposed GP-based pose normalization methods to align facial features from multiple poses. The classification of the target facial expressions was then accomplished using the standard classifiers. In contrast to a small number of works addressing the target problem using independently trained pose-wise classifiers, in our methods we modeled different types of spatial structure in the facial expression data. As a result, the proposed models achieved accurate pose normalization and expression classification of the six basic emotions using a small amount of training data, while also being largely robust to corrupted image features and imbalanced examples of different facial expression categories. They also considerably outperformed existing methods for pose-invariant facial expression classification, as well as the state-of-the-art models for multi-view learning. In what follows, we discuss the proposed contributions and give directions for future research.

Chapters 4 and 5 Our first approach to pose-invariant facial expression classification is based on explicit pose normalization. In this approach, we warped facial features (i.e., locations of characteristic facial points) from non-frontal poses to the frontal pose, followed by their classification. Because of large variation in poses, and facial expressions shown by different

subjects, this required learning of highly non-linear mapping functions that are capable of preserving subtle details of target facial expressions. We showed that such functions can be learned successfully by our pose normalization models. Specifically, the Coupled GP (CGP) model, introduced in Chapter 4, achieved accurate pose normalization using a moderately small amount of training data per pose. Also, to achieve comparable classification performance, this method required far less training data in non-frontal poses than the *pose-wise* facial expression classifiers. More importantly, in contrast to *pose-wise* classifiers, the CGP-based method for pose-invariant expression classification can deal with facial expression categories that were not present in some non-frontal poses during training. This is because in CGP we modeled spatial correlations *between* different poses – something that is not modeled by the *pose-wise* classifiers. However, the CGP model does not account for relationships *within* the pose-normalized facial points, which is important for ensuring that they form a plausible facial configuration. We addressed this in the proposed Shape-conformed GP (SCGP) model in Chapter 5. Unlike existing GP regression models with structured outputs (i.e., Twin GP), which attempt learning of the output structure without taking into account domain knowledge, in our SCGP model this is encoded by means of 2D deformable face-shape models. Specifically, we placed a prior over the kernel functions and defined a novel inference approach, forcing the predictions of the SCGP model conform to a valid facial shape. We showed that this model achieves robust pose normalization in the presence of high levels of noise and outliers in facial points.

Chapter 6 Where the ultimate goal is facial expression classification, by independently minimizing the pose-normalization and classification costs, as we did in our methods mentioned above, we may be expending too much learning effort on the former, without improving the latter. We addressed this in our Discriminative Shared GP Latent Variable model (DS-GPLVM) for pose-invariant facial expression classification with implicit pose-normalization. DS-GPLVM does not require the canonical pose, as it performs classification in a manifold shared among facial expressions from multiple poses, the topology of which is optimized for classification. Another important feature of the DS-GPLVM is that it can deal effectively with facial features of arbitrary high dimension (i.e., appearance features) since it is a kernel-based method. Attempting to pose-normalize appearance features using our models for explicit pose normalization would be challenging computationally, due to the large number of the model outputs (i.e., the feature dimension), and because of the difficulty in preserving the expression details in the pose-normalized facial appearance. We showed that DS-GPLVM outperforms our methods with explicit pose-normalization, which use geometric features, and the state-of-the-art methods for multi-view facial expression classification and general multi-view learning.

Compared to the pose-wise classifiers, DS-GPLVM improves classification of facial expressions in under-performing views (i.e., frontal). This is because of its modeling of spatial correlations among multiple views via the shared manifold.

Future work. One way to improve our methods is to make the pose estimation the integral part of our approach. This is in order to more effectively handle poses in between a discrete set of poses used for training. In our methods with explicit pose normalization, and specifically the CGP model, the gating function for combining the predictions from different non-frontal poses is based on a convex combination of the pose memberships. These are estimated independently from the mapping functions. Yet, this can be achieved simultaneously by using the modeling approach of the mixture of GP experts (e.g., [158, 95, 180]). We believe that this would result in a model that is less sensitive to inaccuracies in pose estimation as well as a better pose-normalization of facial expressions from poses that were not used for training. This, in turn, could improve facial expression classification from pose-normalized facial points. Also, the SCGP model can further be improved by imposing additional constraints on the range of possible variation in the shape parameters, as in [203]. Note that the CGP model is a more suitable choice when training data are unevenly distributed across poses (as in the case of missing data), while the SCGP model should be used when data are contaminated with noise and/or outliers (e.g., due to inaccuracies in facial point localization). This is because the two models encode different types of spatial structure in data. Modeling this structure jointly would result in a pose normalization model capable of handling missing data while also being largely invariant to noise and outliers.

DS-GPLVM can also be improved. In the current model, learning of back-mappings (i.e., mappings from each pose to a shared space), and a shared manifold, is performed independently. However, to reduce overfitting, uncertainty of back-mappings should also be modeled in the manifold. In this case, the learning can be posed as a constrained optimization problem, where errors of the back-mappings from each pose can be used to form (explicit) constraints. Then, different alternating optimization techniques (e.g., [22]) can be employed to find a shared-manifold, and back-mappings simultaneously. Also, how to select the size of the shared manifold automatically is another important question. While we applied a validation procedure for this, minimizing the rank of the manifold, as in [165], may be a better solution. We also assumed that correspondences between facial images taken at different poses are known. Extending DS-GPLVM so that it can learn a shared manifold in the case where these correspondences are unknown (or partially known) would allow the model to be applied to pose-varying facial data recorded with a single camera. This can be addressed efficiently

by combining DS-GPLVM with various co-training frameworks (e.g., see [27]).

Note also that our approach with explicit pose normalization requires a canonical pose, which is usually assumed best for the classification task. DS-GPLVM alleviates this by performing classification in a shared manifold. However, in both of these methods, we assumed that there is a single representation (pose or manifold) that is optimal for classification of facial expressions. As experimentally shown in [136], and also confirmed in our experiments in Chapter 6, the best classification, on average, of six emotion categories can be achieved in 15° pose. However, it has been argued in [148] that the left hemisphere of the face is better for analysis of positive (e.g., happiness), and right for negative (e.g., anger) emotions. Thus, future research should focus on investigating emotion-specific representations (canonical poses or latent spaces). One way to address this is to extend our method with explicit pose normalization by performing emotion-specific pose-normalization to different canonical poses, followed by classification of the joint feature vectors containing pose-normalized features. A more elegant way to address the problem is to use the notion of private spaces in Shared GPs [58], to extend DS-GPLVM so that it can learn emotion-specific shared spaces. This would also allow simultaneous classification of multiple emotion categories, which commonly occur in spontaneous facial data.

While in this part of the thesis we focused on pose-invariant classification of facial expressions of the basic emotions, the proposed methods can be used for pose-invariant classification of AUs - a problem that has not been studied much so far. For instance, pose-normalized facial points, obtained with our methods for explicit pose normalization, can be used as input to the AU classifiers trained in the frontal pose. However, since some AUs cannot be detected solely from the facial points, appearance features would need to be used. In this case, the DSGPLVM is a good alternative. Still, learning the pose-invariant models for each AU independently may not be practical due to the number of possible poses and AUs. Therefore, extensions of these models that can handle multi-label classification should be investigated (as mentioned above). Note also that the proposed DSGPLVM can be used for fusion of different types of features within a pose and/or across poses. Moreover, because of its generative property, it can be used for image synthesis, and, therefore, as a basis for pose-invariant facial point localization and tracking. This can be attained by combining DSGPLVM with the dynamic GP framework in [207].

Part II

Analysis of Facial Expression Dynamics from Image Sequences

Chapter 8

Conditional Ordinal Random Fields (CORF) for Analysis of Facial Expression Dynamics

Contents

8.1 Introduction 115
8.2 Why CORF? 116
8.3 Conditional Ordinal Random Fields 117
8.4 Summary of Proposed Methods 122

8.1 Introduction

As we described in Chapter 1, facial expression dynamics are important for successful interpretation of facial expressions. There are two closely-related lines of research in analysis of facial expression dynamics. The first addresses estimation of temporal segments (neutral, onset, apex and offset) of facial expressions, and the second addresses estimation of facial expression intensity levels. In methods that we propose for temporal segmentation and intensity estimation of facial expressions, we focus on modeling of spatio-temporal structure in data, encoding (i) ordinal relationships between increasing intensity levels of facial expressions, (ii) data topology that is largely invariant to subject differences, and (iii) temporal dependencies between image frames in videos. We also account for influence of context (i.e., who the observed subject is) in which the target facial expressions have been shown. Existing works attempting to model facial expression dynamics fail to account for some or all of these

factors. For instance, the approaches proposed for temporal segmentation of facial expressions model temporal structure in data but they ignore their ordinal relationships and/or influence of context. However, as we show in our evaluation studies, modeling each of these factors (and doing this jointly) is of great importance for faithful representation of the target facial expression dynamics. To accomplish this, we base our methods on the Conditional Random Field (CRF) [108] framework, and, in particular, its adaptation to ordinal data, Conditional Ordinal Random Field (CORF) [102]. In the following, we first provide motivation for using these models as a basis for our methodology. We then describe these models and summarize the methods that we propose in this part of the thesis.

8.2 Why CORF?

The CORF model employs the best of the static ordinal regression and CRF modeling framework for sequence classification. Below we outline its key strengths that make it suitable as a basis for our approach.

- CORF imposes ordering constraints on the values of the dependent variable y. When the data labels are defined on an ordinal scale, the advantage of this is twofold. For large dimensionality of inputs, CORF has far fewer parameters to be tuned compared to standard CRF for nominal outputs. This is even more true in the case of the HCORF model. Also, when the misclassification occurs, it is more likely to be with proximal labels defined on the ordinal scale. This is in contrast to nominal CRF, which is ignorant of the ordinal structure in the labels. In our models for temporal segmentation and intensity estimation of facial expressions, and AUs, we exploit these two features of the CORF model, as both the temporal segments and intensity levels are of ordinal nature.

- The edge features in the CORF model enforce the ordinal labels at different time instances to vary smoothly, with temporally proximal labels likely to be similar. This is in contrast to static ordinal models (e.g. [37, 38]). We use this to perform dynamic modeling of the temporal segments and intensity levels of facial expressions, and AUs, from image sequences.

- Because of the probabilistic formulation of the CORF model, we can place different priors over its parameters to obtain MAP solution. We define different priors that encode discriminative information about various facial expressions, resulting in models that are largely invariant to intra- and inter-subject variability.

- The modeling approach of CORF paves the way for combining more sophisticated ordinal models, which have been thoroughly studied in the statistics community and behavioral sciences (e.g., see [214]). We explore this to obtain different kernel extensions of the model and account for the effects of heteroscedastcity in the data. The former is important in order to efficiently deal with high dimensional facial appearance features, while the latter for accounting for varying facial morphology and expressiveness levels of different subjects.

- The ordinal latent variable model in (8.4), used to define the node potentials in the CORF model, can be generalized so that it allows different factors to influence the ordinal latent variable z. We use this property to perform context-sensitive modeling of intensity of facial expressions, and AUs.

Another incentive for using the CORF model as a basis for our approach is that it has been shown in [102] that this model achieves substantially better results in the task of temporal segmentation of emotion expressions, compared to traditional models for sequence classification (standard CRF [108] and HMM [157]) and state-of-the-art static classification models for ordinal data (GPOR [37] and SVOR [38]).

8.3 Conditional Ordinal Random Fields

In this Section, we first explain the difference between modeling of ordinal and nominal categorical variables. We then give a brief introduction to the modeling framework of CRF, used to define the CORF model. This is followed by the Hidden CRFs (HCRF) [156] framework, used to define the Hidden CORF (HCORF) [101] model. We close this section by providing motivation for using the CORF model as a basis for our approach.

8.3.1 Ordinal vs. Nominal Modeling of Categorical Variables

The goal of ordinal regression is to predict the output y that indicates the ordinal score of an item represented by a feature vector $\mathbf{x} \in \mathbb{R}^{d_x}$, where the ordering of the categorical responses y is described as $y = 1 \prec y = 2 \prec \ldots \prec y = K$, with K being the number of ordinal scores. Below we introduce ordinal modeling using a latent variable approach [1, 214, 37].

Consider an underlying continuous but latent process z that is defined as

$$z = \beta^T x + \epsilon, \tag{8.1}$$

where the parameter vector β is common for all K (ordinal) classes. The random error terms ϵ are assumed to be independently and identically distributed with distribution function $\Upsilon(\epsilon)$ with zero mean and constant variance [214]. As proposed in [37], the noiseless *ordinal* likelihood can then be defined as

$$p_{ideal}(y=k|z) = \begin{cases} 1 & \text{if } z \in (b_{k-1}, b_k] \\ 0 & \text{otherwise} \end{cases}, \quad k = 1, \ldots, K \tag{8.2}$$

where $b_0 = -\infty \leq \cdots \leq b_K = \infty$ are the thresholds or cut-off points that divide the real line into K contagious intervals, thus enforcing the ordinal constraints. When $\Upsilon(\epsilon)$ is assumed to be a Gaussian distribution with zero mean and variance σ^2, the ordinal likelihood is constructed by contaminating the ideal model with noise as

$$\begin{aligned} p(y=k|z) &= \int_\epsilon P_{ideal}(y=k|z) \cdot \mathcal{N}(\epsilon; 0, \sigma^2)\, d\epsilon \\ &= \Phi(\lambda_k) - \Phi(\lambda_{k-1}), \end{aligned}, \tag{8.3}$$

where $\Phi(\lambda) = \int_{-\infty}^{\lambda} \mathcal{N}(\xi; 0, 1) d\xi$ is the normal cumulative distribution function (cdf), and $\lambda_k = \frac{b_k - \beta^T x}{\sigma}$. The noise variance σ is often set to one for identification purpose [214], thus, the parameters of the model become $\{\beta, \{b_k\}_{k=1}^{K-1}\}$. In the literature, this model is usually referred to as the ordinal threshold model [134] or ordered probit model [1].

The ordinal model in (8.3) is commonly used in the literature, however, other ordinal regression models have been proposed (e.g., see [1, 214, 94, 37]). Here we briefly describe two state-of-the-art ordinal models that are considered in our evaluation studies: Gaussian Process Ordinal Regression (GPOR) [37] and Support Vector Ordinal Regression (SVOR) [38]. Specifically, in GPOR [37] the deterministic linear component $\beta^T x$ in (8.3) is replaced with a nonlinear function $f(x)$, where a Gaussian Process prior is placed over the function f. Two learning strategies for this model have been proposed: (i) MAP Approach with Laplace Approximation, and (ii) Expectation Propagation with Variational Methods. We adopt the former approach. On the other hand, SVOR is a deterministic model, which is formulated based on (8.2) using a max-margin approach. This formulation conforms to that of standard SVM while the aim is to maximize margins at the nearby bins. Again, two learning strategies have been proposed: (i) with explicit and (ii) with implicit constraints on the model's thresholds. We use the latter approach.

In contrast to ordinal models, nominal models are ignorant of the ordering of the values of the dependent variable y. In other words, they treat each class as equally different from the rest. By following the same latent variable approach, in nominal models an underlying continuous process z_k is defined for each class $k = 1, \ldots, K$, as

$$z_k = \beta_k^T x + \epsilon_k. \tag{8.4}$$

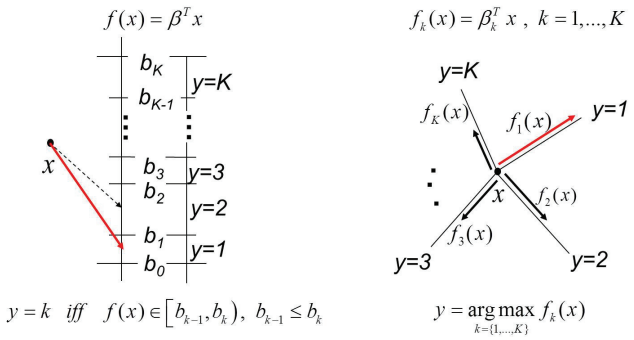


Figure 8.1: Ordinal vs. nominal modeling approach. Note that in ordinal modeling only a single projection vector (β) is learned for all classes (plus the binning parameters b), while in the case of nominal modeling independent projection vectors ($\beta_k, k = 1, \ldots, K$), are learned for each class. When the number of input feature dimensions as well as the number of classes (K) is large, this results in significantly fewer parameters to be learned by ordinal models. Also, nominal models do not impose ordinal monotonicity constraints on the output variable y. However, if the data are of ordinal nature, as, for instance, in the case of expression intensity levels this is important for preserving the structure in the model's output.

In the noiseless case, the observable *nominal* variable $y \in \{1, \ldots, K\}$ is linked to latent variable z_k as

$$p_{ideal}(y = k|z_k) = \begin{cases} 1 & \text{if } z_k = \max(z_1, \ldots, z_K) \\ 0 & \text{otherwise} \end{cases} \tag{8.5}$$

It has been shown in [70] that when $\Upsilon(\epsilon_k)$ is a type I (Gumbel) extreme value distribution, we can obtain the well-known multinomial logistic (MNL) regression model from (8.6) by computing a particular order statistic (the first, i.e., maximum) of a set of values. The *nominal* likelihood of choosing category k in this model is then defined as

$$p\,(y = k|z_k) \;=\; \frac{\exp(\beta_k^T x)}{\sum_{i=1}^{K} \exp(\beta_i^T x)}. \tag{8.6}$$

Thus, for each class k, the hyperplane β_k determines the confidence toward the class k, and the class decision is made by selecting the class with the largest likelihood. The standard SVM model [47] minimizes the hinge loss $\max(0, 1-yf(x))$, where, in the linear case, $f(x) = \beta_k^T x + b_k$ is the sum of the deterministic component in (8.4) and the class-specific bias b_k. The model parameters are then $\{\{\beta_k\}_{k=1}^K, \{b_k\}_{k=1}^K\}$.

The most critical aspect that differentiates the ordinal regression (e.g., [134, 37, 38]) from the multi-class classification (e.g., [47, 108]) is the modeling strategy: while the former learns a single projection (β), which has the same effect on the covariates of different ordinal responses, the latter learns a separate projection (β_k, $k = 1, ..., K$) for each response. When the input x is high-dimensional and/or when the number of classes K is moderately large, the ordinal models

are far more parsimonious. Specifically, the complexity of ordinal models is $\mathcal{O}(K-1+d_x)$, while of nominal models is $\mathcal{O}(K \cdot d_x)$. This is illustrated in Fig.8.1. Also, due to the ordering constraints in the ordinal models, when the misclassification occurs, it is more likely to be close to the true class in the total ordering. On the other hand, the nominal models fail to make use of proximity constraints, which often leads to less accurate predictions by these models on classes of problems where output responses are indeed of ordinal nature [214].

8.3.2 Conditional Random Fields (CRF)

Consider the following setting: we are given a sequence of labels (nominal or ordinal), $\mathbf{y} = \{y_1, \ldots, y_T\}$, and the corresponding observation $\mathbf{x} = \{x_1, \ldots, x_T\}$. CRF [108, 107] defines the conditional distribution $p(\mathbf{y}|\mathbf{x})$ as the Gibbs form clamped on the observation $\mathbf{x}$ as

$$p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) = \frac{1}{Z(\mathbf{x}; \boldsymbol{\theta})} e^{s(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta})}. \tag{8.7}$$

Here $Z(\mathbf{x}; \boldsymbol{\theta}) = \sum_{\mathbf{y} \in \mathcal{Y}} e^{s(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta})}$ is the normalizing partition function ($\mathcal{Y}$ is a set of all possible output configurations), and $\boldsymbol{\theta}$ are the parameters[1] of the *score function* (or the negative energy) that can be written as:

$$s(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta}) = \boldsymbol{\theta}^\top \boldsymbol{\Psi}(\mathbf{x}, \mathbf{y}), \tag{8.8}$$

where $\boldsymbol{\Psi}(\mathbf{x}, \mathbf{y})$ is the joint feature vector.

The choice of the output graph $G = (V, E)$ and the cliques critically affects the model's representational power and inference complexity. For example, the MNL model in (8.6) is a CRF with node cliques only. We further assume that we have either *node* cliques ($r \in V$) or *edge* cliques ($e = (r, s) \in E$), and we denote the node features by $\boldsymbol{\Psi}_r^{(V)}(\mathbf{x}, y_r)$ and the edge features by $\boldsymbol{\Psi}_e^{(E)}(\mathbf{x}, y_r, y_s)$. By letting $\boldsymbol{\theta} = \{\mathbf{v}, \mathbf{u}\}$ be the parameters for the node and edge features, respectively, the score function can then be expressed as:

$$s(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta}) = \sum_{r \in V} \mathbf{v}^\top \boldsymbol{\Psi}_r^{(V)}(\mathbf{x}, y_r) + \sum_{e=(r,s) \in E} \mathbf{u}^\top \boldsymbol{\Psi}_e^{(E)}(\mathbf{x}, y_r, y_s). \tag{8.9}$$

Although the representation in (8.9) is so general that it can subsume nearly arbitrary forms of features, in the conventional modeling practice, the node/edge features are often defined as the product of measurement features confined to cliques and the output class indicators. More specifically, denoting the measurement feature vector at node r as $\boldsymbol{\phi}(\mathbf{x}_r)$, the node feature becomes:

$$\boldsymbol{\Psi}_r^{(V)}(\mathbf{x}, y_r) = \Big[I(y_r = 1), \cdots, I(y_r = R)\Big]^\top \cdot \boldsymbol{\phi}(\mathbf{x}_r), \tag{8.10}$$

[1] For simplicity, we often drop the dependency on $\boldsymbol{\theta}$ in notations.

where $I(\cdot)$ is the indicator function that returns 1 (0) if the argument is true (false). Hence the k-th block ($k = 1, \ldots, R$) of $\mathbf{\Psi}_r^{(V)}(\mathbf{x}, y_r)$ is $\boldsymbol{\phi}(\mathbf{x}_r)$ if $y_r = k$, and the $\mathbf{0}$-vector otherwise. The edge feature is similarly defined where we typically employ the absolute difference between measurement features at adjoining nodes. Thus, $\mathbf{\Psi}_e^{(E)}(\mathbf{x}, y_r, y_s)$ is

$$\Big[I(y_r = k \ \wedge \ y_s = l)\Big]_{R\times R} \cdot \big|\boldsymbol{\phi}(\mathbf{x}_r) - \boldsymbol{\phi}(\mathbf{x}_s)\big|. \tag{8.11}$$

These feature forms are commonly used in CRFs with sequence [108] and lattice outputs [107, 205]. We call the product of parameters and the feature vectors on a clique the *(clique) potential*. For instance, $\mathbf{v}^\top \mathbf{\Psi}_r^{(V)}(\mathbf{x}, y_r)$ and $\mathbf{u}^\top \mathbf{\Psi}_e^{(E)}(\mathbf{x}, y_r, y_s)$ are the node potential and the edge potential, respectively. Hence the score function is the sum of the potentials over all cliques in the graph.

The CORF model. To deal with *ordinal* responses, while still preserving the modeling flexibility of CRFs, we seek an effective way to integrate the modeling approach of ordinal regression into the CRF framework. This is accomplished using the *ordinal* likelihood $p(y_r = k|z)$ in (8.3), and setting the potential at node r as

$$\mathbf{v}^\top \mathbf{\Psi}_r^{(V)}(\mathbf{x}, y_r) \longrightarrow \sum_{k=1}^{K} \mathrm{I}(y_r = k) \cdot log\left(\Phi\left(\frac{b_k - \beta^T x}{\sigma}\right) - \Phi\left(\frac{b_{k-1} - \beta^T x}{\sigma}\right)\right). \tag{8.12}$$

Substituting this expression into (8.9) leads to a discriminative structured output ordinal model, named the Conditional Ordinal Random Field (CORF)[102]. The CORF model imposes ordinal monotonicity constraints on the outputs y through a non-linear binning-based mapping of the inputs $\phi(x) = x$. Note that when the (unnormalized) *nominal* likelihood $p(y_r = k|z_k)$ of the MNL model is used in (8.12), we recover the standard log-linear CRF model. Both CORF and CRF use the same dynamic features defined by (8.11).

Learning of the CRF/CORF model parameters is typically accomplished by maximizing the conditional likelihood objective (8.7) using gradient-based minimization techniques (see [102] for details about gradient derivation in the CORF model). In the standard linear-chain CRF this results in a convex optimization [108], while in CORF the objective is nonlinear and non-convex [102] because of the $log - exp - sum$ term in Eq.8.12. Nevertheless, in both cases it is critical to regularize the conditional data likelihood to improve the model's performance.

8.3.3 Hidden Conditional Random Fields

The structure of graph G of the linear-chain CRF/CORF models introduced above can be extended by an additional layer representing the class label for the whole sequence, which can

take values from a set of nominal values $c = \{1, \ldots, C\}$. This allows us to simultaneously model multiple classes, where each class is now generated by a latent process described as a sequence of nominal or ordinal states given by $\mathbf{y}$. When the sequence of the latent states is not observed, $\mathbf{y}$ are usually denoted by $\mathbf{h}$. This model is called Hidden CRF (HCRF), and it has been extensively studied in computer vision [156, 209] and speech classification [74]. Below we explain the HCORF [101] model, which is defined using the modeling strategy of HCRFs [156].

To deal with multiple classes, in H-CORF C independent CORF models are combined in the joint score function defined as

$$s(c, \mathbf{x}, \mathbf{h}; \boldsymbol{\Omega}) = \sum_{j=1}^{C} I(c = j) \cdot s(\mathbf{x}, \mathbf{h}; \boldsymbol{\theta}_c), \tag{8.13}$$

where $s(\mathbf{x}, \mathbf{h}; \boldsymbol{\theta}_c)$ is defined in (8.8), and $\boldsymbol{\Omega} = \{\boldsymbol{\theta}_c\}_{c=1}^{C}$, where $\boldsymbol{\theta}_c = \{\beta_c, \mathbf{b}_c, \sigma_c, \mathbf{u}_c\}$ for $c = 1, \ldots, C$, are the parameters of HCORF. With the new score function, the joint and class conditional distributions are

$$p(c, \mathbf{h}|\mathbf{x}) = \frac{\exp(s(c, \mathbf{x}, \mathbf{h}))}{Z(\mathbf{x})}. \tag{8.14}$$

$$p(c|\mathbf{x}) = \sum_{\mathbf{h}} p(c, \mathbf{h}|\mathbf{x}) = \frac{\sum_{\mathbf{h}} \exp(s(c, \mathbf{x}, \mathbf{h}))}{Z(\mathbf{x})} \tag{8.15}$$

Evaluation of the class-conditional $p(c|\mathbf{x})$ depends on the partition function $Z(\mathbf{x}) = \sum_{c,\mathbf{h}} \exp(s(c, \mathbf{x}, \mathbf{h}))$ and the class-latent joint posteriors $p(c, h_r, h_s|\mathbf{x})$. Both can be computed from independent consideration of C individual CORFs. Details about the gradient-based optimization of the parameters, $\boldsymbol{\Omega} = \{\beta_c, \mathbf{b}_c, \sigma_c, \mathbf{u}_c\}_{c=1}^{K}$, of the HCORF model can be found in [101]. Note that an equivalent HCRF model can be obtained by using the standard log-linear CRF models for each class.

8.4 Summary of Proposed Methods

Below we summarize the proposed methods.

- In Chapter 9, we propose a method for simultaneous classification and temporal segmentation of facial expressions of six basic emotions. This method is based on the HCORF model, where a sequence of latent ordinal states (temporal segments of emotion) is assumed to generate the output class (emotion). The main limitation of HCORF is that

there is no shared information between different classes since its latent states are defined independently for each class. In our method, we improve this by introducing explicit feature mappings that are shared between different classes and their ordinal states. Furthermore, we define a prior over such mappings that is based on the graph Laplacian matrix. This allows us to easily incorporate prior knowledge about the target task, and, thus, constrain the model parameters to a plausible region in the parameter space, resulting in a model that is largely robust to differences between different subjects. We show that the proposed model outperforms both the standard generative models (HMMs) and discriminative models (H-CORF) on the target task.

- In Chapter 10, we propose a method for temporal segmentation of AUs. This method is a non-linear generalization of the standard CORF model. In particular, we kernelize the linear mapping $\beta^T x$ in the ordinal latent variable model in (8.1). This is achieved by introducing regularization directly in the ambient space of the mapping functions, in addition to the Laplacian regularization performed in the manifold space, and by applying Representer Theorem to obtain the optimal functional form for the mapping functions. This allows us to model complex relationships between high-dimensional input features x and ordinal latent variable z, used to define the node potentials in the model. We also propose the Composite Histogram Intersection kernel that automatically selects the most relevant facial regions for temporal segmentation of the target AUs. We show that this model outperforms its linear counterpart, as well as the SVM-HMM [200], the state-of-the-art model for AU temporal segmentation of AUs.

- In Chapter 11, we propose a method for intensity estimation of spontaneous facial expressions of pain. In the proposed method, we extend the kernel method introduced in Chapter 10 by relaxing its assumption of having the constant variance of the error terms in the ordinal latent variable model in (8.1). We model this variance as a (non-linear) function of the inputs x, which allows them to differently influence the location and thresholds of the ordinal node potentials. This, in turn, results in the heteroscedastic kernel model that is able to better adapt to the varying expressiveness levels of different subjects, outperforming its homoscedastic counterpart, and the traditional models for sequence classification. It also largely outperforms SVM, the state-of-the-art classifier for pain intensity estimation.

- Finally, in Chapter 12, we propose a context-sensitive method for intensity estimation of spontaneous AUs and facial expressions of pain, where the W5+ (*who*, *when*, *what*, *where*, *why* and *how*) definition of the context is used. In this method, we exploit the

context in which the target facial expressions occur in order to facilitate their intensity estimation. This is achieved by modeling the context questions: *who* (the observed subject), *how* (the changes in facial expressions), and *when* (the timing of the facial expression intensity). The context questions *who* and *how* are modeled by means of the newly introduced covariate effects, the influence of which on the expression intensity is modeled by generalizing the ordinal latent variable model in (8.1). The context question *when* is modeled in terms of temporal correlation between the intensity levels, as in the linear-chain CRF model. While in the previously introduced models we use the standard maximum-likelihood learning approach for balanced data, here we also propose a weighted softmax-margin learning from data with a skewed distribution of the intensity levels, as commonly encountered in spontaneously displayed facial expressions. We show that this model achieves substantially better performance compared to that of traditional models for sequence learning and static ordinal regression, as well as the state-of-the-art models for the target tasks.

In Chapters 9-12, we describe in detail each of the proposed contributions. The discussion and suggestions for future work are given in Chapter 13.

Chapter 9

Multi-output CORF for Classification and Temporal Segmentation of Facial Expressions of Emotions

Contents

9.1 Introduction 125

9.2 Methodology 127

9.3 Experiments 135

9.4 Conclusions 141

9.1 Introduction

Most research on automated analysis of facial expressions has so far focused on classification of prototypic facial expressions of the six basic emotions (anger, happiness, fear, surprise, sadness, and disgust) [145]. However, recent psychological studies have shown that only classifying these facial expressions into, for instance, these basic categories, is insufficient to fully understand human emotion [9]. These studies emphasize the importance of explicit analysis of temporal dynamics of facial expressions for deciphering their meaning. In spite of this, the majority of existing works classify facial expressions of emotions without explicitly modeling their underlying dynamics, which are driven by changes in their temporal segments (neutral, onset, apex, and offset). When they do attempt to model these segments, they do so independently for each emotion category. However, modeling of emotion categories and their temporal

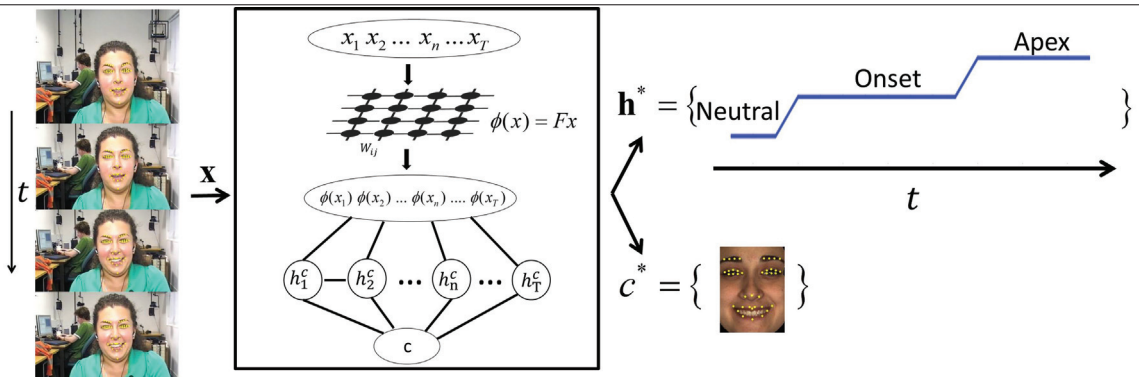


Figure 9.1: **The outline of the proposed approach.** As input **x**, we use the locations of a set of characteristic facial points extracted from each frame in an image sequence depicting facial expressions of a subject. The model maps the input features to a low-dimensional discriminative manifold defined by the explicit feature mappings $\phi(x)$, where the dynamics of different classes c (i.e., emotions) is modeled as an underlying sequence of ordinal states h, the values of which correspond to the temporal segments of emotions. The output of the model is the sequence label c^* (the most likely emotion) as well as the values of the ordinal states over time (the most likely sequence of temporal segments of the winning emotion class).

segments should be accomplished in a unified framework in order to facilitate recognition of both.

In this Chapter, we introduce an approach for simultaneous classification of facial expressions and their temporal segments that is based on the H-CORF framework [101] for ordinal modeling of image sequences. However, a limitation of HCORF is that its learning is performed directly in the measurement space, which can easily lead to overfitting or underfitting of the model parameters (e.g., due to large differences among subjects). We address this by modeling topology of the input data on a low-dimensional manifold that encodes discriminative information about different classes (i.e., emotion categories), and their ordinal states (i.e., temporal segments of emotion), while being laregly invariant to subjects' differences. We incorporate this topology into the H-CORF model by means of the newly defined explicit feature mappings, the learning of which is constrained by means of a prior based on the graph Laplacian matrix. The manifold defined by these mappings is then *jointly* estimated with the other parameters of the model. To the best of our knowledge, the proposed approach is the first that achieves simultaneous classification of facial expressions of emotions, and their temporal segments. The outline of this approach is given in Fig.9.1.

9.2 Methodology

In this section, we introduce the Laplacian Multi-output CORF model for simultaneous classification and segmentation of sequences of ordinal variables. We relate our approach to the H-CORF model introduced in Sec.8.3.3. In standard H-CORF, the ordinal variables are hidden, and thus their values are unknown during training. We also consider a fully supervised setting where the values of ordinal variables are observed during training. To distinguish this setting from standard H-CORF, we call it Multi-output CORF (M-CORF)[1]. In what follows, we first introduce the M-CORF model with explicit feature mappings. We then define a prior over such mappings using the notion of the graph Laplacian matrix. We continue by explaining how this prior is used to obtain the objective function in the proposed model. Lastly, we describe the learning and inference.

We use the following notation: $c \in \{1, ..., K\}$ denotes one of K nominal categories (i.e., the emotion class). Each nominal category c is generated by an underlying sequence of ordinal variables, $\mathbf{h} = \{h_1, \ldots, h_T\}$, where the sequence length T can vary from instance to instance. Furthermore, each variable $h_i \in \{1 < \cdots < R\}$ can take one of R different ordinal values, corresponding to the temporal segments of emotions. The input covariates $\mathbf{x} = \{x_1, \ldots, x_T\}$ are used to predict both c and $\mathbf{h}$. If not stated otherwise, we assume a fully supervised setting: we are given a training set of N data triplets $\mathcal{D} = \{(c_n, \mathbf{h}_n, \mathbf{x}_n)\}_{n=1}^N$, which are i.i.d. samples from an underlying but unknown distribution.s

9.2.1 M-CORF with Explicit Feature Mappings

In standard M-CORF, the node features of individual CORFs, one for each class, are set using the ordinal likelihood from Sec.8.3.2, where the ordinal latent variable z_k for class k, $k = 1, \ldots, K$, is defined as

$$z_k = \beta_k^T \phi(x) + \epsilon_k, \tag{9.1}$$

with $\phi(x) \equiv x$ and Gaussian distribution $\mathcal{N}(\epsilon_k; 0, \sigma_k^2)$ for the error terms. With such defined $\phi(x)$, the ordinal projections β_k are learned directly in the space of the input features x. We generalize the model in (9.1) by introducing explicit feature mapping $\phi(x) \equiv \mathbf{F}_{d_\beta \times d_x} x$, where d_β and d_x represent the size of the ordinal projection vectors β_k, and the inputs x, respectively. This leads to the new ordinal likelihood that is given by

$$p(c = k, h = r | z_k) = \Phi\left(\frac{b_{kr} - \beta_c^T \mathbf{F} x}{\sigma_k}\right) - \Phi\left(\frac{b_{kr-1} - \beta_c^T \mathbf{F} x}{\sigma_k}\right). \tag{9.2}$$

[1]Note that definition of 'multi-output' in our model relates to one output for emotion classes, and one output for their temporal segments. This is different from standard definition of multi-output models, where, e.g., the model output is a vector of class labels.

We use this ordinal likelihood to define the node features of the CORF model for class k as

$$\mathbf{\Psi}_{kr}^{(V)}(\mathbf{x}, h_r) = \sum_{l=1}^{R} I(h_r = l) \cdot log \left(\Phi(\frac{b_{kl} - \beta_k^T \mathbf{F} x_r}{\sigma_k}) - \Phi(\frac{b_{kl-1} - \beta_k^T \mathbf{F} x_r}{\sigma_k}) \right), \tag{9.3}$$

and the edge features as

$$\mathbf{\Psi}_{ke}^{(E)}(\mathbf{x}, h_r, h_s) = \Big[I(h_r = l \ \wedge \ h_s = j) \Big]_{R \times R} \cdot |\mathbf{F} x_r - \mathbf{F} x_s|, \tag{9.4}$$

Then, the score function of the k-th CORF with the explicit feature mappings is given by

$$s_k(\mathbf{x}, \mathbf{h}; \mathbf{F}, \boldsymbol{\theta}_k) = \sum_{r \in V} \mathbf{\Psi}_r^{(V)}(\mathbf{x}, h_r) + \sum_{e=(r,s) \in E} \mathbf{u}_k \, \mathbf{\Psi}_e^{(E)}(\mathbf{x}, h_r, h_s). \tag{9.5}$$

The score functions of K CORFs, one for each class, are then combined in the joint score function of the M-CORF model with explicit feature mappings as

$$s(c, \mathbf{x}, \mathbf{h}; \mathbf{\Omega}) = \sum_{k=1}^{K} I(c = k) \cdot s_k(\mathbf{x}, \mathbf{h}; \boldsymbol{\theta}_k) \tag{9.6}$$

where $\mathbf{\Omega} = \{\mathbf{F}, \{\boldsymbol{\theta}_k\}_{k=1}^{K}\}$, $\boldsymbol{\theta}_k = \{\mathbf{a}_k, \mathbf{b}_k, \sigma_k, \mathbf{u}_k\}$, are the model parameters. The joint and class conditional probabilities are obtained as in the standard M-CORF/H-CORF models, i.e.,

$$p(c, \mathbf{h}|\mathbf{x}) = \frac{\exp(s(c, \mathbf{x}, \mathbf{h}))}{Z(\mathbf{x})}, \tag{9.7}$$

$$p(c|\mathbf{x}) = \sum_{\mathbf{h}} p(c, \mathbf{h}|\mathbf{x}) = \frac{\sum_{\mathbf{h}} \exp(s(c, \mathbf{x}, \mathbf{h}))}{Z(\mathbf{x})}. \tag{9.8}$$

where $Z(\mathbf{x})$ is the normalization constant. The introduced formulation of M-CORF with explicit feature mappings allows us to perform modeling of the dynamic ordinal regression in a low-dimensional manifold ($d_\beta << d_x$), where the multiple classes are related through the parameters of the mapping function $\phi(x)$, i.e., projection matrix $\mathbf{F}$. This is in contrast to standard M-CORF where each class is treated independently from the rest. In what follows, we introduce a prior over the mapping function $\phi(x)$, which allows us to incorporate knowledge about the target task, and thus constrain the model parameters to a plausible region of the parameter space.

9.2.2 Graph Laplacian Prior

In this section, we derive a general prior over the mapping function $\phi(x)$ using the notion of graph Laplacian [40]. The basic idea is to enforce the values of the mapping function $\phi(x)$ to be a Gaussian Markov Random Field (GMRF) w.r.t. a graph constructed based on our prior

knowledge about the target task. Using this GMRF, we obtain a prior over the mapping function $\phi(\cdot)$, as explained below.

GMRF construction. Let $G = \{V, E\}$ denote an undirected fully connected graph with the node set $V = \{V_1, ..., V_N\}$, and the edge set $E = \{f(V_i, V_j) | i \neq j\}$. Each edge in the graph is associated with a weight w_{ij} that is an entry in a weight matrix W. If we further associate each node V_i with a value of the mapping function $\phi(\cdot)$ at the training examples x_i, then GMRF w.r.t. the graph G has the density:

$$p(\phi) = \prod_{q=1}^{d_\beta} p(\phi^q(\mathbf{X})) = \frac{1}{Z_\phi} \exp\left(-\frac{\beta}{2} tr(\phi(\mathbf{X})\hat{L}\phi(\mathbf{X})^T)\right) \tag{9.9}$$

where we assumed GMRF with zero mean and precision matrix given by the graph Laplacian matrix computed as $L = D - W$, where D is a diagonal matrix with $D_{ii} = \sum_j W_{ij}$. To ensure that L is positive-definite, we need to regularize its spectrum to remove the zero eigenvalues [229]. This is attained by adding a diagonal term to L, resulting in the proper prior with the precision matrix given by the *regularized Laplacian* $\hat{L} = L + \mathrm{I}/\sigma_l^2$. Furthermore, Z_ϕ is a normalization constant, $\phi(X) = [\phi_1(X) \ \ldots \ \phi_{d_\beta}(X)] \in \mathcal{R}^{d_\beta \times N_D}$, where X is the collection of N_D training examples, and $\beta > 0$ is a scaling parameter. To better understand the role of the graph Laplacian in the prior, we rewrite the exponent term in (9.9), without the regularization term, as:

$$tr(\phi(X)L\phi(X)^T) = \frac{1}{2}\sum_{i=1}^{N_D}\sum_{j=1}^{N_D} w_{ij}\|\phi(x_i) - \phi(x_j)\|_2. \tag{9.10}$$

We see that the mapping function $\phi(\cdot)$ that brings closer on the manifold the *similar* examples x, the similarity of which is encoded by the weight matrix W, is assigned a higher probability. Thus, the choice of W is critical for the target task. Typically, the elements of W are defined in an unsupervised manner using, e.g., the heat kernel as $w_{ij}^u = \exp(-\sigma_w^{-2}\|x_i - x_j\|^2)$, where σ_w is the width of the kernel, or in a supervised manner using only the label information, i.e., $w_{ij}^s = 1$ iff x_i and x_j belong to the same class, or both $w_{ij} = w_{ij}^u + \lambda w_{ij}^s$, where λ controls the level of supervision [78, 175].

Designing W for the target task. Here we show how we use the domain knowledge to construct W. The facial changes corresponding to the neutral segment should be the same for all emotions since there is no facial activity. Small changes in the facial activity are present during the onset segment, culminating during the apex segment. We encode this in the weight

matrix W, the elements of which are defined as

$$w_{ij} = \begin{cases} 1 - \frac{|h_i - h_j|}{R-1}, & \text{if } (h_i, h_j) \neq 1 \wedge c_i = c_j, \\ 1, & \text{if } (h_i, h_j) = 1, \\ 0, & \text{otherwise}, \end{cases} \tag{9.11}$$

where h_i is used to denote the ordinal levels corresponding to R temporal segments of an emotion, c is the emotion label, and (i, j) are indices running over the training instances. From (9.11), we see that similarity between the 'non-neutral' segments of the same emotion class is increased, based on the difference in their ordinal levels. By contrast, similarity between the 'non-neutral' segments of different emotion classes is set to zero. Finally, similarity between the pairs of neutral segments, regardless of their emotion class, is set to one. In this way, we attain smooth transitions between different temporal segments of emotion on a manifold, which is devised to facilitate the learning of the M-CORF parameters. Note, however, that the ordinal variables h in standard H-CORF are unobserved, so we cannot compute the weights in (9.11). In this case, we define the weights as for standard classification, and they are given by

$$w_{ij} = \begin{cases} 1 & \text{if } c_i = c_j, \\ 0, & \text{otherwise}, \end{cases} \tag{9.12}$$

Finally, the weight matrix W with the entries given in (9.11) and (9.12) is used to compute the graph Laplacian matrix, which is then used to obtain the prior in (9.9) for the mapping functions in M-CORF and H-CORF, respectively.

9.2.3 Laplacian M-CORF

In this section, we use the proposed graph Laplacian prior to define the objective function of our model, which allows arbitrary mapping functions $\phi(\cdot)$ to be used in the node and edge features of the M-CORF model. We then adapt this formulation to the M-CORF model with explicit feature mappings introduced in Sec.9.2.1.

In the Bayesian framework, the goal is to compute the posterior probability distribution over labels and select the label that has the highest probability [5]. Formally, given $\mathbf{x}$, we define the joint distribution for the class c and sequence of ordinal levels $\mathbf{h}$ as

$$p(c, \mathbf{h}|\mathcal{D}, \mathbf{x}) = \int\int p(c, \mathbf{h}|\phi(\mathbf{x}), \boldsymbol{\theta}) p(\phi, \boldsymbol{\theta}|\mathcal{D}) d\phi d\boldsymbol{\theta}, \tag{9.13}$$

where $\mathcal{D} = \{(c_n, \mathbf{h}_n, \mathbf{x}_n)\}_{n=1}^N$ is the collection of all training data, and $p(c, \mathbf{h}|\phi(\mathbf{x}), \theta)$ is the conditional probability of standard HCORF given by (9.7). In general, integrating out the

feature mappings ϕ and the parameters $\boldsymbol{\theta}$ is intractable. Approximate methods such as Monte Carlo methods [23] can be used to approximate the integral effectively. This, however, can be prohibitively expensive to use in practice. Instead, we perform a saddle-point approximation of the integral around the optimal point estimate, which is the maximum aposterior (MAP) estimate: $p(c, \mathbf{h}|\mathcal{D}, \mathbf{x}) \approx p(c, \mathbf{h}|\phi^{map}(\mathbf{x}), \theta^{map})$, where $(\phi^{\text{map}}, \boldsymbol{\theta}^{\text{map}}) = \arg\max_{\phi,\boldsymbol{\theta}} \log p(\phi, \boldsymbol{\theta}|\mathcal{D})$. By exploiting the conditional independence assumptions, we can write the posteriors ϕ^{map} and $\boldsymbol{\theta}^{\text{map}}$, up to a multiplicative constant, as

$$p(\phi, \boldsymbol{\theta}|\mathcal{D}) \propto p(c, \mathbf{h}|\mathbf{x}, \phi,)p(\phi)p(\boldsymbol{\theta}) = \prod_{i=1}^{N} p(c_i, \mathbf{h}_i|\phi(\mathbf{x}_i), \theta)\, p(\phi)p(\boldsymbol{\theta}). \tag{9.14}$$

By using the graph Laplacian prior, defined in Sec.9.2.2, for ϕ, and a flat Gaussian prior for $\boldsymbol{\theta}$, and by applying negative log function to (9.14), we arrive at the following objective of the Laplacian M-CORF model:

$$\underset{\Omega=\{\phi,\boldsymbol{\theta}\}}{\arg\min} - \sum_{i=1}^{N} \log p(c_i, \mathbf{h}_i|\phi(\mathbf{x}_i), \boldsymbol{\theta}) + \lambda_1 tr(\phi(X)\hat{L}\phi(X)^T) + \lambda_2\|\boldsymbol{\theta}\|^2, \tag{9.15}$$

where λ_1 controls the complexity of the mapping function $\phi(\cdot)$, and λ_2 controls the complexity of the ordinal regression model learned in the space defined by the mapping function $\phi(\cdot)$. The objective in (9.15) is the manifold-regularized objective of standard M-CORF, resulting in the model parameters being constrained by the domain knowledge, encoded by means of the graph Laplacian matrix.

The objective in (9.15) allows arbitrary functional forms for $\phi(\cdot)$ to be considered. Here, we focus on a linear class of models by assuming a parametric form $\phi(x) = \mathbf{F}x$, $\mathbf{F} : \mathbf{x} \to \mathcal{R}^{d_\beta}$, resulting in the objective of the (supervised) Laplacian M-CORF (LM-CORF) model

$$\underset{\Omega=\{\mathbf{F},\boldsymbol{\theta}\}}{\arg\min} - \sum_{i=1}^{N} \log p(c_i, \mathbf{h}_i|\mathbf{x}_i, \mathbf{F}, \boldsymbol{\theta}) + \frac{\lambda_1}{2} tr(\mathbf{F}X\hat{L}X^T\mathbf{F}^T) + \frac{\lambda_2}{2}\|\boldsymbol{\theta}\|^2. \tag{9.16}$$

In the case where ordinal variables $\mathbf{h}$ are unobserved, we arrive at the objective of the Laplacian H-CORF (LH-CORF) model, that is

$$\underset{\Omega=\{\mathbf{F},\boldsymbol{\theta}\}}{\arg\min} - \sum_{i=1}^{N} \log p(c_i|\mathbf{x}_i, \mathbf{F}, \boldsymbol{\theta}) + \frac{\lambda_1}{2} tr(\mathbf{F}X\hat{L}X^T\mathbf{F}^T) + \frac{\lambda_2}{2}\|\boldsymbol{\theta}\|^2, \tag{9.17}$$

where $p(c_i|\mathbf{x}_i, \mathbf{F}, \boldsymbol{\theta})$ is defined in (9.8). The objectives in (9.16) and (9.17) are used to learn the parameters of the LM-CORF and LH-CORF models, respectively. This is explained in Sec.9.2.5.

Note that although we use the linear mapping $\mathbf{F}$, it is capable of discovering a *non-linear* manifold structure of X. This is a consequence of using the graph Laplacian matrix in the objectives above, where the optimal $\mathbf{F}$ is found as a linear approximation of the otherwise *non-linear* mappings defined by the eigenfunctions of the Laplace Beltrami operator on the manifold [78].

9.2.4 Laplacian Shared-parameter M-CORF

In LM-CORF, we introduced the mapping function $\mathbf{F}$ that is shared between the CORFs corresponding to different classes. However, an independent set of parameters $\boldsymbol{\theta}_k$, $k = 1, \ldots, K$, has still to be learned for each CORF. Because the objective in (9.15) is non-convex and non-linear, this can lead to the parameters of the model getting easily trapped into a local minimum. To address this, we perform tying of the parameters. Specifically, in the standard M-CORF, independent ordinal lines with different binning parameters $\mathbf{b}$ are learned for each class. By contrast, the model with the tied parameters uses $\mathbf{b} \equiv \mathbf{b}_1 = \cdots = \mathbf{b}_K$, that are the same for all classes (see Fig.9.2). However, the ordinal projections β_k and transition matrices $\mathbf{u}_k$ remain class-specific, thus, allowing the inputs $\mathbf{x}$ to influence the outputs $(c, \mathbf{h})$ as well as their dynamics depending solely on the target class. Note that such parametrization may be too restrictive for standard M-CORF, but not for M-CORF with explicit feature mappings, where the separate binning parameters $\mathbf{b}_k$ may be redundant because of the introduced mapping $\mathbf{F}$. In the context of facial expression classification, this parameter tying is motivated by the fact that there exist a significant overlap in input features of different emotion classes, from their neutral instances being the same or very alike, followed by the increasing difference in the onset segment, with the difference culminating in the apex segment. By modeling the instances of the temporal segments of different emotions on a common ordinal line, partitioned using the same binning parameters $\mathbf{b}$, we effectively leave it to the combination of β_k and $\mathbf{F}$ to compensate for the differences, mentioned above, between the 'non-neutral' temporal segments of different emotions. We name the model with such parametrization Laplacian Shared-parameter M-CORF (LSM-CORF). The Laplacian Shared-parameter H-CORF (LSH-CORF) model is obtained analogously.

9.2.5 Learning and Inference

Here we explain learning and inference in the LSM-CORF and LSH-CORF models. The parameters of the LSM-CORF $\boldsymbol{\Omega} = \{\mathbf{F}, \{\boldsymbol{\theta}_k\}_{k=1}^K\}$, $\boldsymbol{\theta}_k = \{\mathbf{a}_k, \mathbf{b}_k, \sigma_k, \mathbf{u}_k\}$ are found as follows. The mapping function $\mathbf{F}$ is initialized by the Locality Preserving Projection (LPP) [78] method, which uses the notion of the graph Laplacian matrix to compute a linear transformation that

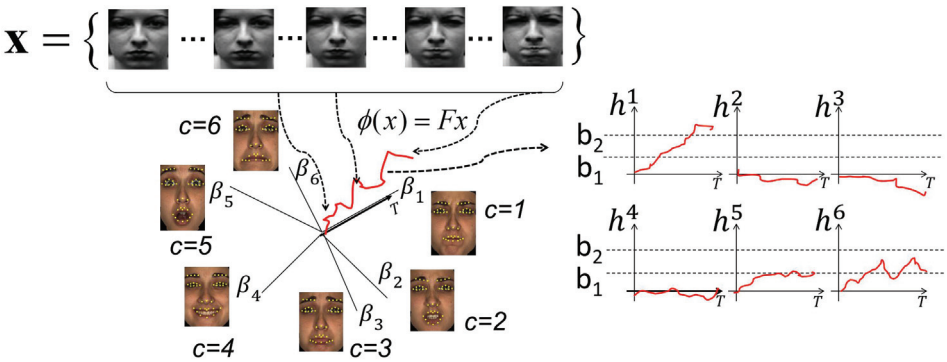


Figure 9.2: **The LSM-CORF model.** An illustration of the parameter sharing in the proposed model. The input sequence is mapped to the low-dimensional manifold using explicit feature mappings $\phi(x)$. The directions of the ordinal projections β_c, $c = 1, ..., 6$ are learned for each emotion class, directly in the data manifold of size d_β. The parameter sharing in the model is attained by using the same ordinal thresholds (b_1 and b_2) for all classes. Note that only the mapping of the input over the ordinal direction of the correct class results in its correct temporal segmentation.

maps the inputs to a low-dimensional space. The optimal solution for the transformation matrix is found by solving a generalized eigenvalue problem. As an input to this method, we provide the weight matrix W, which is constructed as explained in Sec.9.2.2. We use the optimal linear transformation found by LPP to initialize $\mathbf{F}$. The initial values for $\{\mathbf{a}_k, \mathbf{u}_k\}_{k=1}^K$ are set to zero. To enforce ordering of the parameters $\mathbf{b}$, we introduce the displacement variables δ_k, where $b_j = b_1 + \sum_{k=1}^{j-1} \delta_k^2$ for $j = 2, \ldots, R-1$. So, $\mathbf{b}$ is replaced by the unconstrained parameters $\{b_1, \delta_1, \ldots, \delta_{R-2}\}$. The standard deviation $\sigma_{k=1,\ldots,K}$ is set to one, in order to remove additional degrees of freedom in the model. The optimization of the parameters is accomplished by minimizing the objective in (9.16) using the quasi-Newton limited-BFGS (L-BFGS) method, although any unconstrained optimizer can be used. As an input to the L-BFGS, we need to provide the value of the objective function at the current parameters as well as their first order derivatives. This process is repeated until the convergence of the objective function, at which point L-BFGS returns the optimal parameters $\mathbf{\Omega}^{\text{opt}}$. Once these parameters are obtained, the inference of a test sequence $\mathbf{x}^*$ is carried out in two steps. First, we perform classification of the target sequence by applying the MAP rule to the class conditional probability in (9.8) to obtain c^*. The optimal values of the sequence of ordinal variables are found in the second step by applying Viterbi decoding to the joint probability in (9.7), i.e., $p(\mathbf{h}|c^*, \mathbf{x}^*, \Omega^{opt}) \propto p(c^*, \mathbf{h}|\mathbf{x}^*, \Omega^{opt})$, where the class label c^* from the previous step is assumed. All this is summarized in Alg.9.1. The learning and inference in LSH-CORF is done as in LSM-CORF but with a few changes to Alg.9.1. Specifically, since the ordinal variables $\mathbf{h}$ are unobserved, they are integrated out in the model. This is attained in Step 2 of Alg.9.1.

(a) Anger

(b) Disgust

(c) Fear

(d) Happiness

(e) Sadness

(f) Surprise

Figure 9.3: Example sequences of facial expressions of six emotions from the Cohn-Kanade dataset.

The inference part remains the same. The models without the explicit feature mappings and the graph Laplacian prior (M/H-CORF) and their counterparts with the tied $\mathbf{b}$ parameters (SM/SH-CORF) are optimized in a similar way, as in LSM/LSH-CORF. The difference is that the learning is preformed directly in the observation space, i.e, the projection F and the graph Laplacian prior are removed from the objectives in (9.16) and (9.17). Lastly, the choice of the regularization parameters and the size of the manifold are explained in Sec.9.3.

Algorithm 9.1 LSM-CORF

Learning

Input: $\mathcal{D} = \{(c_n, \mathbf{h}_n, \mathbf{x}_n)\}_{n=1}^N$ and Ω

1. Evaluate the objective in (9.16) and calculate the gradients w.r.t. Ω.
2. Feed the evidence and gradients to the L-BFGS method.
3. Update Ω.
4. Repeat (1-3) until convergence of the objective in (9.16).

Output: Ω^{opt}

Inference

Input: $\{\mathbf{x}^*\}$ and Ω^{opt}

1. Compute $c^* = \arg\max_c p(c|\mathbf{x}^*, \Omega^{\text{opt}})$.
2. Compute $\mathbf{h}^* = \arg\max_{\mathbf{h}} p(\mathbf{h}|c^*, \mathbf{x}^*, \Omega^{\text{opt}})$.

Output:$\{c^*, \mathbf{h}^*\}$

9.3 Experiments

In this section, we demonstrate the performance of the proposed method on the task of facial expression classification and its temporal segmentation from the frontal view facial images. For this, we use image sequences from two publicly available datasets: the BU-4DFE dataset [220] and the Cohn-Kanade (CK) dataset [116]. Both datasets contain image sequences of different subjects displaying facial expressions of six basic emotions: Anger (AN), Disgust (DI), Fear (FE), Happiness (HA), Sadness (SA) and Surprise (SU), depicted in Fig.9.3. We selected 120 image sequences of 30 subjects from the BU-4DFE dataset, and 167 image sequences of 98 subjects from the CK dataset. All image sequences start with a neutral facial expression evolving to the apex of the target emotion. Image sequences from the BU-4DFE dataset were sub-sampled, resulting in sequences that are around 20 frames long. Each image sequence was annotated in terms of six basic emotions ($c = \{1, ..., 6\}$), and each image frame was manually labeled as one of three ordinal levels corresponding to the emotions temporal segments: neutral ($h = 1$) $\prec$ onset ($h = 2$) $\prec$ apex ($h = 3$).

As input features, we used locations of a set of characteristic facial points. For the BU-4DFE dataset, we used the locations of 39 facial points obtained by applying the appearance based tracker [56], while for the CK dataset, we used the locations of 20 facial points extracted using the particle-filter-based tracker [154]. Fig.9.9 shows examples of the tracked points. These points were registered to a reference face, computed as an average face in the target dataset, and normalized by subtracting the first frame from the remaining frames in each sequence. Further processing of the input features was performed by applying Principal Component Analysis (PCA) [23], resulting in 16-D feature vectors for BU-4DFE, and 24-D feature vectors for CK, where $\sim 95\%$ of the energy was preserved.

We considered two settings in our experiments: the fully supervised setting, where the labels for both the emotion class (c) and their temporal segments (h) were used during training, and the semi-supervised setting, where only the emotion class labels were used. In the fully supervised setting, we compared the performance of the proposed LSM-CORF model with the first-order Hidden Markov Models (HMM), where for each emotion class we trained a separate HMM [23]. The states in the HMMs were observed during training and they were set using the labels for the temporal segments. In the observation model, each state was represented using a single Gaussian with a diagonal covariance matrix. During inference, the most likely HMM determined the emotion class (c), while the state-sequence of the winning HMM corresponded to the frame-based labeling of the temporal segments. We used this approach, denoted as M-HMM, as the baseline. We further compared our model with M-

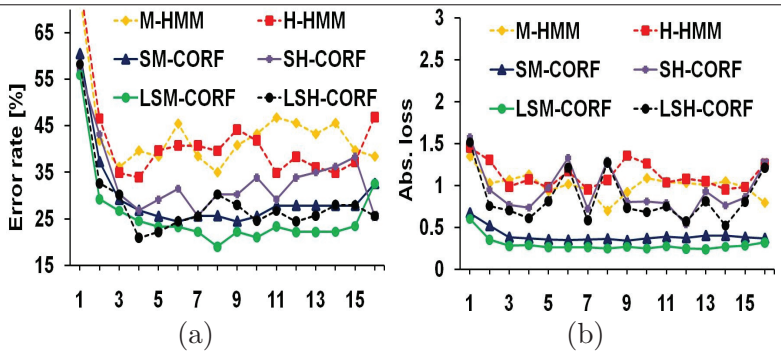


Figure 9.4: **BU-4DFE dataset.** The performance of the compared approaches w.r.t. the manifold size. (a) MER (in %) for facial expression classification and (b) MAE for their temporal segmentation.

CORF, and its counterpart with the tied parameters, i.e., SM-CORF. In the semi-supervised setting, we used the 'hidden' models (H-HMM/H-CORF/SH-CORF/LSH-CORF), in all of which the labeling of the temporal segments was unknown during training. Note that we do not include comparisons with nominal H-CRFs, since it has already been shown in [101] that standard H-CORF outperforms this model on the target task. In all our experiments, we applied a 10-fold cross-validation procedure, where each fold contained image sequences of different subjects. We report the accuracy of the models using the mean error rate for facial expression classification, which is defined as $MER = \frac{1}{N}\sum_n I(c_n \neq \bar{c}_n))$, and mean absolute error for labeling of the temporal segments, which is defined as $MAE = \frac{1}{NT}\sum_n \sum_t |h_{nt} - \bar{h}_{nt}|$. Here, (c_n, h_{nt}) and $(\bar{c}_n, \bar{h}_{nt})$ denote the predicted/ground-truth labels for emotions and their temporal segments, respectively. Due to the ordinal nature of the labels for the temporal segments, we use MAE to measure their estimation accuracy as this measure is better suited for ordinal data than MER [102, 13, 37, 32]. The regularization parameters in the model were estimated by a grid search under cross validation on the training set.

Experiments on the BU-4DFE dataset To determine the size d_β of the manifold, defined by the explicit feature mapping $\mathbf{F}$, we tested the performance of the proposed model w.r.t. different sizes of the manifold. To see if there is any benefit in simultaneous modeling of dynamic ordinal regression and the manifold, as done in LSM/LSH-CORF, we also show the performance of the other models where the LPP technique is used to find the manifold and then learn the model in such a manifold. In other words, the latter approach uses the discriminative features obtained by LPP as the input to the models. The average accuracy of the compared models is shown in Fig.9.4. Here we do not report the results for standard H/M-CORF as their performance was worse than that obtained by SH/SM-CORF models. We observe that the generative nominal classification models (H/M-HMM) are outperformed by the discriminative

Table 9.1: **BU-4DFE dataset.** The performance of the compared models per emotion class.

Method	MER (in %) for Facial Expr. classification							MAE for Temp. Segm. of Facial Expr.						
	AN	DI	FE	HA	SA	SU	Ave.	AN	DI	FE	HA	SA	SU	Ave.
M-HMM	27.0	51.4	48.6	29.2	53.1	17.5	34.0	0.74	0.67	0.95	0.34	1.15	0.27	0.69
M-CORF	33.3	33.3	55.5	16.6	38.5	5.26	26.0	1.06	0.58	1.33	0.27	1.00	0.21	0.74
SM-CORF	58.3	15.8	44.4	11.1	30.7	6.67	24.0	1.17	0.32	1.00	0.28	0.92	0.27	0.66
LSM-CORF	31.6	15.7	33.3	5.55	26.1	0.00	**19.0**	0.75	0.21	0.66	0.11	0.46	0.00	**0.36**
H-HMM	27.0	40.3	51.4	28.1	60.8	12.5	36.7	1.00	0.90	1.40	0.76	2.09	0.51	1.11
H-CORF	36.1	37.7	35.2	21.1	40.3	14.0	30.1	1.2	0.79	1.40	0.45	1.6	0.35	0.96
SH-CORF	40.0	41.6	33.3	15.7	30.7	5.55	27.8	1.2	0.75	0.77	0.26	0.84	0.16	0.66
LSH-CORF	26.6	16.6	44.4	15.7	23.1	11.1	**22.9**	0.81	0.11	1.06	0.21	0.64	0.22	**0.50**

ordinal models (CORF-based models), as expected. Furthermore, the proposed LSM/LSH-CORF models consistently showed a better performance in terms of emotion classification compared to that of SH/SM-CORF models, where the manifold is learned separately from the other parameters of the model, which clearly shows the benefit of the simultaneous modeling of the manifold as done in LSM-CORF. Also, in terms of MAE, the LSM-CORF achieved a stable performance in the manifold composed of only a few data dimensions. However, all 'hidden' models exhibited a lower and less stable performance, as expected, since the labeling of the temporal segments is learned in a fully unsupervised manner. In the remaining experiments on the BU-4DFE dataset, the dimensionality of the manifold in LSM/LSH-CORF is fixed to $d_\beta = 7$.

Table. 9.1 shows the performance of the models per emotion class. In almost all cases, the HMMs are outperformed by the ordinal models in terms of both MER and MAE. The improvements by SM/SH-CORF over standard M/H-CORF are evident from the evaluation scores. We also see that the performance of LSH/LSM-CORF can further be improved by also modeling the explicit feature mappings, and by constraining the parameter space using the graph Laplacian prior. The inclusion of the effects mentioned above increases the ability of the models to better discriminate between different emotion categories, and their temporal segments. On average, this leads to the lowest MER and MAE by LSM/LSH-CORF, with LSM-CORF performing the best. Note, however, that in the case of the facial expressions of Anger and Sadness, LSH-CORF performs better than fully supervised LSM-CORF. This indicates that the supervised temporal segmentation of the sequences, enforced by the labels used in LSM-CORF during training, may not always be optimal for modeling dynamics of these two emotions. In LSH-CORF, temporal segmentation of the target sequences is performed in an unsupervised manner that is optimal for predicting the target class only, but not for the pre-defined temporal segments. For this reason, MAE of the LSH-CORF model is sometimes significantly larger than that of the LSM-CORF model. To further demonstrate the benefits of

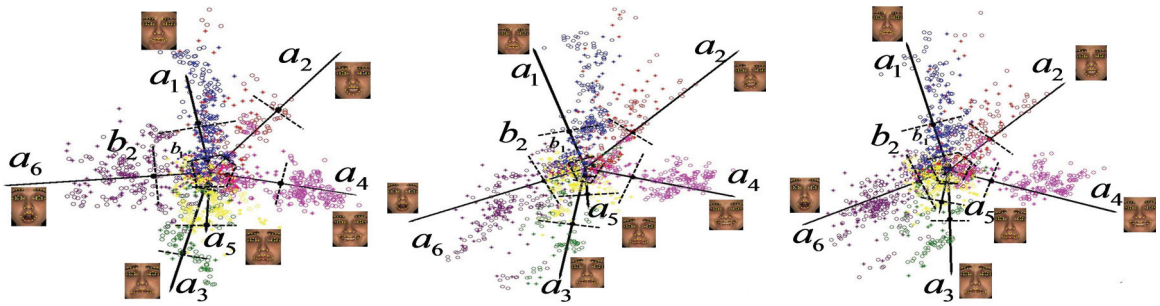
	AN	DI	FE	HA	SA	SU
AN	63.9	7.2	0.0	14.4	7.2	7.2
DI	7.5	62.3	7.5	3.7	7.5	11.3
FE	5.0	0.0	64.8	25.1	5.0	0.0
HA	1.9	1.9	5.7	78.9	1.9	9.5
SA	13.4	0.0	0.0	6.7	59.7	20.1
SU	1.7	4.3	2.6	2.6	2.6	86.0

(a) H-CORF

	AN	DI	FE	HA	SA	SU
AN	73.4	5.3	5.3	5.3	10.6	0.0
DI	8.3	83.4	0.0	8.3	0.0	0.0
FE	0.0	22.2	55.6	0.0	0.0	22.2
HA	0.0	0.0	0.0	84.3	15.7	0.0
SA	11.5	0.0	3.8	0.0	76.9	7.7
SU	0.0	0.0	11.1	0.0	0.0	88.9

(b) LSH-CORF

Figure 9.5: **BU-4DFE dataset.** Confusion matrices for the facial expression classification.



(a) it=80, 20.9% (0.14) (b) it=40, 21.5% (0.14) (c) it=80, 13.0% (0.09)

Figure 9.6: **BU-4DFE dataset.** Adaptation of the (a) SM-CORF and (b)-(c) LSM-CORF, models in a 3D manifold. In SM-CORF, the LPP obtained features remain unchanged during optimization of $\boldsymbol{\theta}$, while in LSM-CORF, $(\mathbf{F}, \boldsymbol{\theta})$ are jointly optimized. Both algorithms converged in less than 80 iterations. Below each manifold are shown MER for facial expression classification, and MAE for temporal segmentation of the facial expressions obtained after the depicted number of iterations (it). Different colors in the images depict the embeddings of facial expressions of different emotion classes, and $(\cdot, *, \circ)$ correspond to the instances of their temporal segments neutral, onset and apex, respectively.

the proposed extensions, in Fig.9.5 we compare confusion matrices of H-CORF [101], the state-of-the-art model for the target task, and the proposed LSH-CORF model. As can be seen, the latter leads to better performance in all cases except the facial expressions of Fear, which are sometimes classified by our model as facial expressions of Disgust or Surprise. A possible reason for this is that the selected size of the manifold is too small to separate successfully facial expressions of Fear from those of Disgust and Surprise.

We also observed the model adaptation in the LSM/SM-CORF models. For visualization purposes, we learned the models using 3D manifolds ($d_\beta = 3$). Fig.9.6(a) shows the SM-CORF model estimated on the 'fixed' manifold obtained by applying the LPP method to the inputs X. Fig.9.6(b)-(c) show how the topology of the manifold changes during the simultaneous estimation of the manifold ($\mathbf{F}$) and the other parameters in the LSM-CORF model. As can be seen from Fig.9.6(a), the SM-CORF model cannot recover from the initial overlap in the

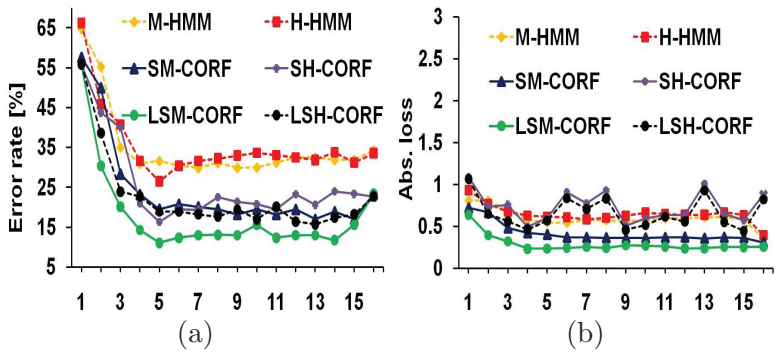


Figure 9.7: **CK dataset.** The performance of the compared approaches w.r.t. the manifold size. (a) MER (in %) for facial expression classification and (b) MAE for their temporal segmentation.

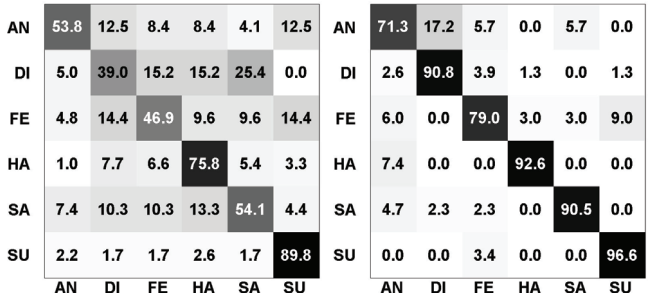
features of, for instance, facial expressions of Disgust and Happiness that are shown along the ordinal projections a_2 and a_4, respectively. On the other hand, the proposed LSM-CORF model reduces this overlap in the features by simultaneously refining the manifold structure and estimating the ordinal regression parameters. This, in turn, resulted in a significant improvement in MER and MAE scores attained by the proposed model after more iterations of the L-BFGS method.

Experiments on the CK dataset Fig.9.7 shows the performance of the compared model w.r.t. the size of the manifold, and Table 9.2 shows the performance per emotion class, where the size in the manifold of the LSM/LSH-CORF models is set to $d_\beta = 5$. Again, the proposed approach consistently outperformed the other models in both the supervised and semi-supervised settings. Note, however, that the performance of the fully supervised LSM-CORF model is slightly better than that of the LSH-CORF model, based on the average MER and MAE scores. Compared to the results obtained on BU-4DFE dataset, the average MAE score of the LSH-CORF model significantly improved compared to that of the LSM-CORF model. This can be due to inaccuracies in the annotations of the temporal segments, as well as the difference in the features used. Also, the improvement in the performance of SM-CORF/SH-CORF with the proposed parameter tying, over standard M/H-CORF is more pronounced on the CK than BU-4DFE dataset. Similar conclusions can be derived by looking at the the confusion matrices in Fig.9.8, reflecting the superior performance of our LSH-CORF model compared to standard H-CORF [101], proposed for emotion classification.

Experiments on spontaneous facial data. We also tested the proposed model on an example sequence of spontaneous facial data. For this, we recorded a subject while watching

Table 9.2: **CK dataset.** The performance of the compared models per emotion class.

Method	MER (in %) for Facial Expr. classification							MAE for Temp. Segm. of Facial Expr.						
	AN	DI	FE	HA	SA	SU	Ave.	AN	DI	FE	HA	SA	SU	Ave.
M-HMM	50.7	22.0	35.0	15.6	49.8	9.70	30.5	0.68	0.36	0.48	0.28	1.19	0.05	0.50
M-CORF	38.7	32.0	20.1	20.5	33.3	8.57	25.5	1.25	0.68	0.48	0.54	0.76	0.17	0.64
SM-CORF	35.5	16.1	24.0	2.70	42.8	2.85	20.9	0.81	0.32	0.40	0.05	1.00	0.14	0.45
LSM-CORF	23.0	12.1	8.00	2.70	23.8	2.85	**12.0**	0.75	0.16	0.16	0.05	0.67	0.14	**0.32**
H-HMM	60.0	22.0	22.0	12.7	48.1	15.7	35.8	1.06	0.16	0.20	0.05	1.00	0.20	0.45
H-CORF	46.2	61.0	53.1	24.2	45.9	10.2	40.0	1.18	1.28	0.56	0.44	0.52	0.34	0.72
SH-CORF	32.5	8.00	22.0	2.72	32.3	2.85	16.7	1.12	0.12	0.44	0.05	1.47	0.11	0.55
LSH-CORF	28.7	9.20	21.0	7.40	9.50	3.40	**13.2**	1.06	0.24	0.52	0.17	0.28	0.08	**0.39**



(a) H-CORF (b) LSH-CORF

Figure 9.8: **CK dataset.** Confusion matrices for the facial expression classification.

some humorous YouTube videos. We tracked the obtained video with two trackers, [56] and [154], which were used to obtain the features from the BU-4DFE and CK datasets, respectively. We then trained two separate LSM-CORF models using all the data from the BU-4DFE and CK datasets, respectively. Fig.9.9 shows the tracking results as well as the quantitative results for classification of facial expressions of various emotions and their temporal segments. Note that both models discriminate well between different emotions and give smooth predictions of their temporal segments. Although both models classify the test sequence as a joyful display overall, the model trained using the BU-4DFE dataset encodes high levels of Disgust. As can be seen from the bottom row in Fig.9.9, which depicts the imagery from the BU-4DFE dataset most similar to the test one, expressions similar to those depicted in the test video were labeled as Disgust in this dataset. On the other hand, the model trained on the CK dataset encodes Surprise in addition to Happiness, which is in agreement with the manual annotation of the test video that we obtained by asking three lay experts to score the video in terms of six basic emotions classes.

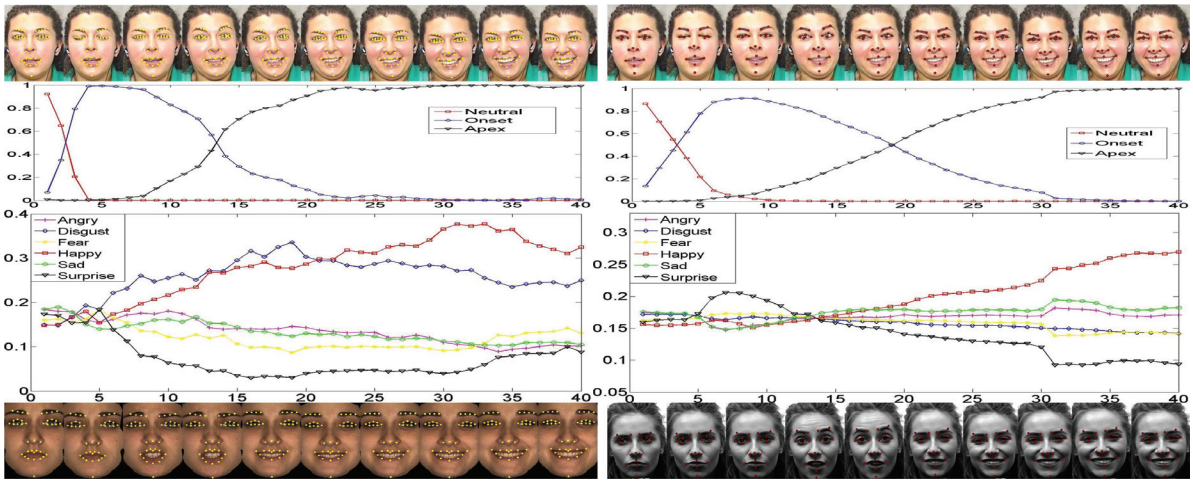


Figure 9.9: **Prediction of spontaneous facial expressions and their temporal segments.** The images shown are uniformly sampled 25% of the images from the test sequence. The graphs in between show the estimated probability of different emotions and their temporal segments, obtained by the proposed LSM-CORF model. The model was trained using data from the BU-4FE (*left*) and CK (*right*) datasets. The bottom row shows examples of the training images/features from these two datasets.

9.4 Conclusions

in this Chapter, we have proposed the Multi-output CORF model for classification of facial expressions and their temporal segments. This approach addresses the limitation of existing approaches, all of which fail to model these two tasks simultaneously. We extended the HCORF model by introducing explicit feature mappings that are shared between different classes (i.e., emotions) and their ordinal states (i.e., temporal segments of emotion). Furthermore, we defined a non-parametric prior over such mappings, which allowed us to easily incorporate prior knowledge about the target task. In this way, we constrained the feature mappings in the model to a plausible region in the parameter space represented by a low-dimensional non-linear manifold of the data. We have shown on data of facial expressions of emotion sequences that the proposed model achieves more accurate expression classification and its temporal segmentation than the state-of-the-art model for ordinal sequence modeling (H-CORF) and traditional model for sequence modeling (HMMs).

Chapter 10

Kernel CORF for Temporal Segmentation of AUs

Contents

10.1 Introduction 143
10.2 Methodology 144
10.3 Experiments 149
10.4 Conclusions 152

10.1 Introduction

Facial expression dynamics can be analyzed explicitly by detecting temporal segments (neutral, onset, apex, offset) of facial muscle actions, i.e., AUs, and, in turn, their duration, speed and co-occurrences [144]. The existing static approaches for temporal segmentation of AUs fail to exploit the temporal dependence between different segments of AUs, which is crucial for their discrimination [60]. On the other hand, the dynamic approaches are based mainly on the generative models for sequence learning (i.e., HMMs and their variants). As well as not being fully discriminative, they also fail to model the ordinal structure between the temporal segments, which is reflected in the increasing levels of the corresponding facial appearance changes (neutral $= 1 \prec$ onset, offset $= 2 \prec$ apex $= 3$). However, this spatial structure can be exploited to augment the classification of the temporal segments of AUs.

The ordinal spatio-temporal structure in temporal segments of AUs mentioned above can be modeled using the LSM-CORF model introduced in Chapter 9. Note, however, that explicit feature mappings in LSM-CORF are a linear approximation of the otherwise non-linear

mapping functions due to the use of the graph Laplacian matrix in the prior. Although such mappings have been shown effective in the task of temporal segmentation of facial expressions of emotions, they are limited by their linear form. This constrains their ability to unravel more complex relationships that may exist between input features and ordinal labels. This is especially true in the case of temporal segments of AUs, the analysis of which often involves detection of subtle changes in local facial appearance. To describe these appearance changes, high-dimensional facial appearance features are typically used. Yet, this would result in a large number of model parameters in the LSM-CORF model, which, in turn, can easily lead to overfitting. To address these limitations, in this Chapter we propose the Laplacian-regularized Kernel Conditional Ordinal Random Field (Lap-KCORF) model. This model generalizes the CORF/LSM-CORF models by introducing feature mappings that permit the use of implicit feature spaces through Mercer kernels. The resulting model can easily be applied to high-dimensional facial appearance features as well as model non-linear mappings to the ordinal space.[1]

10.2 Methodology

In this section, we first describe the Laplacian Kernel CORF (Lap-KCORF) model that is a generalization of the linear Lap-CORF model based on the general theory of functional optimization in Reproducing Kernel Hilbert Spaces (RKHS). We then introduce a kernel function for learning the relevance of the facial regions for classification of the temporal segments of AUs. Finally, we explain learning and inference in the proposed model, and its adaptation to the target task.

10.2.1 Laplacian-regularized Kernel CORF

In standard CORF, the node features are set using the ordinal likelihood from Sec.8.3.1, where the ordinal latent variable z can be defined as

$$z = f_s(x) + \epsilon, \tag{10.1}$$

with $f_s(x) = \beta^T x$. Instead of assuming the parametric form for $f_s(x)$, let us assume that it can be an arbitrary function of x. Consequently, the ordinal likelihood is given by

$$p(h = r|z) = \Phi\left(\frac{b_r - f_s(x)}{\sigma}\right) - \Phi\left(\frac{b_{r-1} - f_s(x)}{\sigma}\right), \; r = 1, \ldots, R. \tag{10.2}$$

[1]Note that in contrast to LSM-CORF, which we proposed for simultaneous classification and temporal segmentation of multiple emotion expressions, here we build separate models for each class, i.e., AU. This is because AUs are atomic facial actions that co-occur, and thus their modeling is different from that for holistic facial expressions of the basic emotions, which are assumed to occur one at a time in available datasets.

Analogously to the standard CORF model, we use this ordinal likelihood to define the node features in the CRF model as

$$\mathbf{\Psi}_r^{(V)}(\mathbf{x}, h_r) = \sum_{l=1}^{R} I(h_r = l) \cdot log \left(\Phi \left(\frac{b_l - f_s(x)}{\sigma} \right) - \Phi \left(\frac{b_{l-1} - f_s(x)}{\sigma} \right) \right), \tag{10.3}$$

and the edge features as

$$\mathbf{\Psi}_e^{(E)}(\mathbf{x}, h_r, h_s) = \Big[I(h_r = l \ \wedge \ h_s = j) \Big]_{R \times R} \cdot |f_s(x_r) - f_s(x_s)|, \tag{10.4}$$

Again, the score function of the model is

$$s(\mathbf{x}, \mathbf{h}; f_s(\cdot), \theta) = \sum_{r \in V} \mathbf{\Psi}_r^{(V)}(\mathbf{x}, h_r) + \sum_{e=(r,s) \in E} \mathbf{u}\, \mathbf{\Psi}_e^{(E)}(\mathbf{x}, h_r, h_s), \tag{10.5}$$

where $\theta = \{\mathbf{b}, \sigma, \mathbf{u}\}$, and is used to define the conditional likelihood

$$p(h|\mathbf{x}, f_s(\cdot), \theta) = \frac{\exp(s(\mathbf{x}, \mathbf{h}; f_s(\cdot), \theta))}{Z(\mathbf{x})}. \tag{10.6}$$

By applying the negative log function to the posterior $p(f_s, \theta|h, \mathbf{x}) \propto p(h|\mathbf{x}, f_s, \theta)p(\theta)p(f_s)$, we arrive at the objective function of the model that is

$$\underset{\Omega=(f_s,\theta)}{\arg\min} - \sum_{i=1}^{N} \log p(h_i|\mathbf{x}_i, f_s(\cdot), \theta) + \lambda_1 \|\theta\|^2 + \lambda_2 \mathbf{f}_s(X) \hat{L} \mathbf{f}_s(X)^T + \lambda_3 \|f_s\|_{\mathcal{H}_k}^2, \tag{10.7}$$

where in the posterior likelihood we used standard Gaussian prior for θ, and the Laplacian prior for the function f_s, similar to that used in (9.15). However, in contrast to the representation in (9.15), here we also include the L-2 kernel-inducing regularizer defined in the RKHS $\mathcal{H}_k$ associated with a kernel function k. As shown below, this allows us to find an optimal functional form for $f_s^*(\cdot)$. To this end, we first introduce Representer Theorem for conditional graphical models proposed by Lafferty et al. [109].

Representer theorem for CRFs [109]. *Let $k(\cdot,\cdot)$ be a Mercer kernel on $\mathcal{X}\,\mathbf{h}_C$ with associated RKHS norm $\|\cdot\|_{\mathcal{H}_k}$, and let $\Lambda : R_+ \to R_+$ be strictly increasing. Then the minimizer f_s^* of the regularized loss*

$$-\sum_{i=1}^{N} \mathcal{L}(h_i|f_s(g_i, \mathbf{x}_i)) + \Lambda_K \left(\|f_s\|_{\mathcal{H}_k} \right), \tag{10.8}$$

if it exists, has the form:

$$f_s^*(\cdot) = \sum_{i=1}^{N_D} \sum_{c \in C(g_i)} \sum_{h^c \in \mathcal{H}^{|c|}} \alpha_i^c(h^c) k^c(x_i, h_i^c; \cdot), \tag{10.9}$$

Here c is a clique among all the cliques of the graph g_i denoted by $C(g_i)$, and $h^c \in \mathcal{H}^{|c|}$ are all possible labellings of that clique. The key property distinguishing this result from the standard Representer Theorem for kernel machines ([103, 170]) is that the "dual parameters" $\alpha_c^{(i)}(h_c)$ now depend on all assignments of the labels [109]. By identifying the likelihood loss and regularization terms of the problem in (10.7) with those in (10.8), and since L-2 norm is strictly increasing on the required interval, the optimal functional form for f_s is given by (10.9). Note, however, that in our ordinal model, the mapping function f_s is class-independent, so we can drop dependence on the labels $\mathbf{h}$ in (10.9). Also, because of the way the edge features are defined in (10.4), only the node cliques in the graph G need be considered. Therefore, we can write the optimal functional form for f_s as

$$f_s^*(x) = \sum_{i=1}^{N_D} \alpha_i k(x, x_i), \tag{10.10}$$

where N_D is the number of the kernel bases selected from training data. By writing (10.10) in the matrix form and by plugging it in the objective function in (10.7), we arrive at the objective function of the Laplacian KCORF model

$$\underset{\Omega=(\alpha,\theta)}{\arg\min} - \sum_{i=1}^{N} \log p(h_i|\mathbf{x}_i, \alpha, \theta) + \lambda_1 \|\theta\|^2 + \lambda_2 \alpha K \hat{L} K \alpha^T + \lambda_3 \alpha K \alpha^T, \tag{10.11}$$

where $(\theta_1, \theta_2, \theta_3)$ are the parameters balancing each term in the objective. Now we explain the reason for introducing the additional regularizer $\|f_s\|^2_{\mathcal{H}_k}$. Namely, the regularizer based on the graph Laplacian is an RKHS norm in the subspace orthogonal to the null space of L, and its reproducing kernel matrix is the pseudo-inverse of L [21]. As a result of applying Representer Theorem, we would obtain a kernel expansion for f_s that is applicable to the transductive setting only. By introducing this new regularizer, Representer Theorem permits the functional form (10.10) for f_s, which can be used for inductive inference. Thus, the used regularizers combine the inductive ability and the geometry of the data domain. For further details on the role of these regularizers, see [21].

10.2.2 Composite Histogram Intersection Kernel

The functional form in (10.10) permits the use of any valid kernel function (e.g., RBF, polynomial, etc.) for encoding similarity between facial descriptors. To encode similarity between *locally* derived facial descriptors, describing variation in facial texture that corresponds to the target AU, we propose the Composite Histogram Intersection (CHI) kernel. The goal of CHI kernel is to automatically select facial regions that are relevant for classification of temporal segments of the target AU, and discard the irrelevant ones. This is important for reducing

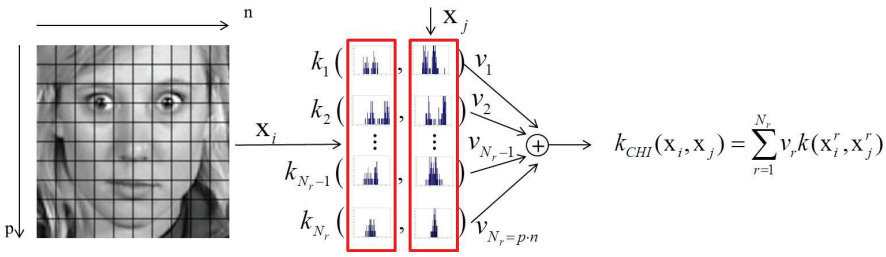


Figure 10.1: CHI kernel. Given a face image, LBP-based histograms x_i are extracted from each region in the image. Standard HI kernel is first applied to the local histograms of the input x_i and the kernel basis x_j, corresponding to the same regions in the input image (i.e., x_i^r and x_j^r). The CHI kernel is then computed as a convex combination of these local kernels. Note that since the AUs appear locally, the role of the weights v_r is to determine for each AU the regions relevant for classification of their temporal segments.

overfitting of the model. We design this kernel using the modeling strategy of the multiple-kernel learning (MKL) methods [69]. Specifically, the CHI kernel is computed as a convex combination of the base Histogram Intersection (HI) kernels [14], encoding similarity between locally extracted LBP histograms [3]. We use LBPs as they have been shown to be effective in AU detection tasks [88].

Formally, given two LBP histograms, x_i and x_j, each containing m bins, where the value of the b-th bin is x_i^b and x_j^b, respectively, the HI kernel is computed as: $\mathrm{k}(\mathrm{x}_i, \mathrm{x}_j) = \sum_{b=1}^{m} \min\left\{\mathrm{x}_i^b, \mathrm{x}_j^b\right\}$. We assume here that x_i and x_j are of the same size, i.e., $\sum_{b=1}^{m} \mathrm{x}_i^b = \sum_{b=1}^{m} \mathrm{x}_j^b$. Then, similarity between histograms extracted from $N_r = n \times p$ regions of a face in images i and j, respectively, is computed using the CHI kernel:

$$k_{chi}(\mathrm{x}_i,\mathrm{x}_j) = \sum_{r=1}^{N_r} v_r \mathrm{k}(\mathrm{x}_i^r,\mathrm{x}_j^r), \quad v_r \geq 0, \quad \sum_{r=1}^{N_r} v_r = 1, \tag{10.12}$$

where the weights v_r reflect relevance of the r-th facial region for the target task (i.e., AU segmentation). The positiveness constraint ensures that $k_{chi}(\cdot,\cdot)$ is positive definite, and the unitary constraint is necessary to avoid diverging solutions. The construction of the CHI kernel is illustrated in Fig.10.1.

10.2.3 Lap-KCORF: Learning and Inference

Learning of the Lap-KCORF parameters is accomplished as follows. To arrive at an unconstrained optimization problem, we first re-parametrize the ordinal thresholds $\mathbf{b}$ as explained in Chapter 9. We also introduce re-parameterization of the kernel parameters $\mathbf{v} = \{v_1, ..., v_{N_r}\}$ as $v_r = Z_\tau^{-1} e^{\tau_r}$, where $Z_\tau = \sum_{i=1}^{N_r} e^{\tau_i}$ is a normalization constant. The choice of the Laplacian matrix and the kernel bases is explained below. The minimization of the objective in (10.11)

w.r.t. the model parameters is then performed by using the quasi-Newton limited-BFGS method. The learning strategy proceeds as follows. Initially, the Kernel LPP (KLPP)[78] method, which is a non-linear extension of the LPP method that we used in Chapter 9 to initialize the emotion subspace, is employed to set the kernel weights α. Then, in the first minimization round, we fix edge parameters as $u = 0$ in order to form a static ordinal model that treats each node independently. After learning the node and kernel parameters $\{\alpha, \tau, b_1, \delta_1, \ldots, \delta_{R-2}\}$, we optimize the model w.r.t. u while holding the other parameters fixed. In the final step, we optimize all parameters of the model, $\Omega = \{\alpha, \tau, b_1, \delta_1, \ldots, \delta_{R-2}\}$, together. Note that, as before, we set $\sigma = 1$ for identification purpose. Once we have learned the parameters, the inference of test sequences is carried out using Viterbi decoding.

Adaptation of the KCORF to the target task. For the classification of temporal segments of an AU, we label the segments as follows: neutral ($r = 1$), onset ($r = 2$), apex ($r = 3$) and offset ($r = 4$), thus, $R = 4$ in our notation. Note that these segments also hold ordinal relationships that can be expressed as neutral $\prec$ onset, offset $\prec$ apex. This is motivated by the fact that the onset and offset segments typically reflect the same 'intensity' (ordinal level) of the target facial action. However, they do not occur at the same time in a sequence, as illustrated in Fig.10.2. To incorporate this in the Lap-KCORF model, we need to lower its assumption that all classes have different and monotonically increasing ordinal scores. This is attained by imposing the additional constraint

$$p(h = \text{onset}|f_s(x)) = p(h = \text{offset}|f_s(x)) = p(f_s(x) \in [b_0, b_1)), \tag{10.13}$$

where, ideally, we would like the ordinal projections of the onset and offset features to fall into the same bin, as they differ mostly in temporal domain. This can easily be included in the model by re-defining its node features as

$$\mathbf{\Psi}_r^{(V)}(\mathbf{x}, h_r) = \sum_{l=1}^{R} I(h_r = l) \cdot \left[\Phi(\frac{b_{l-2I(l=R)} - f_s(x_r)}{\sigma}) - \Phi(\frac{b_{l-2I(l=R)-1} - f_s(x_r)}{\sigma})\right], \tag{10.14}$$

which effectively results in one threshold parameter less in the Lap-CORF model, i.e., $\mathbf{b} = \{b_0 = -\infty, b_1, b_2, b_3 = +\infty\}$. Note that with such node features, in the static setting the Lap-KCORF model can only separate the onset/offset segments from the neutral and apex segments, but it cannot differentiate one from the other. For this, Lap-KCORF has to rely completely on its dynamic features, where the transition matrix u and the intensity of the appearance changes, measured in the ordinal space by $f_d(x_t, x_{t-1}) = f_s(x_t) - f_s(x_{t-1})$, play the key role.

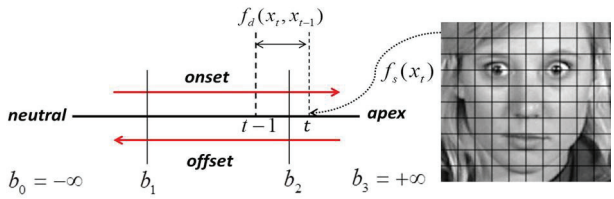


Figure 10.2: Modeling of AUs' temporal segments in the ordinal space of Lap-KCORF. Static features of the onset/offset temporal segments are often the same. However, these two segments differ in temporal domain, as the onset is usually preceded by the neutral and followed by the apex, in contrast to the offset, which is usually preceded by the apex and followed by the neutral.

Next, the graph Laplacian matrix, used in the objective function in (10.11), is computed from the weight matrix $W(L = D - W)$, the elements of which are defined as

$$w_{ij} = 1 - \frac{|h_i - 2I(h_i = R) - h_j + 2I(h_j = R)|}{R - 2}, \quad h_i, h_j = 1, ..., R. \tag{10.15}$$

Note that when the difference between the labels h_i and h_j increases, the extent of the distance enlargement (the second term in w_{ij}) in the weight matrix increases accordingly, which makes it suitable for the target task. As in the node features, here we also imposed the constraint on the ordinal levels of the onset and offset segments. This is attained by means of the indicator function $I(\cdot)$, which is used to transform the label for the offset segment ($h = 4$) to that of the onset segment ($h = 2$).

10.3 Experiments

We evaluated the proposed approach on the MMI (MMI)[151], parts I and II, facial expression dataset. Specifically, we used videos depicting facial expressions of a single AU activation, performed by different subjects. We report the results for the upper-face AUs: AU1, AU2, AU4, AU5, AU6, AU7, AU43, AU45 and AU46 (see Fig.10.3). The activation of each AU was manually coded per frame into one of four temporal segments (neutral, onset, apex or offset), and which is provided by the db creators. We refer our reader to [200] for more details about the dataset, and the AUs that we address in these experiments.

We trained the proposed Lap-KCORF model for each AU separately, using the corresponding image sequences. The parameter learning was performed as explained in Sec.10.2.3. As input features, we used 5x10 LBP histograms computed from the upper face of the aligned training images, where the alignment was attained using a set of 20 characteristic facial points and by applying an affine transform that maps these points to the reference face (i..e, the average face in the dataset). Specifically, the learned affine transform was used to map the

AU1 (Inner Brow Raiser) | AU2 (Outer Brow Raiser) | AU4 (Brow Lowerer) | AU5 (Upper Lid Raiser) | AU6 (Cheek Raiser) | AU7 (Lid Tightener) | AU43 (Eyes closed) | AU45 (Blink) | AU46 (Wink)

Figure 10.3: Examples of the upper-face AUs from the MMI dataset that we used in our experiments.

Table 10.1: F1-score for each AU.

Method	AU1	AU2	AU4	AU5	AU6	AU7	AU43	AU45	AU46	Av.
SVM-HMM[200]	**0.65**	**0.69**	0.54	0.45	**0.58**	0.34	**0.72**	**0.78**	0.29	0.56
Lap-CORF	0.54	0.52	0.48	0.38	0.45	0.38	0.53	0.66	0.51	0.49
Lap-KCORF	0.58	0.66	0.54	**0.54**	0.55	**0.60**	0.58	0.70	**0.62**	**0.60**

facial texture to the reference face. The aligned facial images were then divided into 10x10 equally sized non-overlapping regions, and the LBP histograms were extracted from the upper half (5x10) regions, resulting in a 59-D feature vector per region. The weights of the CHI kernel were initialized as $\mathrm{v}_1 = ... = \mathrm{v}_{N_r} = \frac{1}{50}$, i.e., a uniform prior was assumed. To reduce the computational cost of the Lap-KCORF model, without significantly reducing the model's performance, we used 300 kernel bases in the mapping function f_s. These bases were sampled uniformly at random from the training examples of each temporal segment.

We compared the performance of Lap-KCORF to that of its linear counterpart Lap-CORF, which is the Laplacian regularized version of the base CORF model. In Lap-CORF, we used the Laplacian prior and modification of its node features as in Lap-KCORF (see Sec.10.2.3). For Lap-CORF, the values of the 50 histograms of each image were concatenated in the 50x59-D feature vector x. Since Lap-CORF uses a linear mapping function, i.e., $f_s(x) = \beta^T x$, the learning of its parameter vector β for such high dimensional input features is intractable. For this reason, we applied PCA to reduce the dimensionality of the feature vectors, which resulted in ∼25-D vectors preserving 98% of the energy. On the other hand, for Lap-KCORF we used the full histogram features. We also show the performance of the Hybrid SVM-HMM [200] model, the state-of-the-art approach for automatic classification of AUs' temporal segments, which is based on geometric features, i.e., a set of 20 characteristic facial points. Finally, in all our experiments we applied a 5-fold cross validation procedure[2], where each fold contained image sequences of different subjects. The accuracy is reported using the F-1 measure, defined

[2]We used three folds for training, one for validation - to find the regularization parameters, and one for testing.

Table 10.2: F1-score for each temporal segment.

Method	neutral	onset	apex	offset
SVM-HMM[200]	**0.78**	0.45	0.57	0.44
Lap-CORF	0.59	0.39	0.68	0.31
Lap-KCORF	0.67	**0.48**	**0.75**	**0.49**

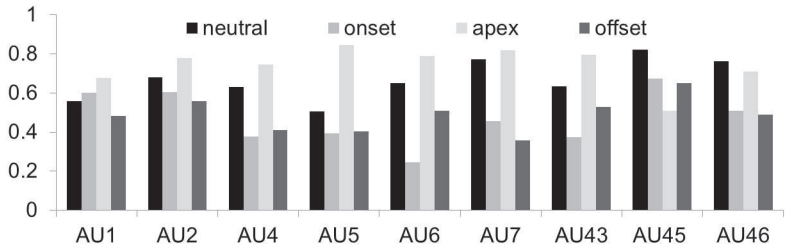


Figure 10.4: Lap-KCORF: The F1 scores for temporal segments of different AUs.

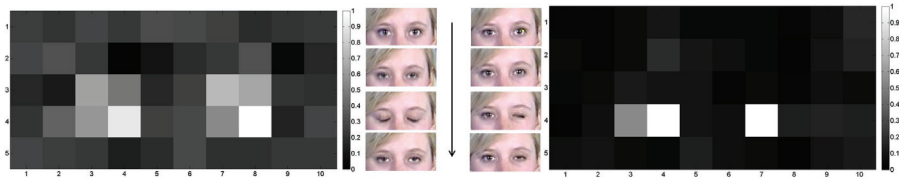


Figure 10.5: The weights of the CHI kernel learned for AU45 (*left*) and AU46 (*right*).

as $2pr/(p+r)$, where p and r represent obtained precision and recall, respectively.

Table 10.1 shows the average performance of classification of temporal segments of different AUs. The kernel models (SVM-HMM and Lap-KCORF) outperform parametric Lap-CORF on all AUs. This is because the linear approximation of the non-linear projection function in Lap-CORF is incapable of fully accounting for non-linear effects in the high-dimensional input data. Specifically, the reduction in the feature representation for this model evidently results in its being unable to fully recover the highly non-linear mapping to the ordinal space. Based on the results for each AU, in most cases SVM-HMM performs better than Lap-KCORF. On the other hand, the proposed Lap-KCORF performs better on average, mostly because it substantially improves classification of AU7 and AU46. However, by inspecting the results per temporal segment, shown in Table 10.2, we see that Lap-KCORF outperforms SVM-HMM on all temporal segments except the neutral. This indicates that Lap-KCORF is better able to model the *dynamics* of AU activations. Also, since the F1 scores per AU are obtained as average of the F1 scores for each temporal segment, it is clear that SVM-HMM achieves much higher results for the neutral, which is why the score values per AU are not always in favor

of Lap-KCORF. The superior performance of Lap-KCORF in the segment classification task is attributed to: (i) its modeling of the static ordinal constraints, which is important for the apex-segment classification, (ii) its modeling of the temporal dynamics in the ordinal space (see Sec.10.2.1), which is crucial for the model to be able to differentiate between the onset and the offset segments, and (iii) the proposed CHI kernel, which selects the most relevant features for the target task. Also, from Fig.10.4, we see that Lap-KCORF achieves improved classification of the apex of AUs in all cases except AU45. This is because only a few examples of the apex segment of this AU were available in the dataset used. Therefore, Lap-KCORF did not have sufficient support of the kernel bases for this segment, which, evidently, impaired its performance on this particular task.

Fig.10.5 depicts the learned relevance of facial regions, as measured by values of v_r in the CHI kernel, for classification of AU45 (blink) and AU46 (wink). The reason why in AU46 the most 'relevant' regions appear on both sides of the face is that we used examples of both AU46L (left wink) and AU46R (right blink) to train the model. Note that in the case of AU46, we have much sparser feature representation, as most of the learned v parameters have low values. This is because the closure of the eye in AU46, which is annotated as the apex of AU46, lasts much longer than in AU45, where the actual eye closure is rather short. Thus, the model puts more weight on the region where the eye stays closed for longer. This is why the 'weight map' of AU45 is more diffused compared to that of AU46.

10.4 Conclusions

In this Chapter, we proposed the Kernel CORF model for classification of temporal segments of AUs. This model is a non-linear generalization of the linear CORF model, achieved through implicit feature mappings defined directly in the RKHS. This allowed us to learn highly complex mappings to the ordinal space of temporal segments of AUs, as well as deal with high dimensional inout features. For this, we also proposed the Composite Histogram Intersection kernel for automatic selection of the facial regions that are most relevant for the target task. In contrast to existing models for the target task, this model accounts for ordinal relationships (neutral $\prec$ onset, offset $\prec$ apex) between the temporal segments of AUs in order to augment their classification. We showed that this model outperforms its linear counterpart, and the SVM-HMM [200], the state-of-the-art model for AU temporal segmentation of AUs.

Chapter 11

Heteroscedastic KCORF for Intensity Estimation of Facial Expressions of Pain

Contents

11.1 Introduction 153
11.2 Methodology 155
11.3 Experiments 158
11.4 Conclusions 162

11.1 Introduction

Automatic analysis of pain has received increasing attention over the last few years, mostly because of its applications in health care. For example, in intensive care units in hospitals, it has been shown recently that enormous improvements in patient outcomes can be gained from the medical staff periodically monitoring patient pain levels. However, due to the burden of work/stress that the staff are already under, this type of monitoring has been difficult to sustain, so an automatic system would be an ideal solution [124]. Recent research has evidenced the usefulness of facial cues for automated analysis of pain (e.g., see [125]), but, it has mainly focused on detection of presence/absence of pain.

A dataset named UNBC-MacMaster Shoulder Pain Expression Archive Database [125], containing video recordings of facial expressions of patients suffering from shoulder pain, has recently been released. In this dataset, the intensity of pain expression in each image frame is

Frame 27 PSPI=2 | Frame 90 PSPI=3 | Frame 149 PSPI=0 | Frame 184 PSPI=8 | Frame 234 PSPI=4 | Frame 297 PSPI=2

Figure 11.1: Examples of facial expressions of pain in an image sequence from the UNBC Shoulder Pain dataset [125]. PSPI scores, quantifying the pain intensity levels, are given below each image.

defined on an ordinal scale using the Prkachin and Solomon Pain Intensity (PSPI)[155] metric:

$$\text{PSPI} = \text{AU4} + \max\left(\text{AU6}, \text{AU7}\right) + \max\left(\text{AU9}, \text{AU10}\right) + \text{AU43}, \tag{11.1}$$

where the intensity of the AUs, defined using the A-B-C-D-E coding scheme, is used in the computation above. Thus, given a set of image sequences that are PSPI coded per frame, as those depicted in Fig.11.1, our goal is to estimate the intensity of facial expressions of pain automatically. So far, only a few approaches have been proposed for this task. All these approaches focus on the feature extraction step, while classification/regression of the target pain intensity is performed by applying the standard (static) learning techniques for nominal data (e.g., SVM and RVR), therefore ignoring the fact that pain intensity is defined on the ordinal scale. Also, none of these methods accounts for temporal pattern in intensity changes of facial expressions of pain.

In addition to the limitations mentioned above, so far the related work has not considered heteroscedasticity (changing variance levels) in facial data, which is expected in data of spontaneous facial expressions. This is because spontaneously displayed facial expressions usually cause subtle changes in facial appearance, which can vary significantly among different subjects. Heteroscedasticity can also arise due to errors in feature alignment (e.g., pose normalization), and/or model misspecification. The latter relates to possibly wrong assumptions made in a model (e.g. assumptions about noise, the order of temporal dependence in data, etc.). Furthermore, wrong or inconsistent annotations of facial expressions can be a source of heteroscedasticity. In the case of pain intensity, this can be even more pronounced as the PSPI scores are obtained as a non-linear function of manually annotated intensity levels of multiple AUs. Therefore, by accounting for different sources of heteroscedasticity in data, we can obtain more flexible and robust models for the target task. To this end, in this Chapter we propose the Heteroscedastic Conditional Ordinal Random Field model for intensity estimation of facial expressions of pain. This model generalizes the CORF framework for modeling sequences of

ordinal variables by adapting it for heteroscedasticity. Although the CORF model, and the KCORF model, proposed in Chapter 10, can address the limitations of the existing methods for pain intensity estimation, by performing temporal and ordinal modeling, their underlying assumption is that the noise on the ordinal targets (in our case, pain intensity) is constant. However, to account for heteroscedasticity in data, we need to relax this assumption. This is attained by allowing the variance of ordinal feature functions in the model to change depending on input. As we show in our experiments on the UNBC Shoulder Pain Database, this is important for improving intensity estimation of spontaneously displayed facial expressions of pain. Lastly, note that the proposed heteroscedastic CORF model is a preliminary version of our fully context sensitive model proposed in Chapter 12, where we model subject variability in data by also allowing the subject-specific biases to influence the model parameters.

11.2 Methodology

In this section, we first show how heteroscedasticity in data can be incorporated into the KCORF model introduced in Chapter 10, resulting in the Heteroscedastic KCORF model. We then explain learning and inference in the proposed model.

11.2.1 Heteroscedastic KCORF

So far, we have assumed that the error terms in the ordinal latent variable model in (10.1) are constant. We extend this model by allowing its variance to depend on the inputs. Formally, the heteroscedastic ordinal latent variable model is given by

$$z = f_s(x) + \epsilon(x), \tag{11.2}$$

where we assume independent, normally distributed noise terms $\mathcal{N}(\epsilon(x); 0, \sigma(x))$, where the noise variances $\sigma(x)$ are modeled by $\sigma(x) = \exp(g_s(x))$, i.e., as a function of x. We use $\exp(\cdot)$ to enforce the positivity of $\sigma(x)$. The choice of the function $g_s(\cdot)$ is explained below. Consequently, the heteroscedastic ordinal likelihood is given by

$$p(h = r|z) = \Phi\left(\frac{b_r - f_s(x)}{\exp(g_s(x))}\right) - \Phi\left(\frac{b_{r-1} - f_s(x)}{\exp(g_s(x))}\right), \; r = 1, \ldots, R. \tag{11.3}$$

Analogously to the standard CORF model, we use this ordinal likelihood to define the node features in the heteroscedastic CORF model as

$$\mathbf{\Psi}_r^{(V)}(\mathbf{x}, h_r) = \sum_{l=1}^{R} I(h_r = l) \cdot log\left(\Phi\left(\frac{b_l - f_s(x_r)}{\exp(g_s(x_r))}\right) - \Phi\left(\frac{b_{l-1} - f_s(x_r)}{\exp(g_s(x_r))}\right)\right). \tag{11.4}$$

The edge features are now defined in the input space, i.e., as

$$\mathbf{\Psi}_e^{(E)}(\mathbf{x}, h_r, h_s) = \Big[I(h_r = l \ \wedge \ h_s = j)\Big]_{R\times R} \cdot |x_r - x_s|, \tag{11.5}$$

since the difference in the ordinal projections $|f_s(x_r) - f_s(x_s)|$, as given by (10.4), may not be suitable because of the varying scale $\sigma(x)$ in the static model. With such defined feature functions, the score function of the model is given by

$$s(\mathbf{x}, \mathbf{h}; f_s(\cdot), g_s(\cdot), \theta) = \sum_{r\in V} \mathbf{\Psi}_r^{(V)}(\mathbf{x}, h_r) + \sum_{e=(r,s)\in E} \mathbf{u}\, \mathbf{\Psi}_e^{(E)}(\mathbf{x}, h_r, h_s), \tag{11.6}$$

where $\theta = \{\mathbf{b}, \sigma, \mathbf{u}\}$, and is used to define the conditional likelihood

$$p(h|\mathbf{x}, f_s(\cdot), g_s(\cdot), \theta) = \frac{\exp(s(\mathbf{x}, \mathbf{h}; f_s(\cdot), g_s(\cdot), \theta))}{Z(\mathbf{x})}. \tag{11.7}$$

We arrive at the objective function of the heteroscedastic model by applying the negative log function to the posterior $p(f_s, g_s, \theta|h, \mathbf{x}) \propto p(h|\mathbf{x}, f_s, \theta)p(\theta)p(f_s)p(g_s)$, that is

$$\underset{\Omega=(f_s,g_s,\theta)}{\arg\min} -\sum_{i=1}^{N} \log p(h_i|\mathbf{x}_i, f_s(\cdot), g_s(\cdot), \theta) + \lambda_1\|\theta\|^2 + \lambda_2 \mathbf{f}_s(X)\hat{L}\mathbf{f}_s(X)^T + \lambda_3 \|f_s\|_{H_k}^2 + \lambda_4 \|g_s\|_{H_k}^2 . \tag{11.8}$$

In contrast to the objective of the homoscedastic model in (10.7), here we also include the L-2 kernel-inducing regularizer for g_s. We again use the graph Laplacian term in the objective function, although this penalty may now be conflicting with the log-likelihood under the sum. This is because of the scaling term $\exp(g_s(x))$ in (11.3), which affects both the thresholds $\mathbf{b}$ and the locations $f_s(x)$ in the model. Nevertheless, we leave it in the model because the learning can be fragile due to the division $f_s(x)/exp(g_s(x))$ in the node features, as asserted in [214]. On the other hand, this regularizer is useful when there is no heteroscedasticity in data, in case of which it should drive the variance to a constant, otherwise, $\lambda_2 \to 0$ as a result of the validation procedure determining the balance parameters $\{\lambda_i\}_{i=1}^4$.

The optimal functional forms for f_s and g_s are obtained as a result of applying Representer Theorem to the objective function in (11.8). This leads to

$$f_s^*(x) = \sum_{i=1}^{N_D} \alpha_i k(x, x_i), \tag{11.9}$$

and

$$g_s^*(x) = \sum_{i=1}^{N_D} \rho_i k(x, x_i), \tag{11.10}$$

where N_D is the number of kernel bases, and we used the same kernel for the location f_s and scale g_s models, although different kernels are permitted. The objective function of the Heteroscedastic Kernel CORF model can now be written as

$$\underset{\Omega=(\alpha,\rho,\theta)}{\arg\min} -\sum_{i=1}^{N} \log p(h_i|\mathbf{x}_i,\alpha,\rho,\theta) + \lambda_1\|\theta\|^2 + \lambda_2\alpha K\hat{L}K\alpha^T + \lambda_3\alpha K\alpha^T + \lambda_4\rho K\rho^T, \quad (11.11)$$

The most important aspect of using the varying scale $\sigma(x)$ is that the inputs x can now directly influence the locations of the thresholds $\mathbf{b}$ in the model, which remain constant in the homoscedastic KCORF model. As a result, the proposed heteroscedastic KCORF model can automatically adapt its thresholds to account for, e.g., the subject's differences in pain tolerance and/or facial expressiveness.

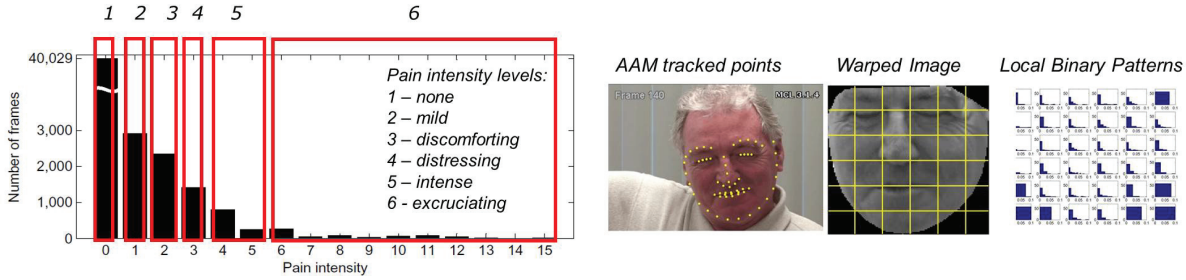
11.2.2 Learning and Inference

Learning of the model parameter is performed similarly to that in the KCORF model. Specifically, to form the kernel matrix K, we apply CHI kernel, introduced in Chapter 10, to the LBP features extracted from facial regions, as explained below. However, here we do not optimize w.r.t. the weights of this kernel in order to limit the number of parameters. For this, we set weights of the CHI kernel to $1/n$, where n is the number of face regions. Furthermore, we compute the graph Laplacian matrix based on the weight matrix W with the elements defined as:

$$w_{ij} = 1 - \frac{|h_i - h_j|}{R-1}, \; h_i, h_j = 1, ..., R. \quad (11.12)$$

The suitability of such weights for the target task has been discussed in Chapter 10. Note also that, in contrast to the problem of AU temporal segmentation, here we do not need any adaptation of the node features in the model since the intensity levels are all ordinal.

We now briefly describe the learning strategy. Initially, we set the scale model σ to 1 (i.e., $\rho = 0$), and the transition parameters $\mathbf{u} = 0$ to form a static homoscedastic model. This is accomplished by first optimizing the parameters α of the location model in (11.9), and the ordinal thresholds $\mathbf{b}$ in (11.3), by applying the quasi-Newton limited-memory BFGS method to the cost in (11.8). The kernel parameters α were initialized to $1/N_d$, where N_d is the number of the kernel bases used. The selection of the kernel bases is explained in Sec.11.3. The initialization of $\mathbf{b}$ is explained in detail in Sec.9.2.5. In the next step, we optimize $(\alpha, \mathbf{b}, \rho)$ all together, while still keeping $\mathbf{u} = 0$. Finally, we optimize all the parameters in the model simultaneously. Once the parameters of the model were estimated, the inference of test sequences was carried out using Viterbi decoding.



(a) (b)

Figure 11.2: a) Distribution of the pain intensity levels in The ShoulderPain[125] dataset , b) The feature extraction.

11.3 Experiments

We conducted experiments on The ShoulderPain dataset [125] introduced in Sec.11.1. 200 sequences of 25 subjects were recorded (48,398 frames in total). For each frame, discrete pain intensities (0-15) according to Prkachin and Solomon [155] are provided by the database creators (see Fig.11.2(a)). The image sequences with the intensity of pain expression greater than 0 were pre-segmented, so that the number of frames with the intensity 0, the most frequent in the dataset, was balanced with the second most frequent intensity. The resulting intensity distribution was still highly imbalanced, so we discretized it into 6 pain levels as: 0 (none), 1 (mild), 2 (discomforting), 3 (distressing), 4-5 (intense), and 6-15 (excruciating). This data balancing was performed in order to avoid the model being biased towards the majority classes. For our experiments, we selected 147 image sequences from 22 subjects. Image sequences of 10 subjects were for training, and the rest for testing.

To obtain the input features, we first aligned the facial images using a piece-wise affine warp based on the 66 points of the AAM provided by the database creators (see [125, 92] for details). The aligned images were then divided into 6x6 even regions to preserve local texture information. From each region, we extracted Local Binary Patterns (LBP) [139] with radius 2, resulting in 59 histogram bins per patch. This is illustrated in Fig.11.2(b). We used LBPs as the input features since they have been shown to perform well for the facial affect data (e.g., see [161, 92]).

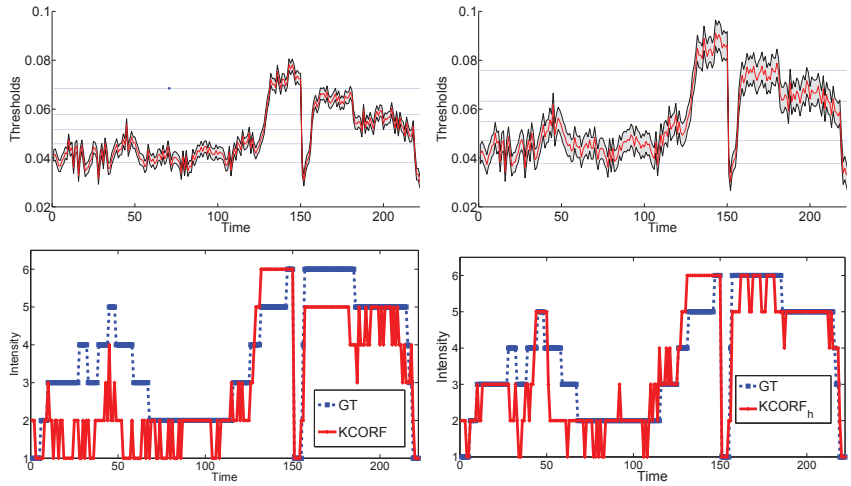
We compare the proposed heteroscedastic (kernel) CORF (KCORF$_h$) model with its homoscedastic counterpart, KCORF, which we proposed in Chapter 10 for AU temporal segmentation. We used 150 kernel bases for the location and scale models. Their selection was performed by sampling 25 kernel bases from each pain intensity level at random. We found that this is a good trade-off between the performance and computational complexity of the

models. Using a small number of kernel bases also helped to reduce the overfitting. For both the kernel methods, we used the Composite Histogram Intersection (CHI) kernel, introduced in Chapter 10. The balancing trade-off between the regularization and the log-likelihood terms was estimated by grid search under a cross validation on the training data.

As a baseline model, we use one-vs-all SVM [33], since most of the previous work on pain intensity estimation is based on this classifier. We also performed comparisons with the state-of-the-art static ordinal regression models, Support Vector Ordinal Regression with implicit constraints (SVOR) [38] and Gaussian Process Ordinal Regression [37]. For the kernel methods, we use the same kernel function as explained above. Finally, we performed comparisons with the base models for sequential data: Gaussian Hidden Markov Models (GHMM)[137] and linear-chain Conditional Random fields (CRFs) [108]. For the GHMM, each pain intensity level was treated as a state in the model, parametrized using a single Gaussian. We also included comparisons with the (linear) Laplacian-regularized CORF model. Because learning in the linear models (GHMM/CRF/CORF) is intractable due to the high dimensionality of the input features, we applied different dimensionality reduction techniques. The reported results are the best obtained, and they were achieved with 6D features derived using Kernel Locality Preserving Projections (KLPP) [78]. The performance of the models is reported using: (i) average F-1 computed from F-1 scores for each pain intensity, (ii) mean absolute error (MAE), computed between actual and predicted pain intensities, and (iii) Intra-Class Correlation (ICC) [181]. This score is commonly used to quantify agreement/consistency between different raters, and is a measure of correlation or conformity for data with multiple targets [130, 133, 181]. Depending on how the ratings are obtained, different types of this score should be used (see [181] for details). We use the ICC(3,1) model that is based on a Mixed Model ANOVA, where J judges are treated as fixed effects, and N targets are considered random effects. In our case, $J = 2$ (the true and predicted values), and N is the total number of test examples. Then, ICC is computed as

$$ICC = \frac{BMS - EMS}{BMS + (J-1)EMS}$$

where $BMS = \frac{BSS}{N-1}$ is between-class mean squares and $EMS = \frac{ESS}{(J-1)\times(N-1)}$ is residual mean squares. BSS and $ESS = WSS - RSS$ are defined as between target sum squares and residual sum of squares, while WSS and RSS are within-target and between raters sum squares, respectively. The ICC defined above measures consistency between raters, and, much like Pearson's Correlation, is insensitive to mean bias in ratings. Yet, unlike Pearson's Correlation, it is sensitive to scale of the ratings. This score is a *similarity* measure ranging from 0 to 100 (in %), but sometimes negative values can occur [181].



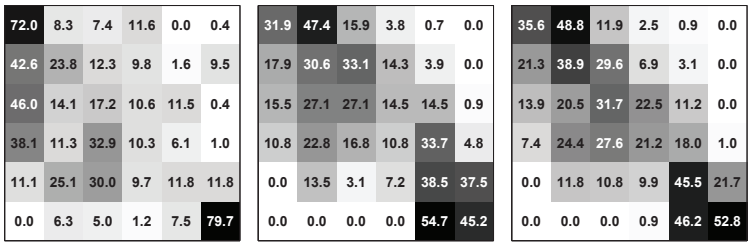
(a) KCORF (b) KCORF$_h$

Figure 11.3: Comparison of the: (a) homoscedastic and (b) heteroscedastic KCORF models (both models use the same dynamic features). The upper row shows the values of the latent variable z^* across time, where the horizontal lines are the learned thresholds. The estimated variance is also shown on z^*. The *Time* represents the frame number, where we concatenated two sequences of two test subjects (1-150 / subject 1, 151-222 / subject 2). Note the change in variance of KCORF$_h$ for the two subjects. The bottom row shows the intensity prediction by the two models against the ground-truth (GT).

Methods	SVM	SVOR	GPOR	GHMM	CRF	CORF	KCORF	KCORF$_h$
F-1 [%]	31.1	33.9	34.1	24.8	34.7	35.5	36.8	**40.2**
MAE	1.25	1.10	1.07	1.30	1.22	0.92	0.88	**0.80**
ICC [%]	46.5	57.1	57.8	39.4	49.0	63.2	66.5	**70.3**

Table 11.1: The performance of different methods applied to the task of automatic pain intensity estimation. The features for the linear models (GHMM/CRF/CORF) were pre-processed using KLPP[78].

Fig.11.3 shows the latent variable learned in the homoscedastic KCORF and the proposed heteroscedastic KCORF$_h$ model. Note the variance changes across time in the heteroscedastic model. This is especially true when switching between the subjects. This change in variance helps to adjust the locations of the intensity thresholds in the heteroscedastic ordinal model depending on the input (e.g., the test subject). Therefore, the model scales its threshold and location parameters based on the pain expressiveness level of the target subject. From Fig.11.3, it is evident that this adaptation helps to improve estimation of the pain intensity levels, especially of the higher intensity levels. For example, around frame 50, the heteroscedastic model correctly estimates level 5, in contrast to the homoscedastic model. Also, the heteroscedatic model provides smoother predictions than those by the homoscedastic model. Since both models use the same dynamic features, we attribute this to the heteroscedastic component in KCORF$_h$.



(a) SVM (b) KCORF (c) KCORF$_h$

Figure 11.4: Confusion matrices obtained using different models. We include the results attained by the SVM, which serves as the baseline model.

Table 11.1 shows the performance of different methods applied to the target task. First, note that all methods attain low F-1 scores. This is expected because the large variation in facial appearance of different subjects poses a serious challenge for any classifier. We checked the training results of the methods evaluated, and found that they all attained significantly higher F1 values. This overfitting of the models is ascribed to the fact that subject-specific variation in the features used dominates the pain-level-specific variation. We next examine how far off are the predictions from true labels. This is reflected in the MAE loss by the models. Note that the standard classification methods (SVM/GHMM/CRF) incur the highest loss, followed by the static ordinal regression models (SVOR/GPOR). Improved results are attained by the dynamic ordinal models, i.e., KCORF and KCORF$_h$, with the latter performing the best. This evidences that both the ordinal and temporal modeling contribute to improving the pain intensity estimation. Furthermore, accounting for heteroscedasticity in the data further helps to enhancing the estimation performance. The same conclusions can be drawn from the ICC scores for the models. However, it is important to mention that the ICC used here is insensitive to bias in the predictions, in contrast to MAE. Nevertheless, the scores obtained reveal that the ordinal models exhibit better conformity between the predictions and the labels, with the proposed model achieving the highest score. To further analyze the performance of the models, in Fig.11.4 we plot confusion matrices for the SVM, KCORF and KCORF$_h$ models. Note that the misclassification by the ordinal models, in contrast to SVM, occurs mostly at the neighboring intensity levels, which explains their high ICC scores and low MAE. The low performance by SVM is a consequence of treating the output variables as static and nominal. From Fig.11.4(a), it is also evident that SVM fails to differentiate well between the intermediate intensity levels, as opposed to the ordinal models. Compared to KCORF, KCORF$_h$ is less prone to misclassification of the classes further away from the diagonal, which is ascribed to its modeling of heteroscedasticity in the data.

11.4 Conclusions

In this Chapter, we proposed the heteroscedastic kernel CORF model for intensity estimation of facial expressions of pain. The proposed model relaxes the homoscedasticity assumption in the CORF model. On the other hand, the standard classification methods such as SVM or CRF do not provide a principled way of accounting for heteroscedastic effects. Therefore, they cannot fully normalize the subject variability in data. Our experimental results show that, when LBPs are used as image descriptors, the heteroscedastic KCORF model attains better estimation of pain intensity than the homoscedastic KCORF model, and the other models for sequence classification (CORF and CRF). It also largely outperforms SVM, the state-of-the-art classifier for estimation the intensity of pain.

Chapter 12

Context-sensitive CORF for Intensity Estimation of AUs and Facial Expressions of Pain

Contents

12.1 Introduction 163
12.2 Methodology 165
12.3 Experiments 172
12.4 Conclusions 183

12.1 Introduction

Only a few approaches for AU intensity estimation have been proposed so far (see Sec. 2.3.4). These are based on either static classifiers such as SVM, or regression models such as RVM or SVR. However, from the modeling perspective, these approaches have the following limitations.

- Modeling the intensity levels on a nominal scale, as in the classification methods based on SVMs, is feasible but inefficient since these models ignore ordering of the intensity levels.
- Modeling the intensity levels on a continuous scale, as in the regression methods, is not optimal because of their implicit assumption of an interval scale. As can be seen from Fig.12.1, C and D intensity levels cover a larger range of appearance changes than the other levels. Moreover, discrete rating of intensity levels is often preferred and can

be accomplished more easily by human coders than the labeling of continuous-valued intensities.

- The learning/inference is static, i.e, per-frame/window. However, as argued in [60], modeling temporal dependencies in data is important for distinguishing between different intensity levels of afacial expression.

- The context in which intensity levels of AUs occur is not exploited. Consequently, these models do not account for factors such as (the expressivity of) the observed subject (see Fig.12.2), or the subject's current task. However, importance of these and the other context factors for modeling of facial expressions has been emphasized in [149].

- The frequency of occurrence of various intensity levels of AUs in spontaneous facial expressions is usually highly skewed to lower intensities (see Fig.12.5). Because the traditional models are designed for balanced data, this poses a serious challenge when learning the minority classes (i.e., the higher intensity levels).

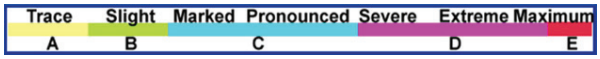
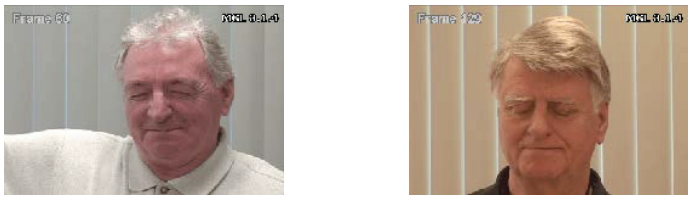


Figure 12.1: Relationship between the scale of facial appearance change and intensity levels when evidence of an AU is present [60].



(a) AU6C, PSPI=6 (b) AU6C, PSPI=6

Figure 12.2: Example images of two subjects from the UNBC Shoulder Pain dataset [125], whose facial action unit AU6 (cheek raiser and lid compressor) was coded with intensity C on the A-B-C-D-E ordinal scale. The intensity of the pain expression was computed from the codes of the co-occurring AUs in the images shown, using Prkachin and Solomon Pain Intensity (PSPI) rating. Observe the difference in the facial appearance of these two subjects whose AU6 and expressions of pain have the same intensity.

In this Chapter, we propose the Context-sensitive Conditional Ordinal Random Field (cs-CORF) model for dynamic estimation of the AU intensity levels that addresses the limitations mentioned above. This model is based on the standard CORF model and its heteroscedatic counterpart introduced in Chapter 11. Specifically, in the cs-CORF model we also account for the impact of context and biased intensity levels on AU intensity estimation. The omnipresent

influence of context is addressed by modeling context-sensitive variability in data. To this end, we adopt the *W5+* context model [149], where the six questions: *who* (the subject's identity, age and expressiveness level of the observed subject), *where* (environmental characteristics such as illumination), *what* (task-related cues of the facial action such as head tilts, nods, etc.), *how* (the information is passed on by means of facial expression intensity), *when* (timing of facial expressions and their intensity) and *why* (the context stimulus such as humorous videos), are used to summarize the key aspects of the context in which target expressions occur. Previously proposed approaches to AU intensity estimation (e.g., [130, 160, 168]) focus on the context question *how*, without taking into account the other context questions. By contrast, in cs-CORF we model the context questions *who*, *how* and *when*[1]. The context questions *who* and *how* are modeled by introducing separate Context-related Covariate Effects, named CCE, and Context-free Covariate Effects, named FCE (which coincide with the covariates used in the context-free models), respectively. These effects are efficiently embedded in the *ordinal* node features of a CRF model. Likewise, the context question *when* is modeled by the edge features of the model. The CCE component is derived from the subject's characteristics such as facial shape (when there is no AU activation present) that are considered constant across the sequence. This component is of particular importance because it directly accounts for the subject-specific bias in the parameters of the model. We also account for heteroscedasticity in data by allowing the model's variance to change depending on both the CCE and FCE components. This, in turn, allows the model to capture the expressiveness level of each subject. All these effects are summarized in the graphical representation of the proposed cs-CORF model shown in Fig. 12.3. Lastly, to address the problem of label/level imbalance in a principled manner, we introduce a weighted softmax-margin learning approach for CRFs, based on a generalization of the slack and margin rescaling modeling criteria in [196, 90].

12.2 Methodology

In this section, we first introduce the concept of context sensitive modeling of ordinal variables (i.e., the intensity levels of AUs). We then demonstrate how this concept can be used to model the context questions *who* and *how*. We continue by introducing the heteroscedastic effects in the model by allowing its variance to be a function of the context-sensitive covariates. The resulting model is then integrated into the framework of CRFs to model the context question *when* by accounting for temporal dependence between the ordinal variables. We then explain

[1] In this thesis, we limit our consideration to these three context questions because of their importance for describing context, as asserted in [149]. However, the other three context questions can be modeled in a similar manner.

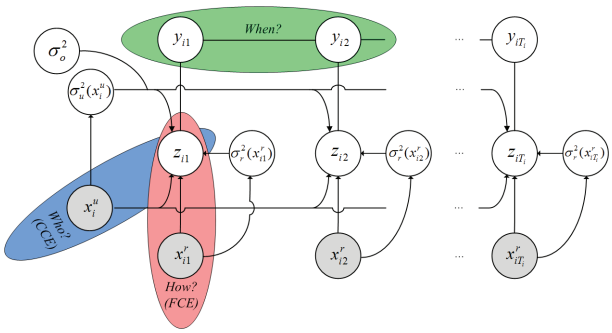


Figure 12.3: The cs-CORF model. The model's inputs are the time-varying FCE covariates (x^r_{ij}) and the constant (on the sequence level) CCE covariates (x^u_i), used to model the context-questions *how* and *who*, respectively. These effects are linearly related to the latent variable z_i, contaminated by Gaussian noise with zero mean and variance defined as the sum of the CCE ($\sigma^2_u(x^u_i)$) and FCE ($\sigma^2_r(x^r_{ii})$) heteroscedastic variance, as well as σ^2_o that accounts for unexplained variation in the data. The latent variable z_i is non-linearly connected to the ordinal labels y_i via the probit link function, used to define the node features in the cs-CORF model, which imposes ordinal constraints on the labels. The context question *when* is modeled in terms of interactions between the labels that are encoded by the edge features in the cs-CORF model.

learning and inference in the proposed model: we first introduce a weighted softmax-margin learning approach for data with skewed distribution of the intensity levels. Subsequently, we describe the regularizers used and the inference procedure.

12.2.1 Context-sensitive modeling

The context-sensitive modeling of data is attained by allowing the effects corresponding to different context questions to influence the output responses via the latent variable z. To this end, we extend the model in (8.1) as

$$\begin{aligned} z = \beta_1^T x^{who} + \beta_2^T x^{where} + \beta_3^T x^{what} \\ +\beta_4^T x^{how} + \beta_5^T x^{when} \ + \beta_6^T x^{why} + \epsilon, \end{aligned} \tag{12.1}$$

where the noise term has Gaussian distribution $\mathcal{N}(\epsilon; 0, \sigma^2)$. The covariates $(x^{who}, x^{where}, x^{how}, x^{when}, x^{why})$ are used to 'answer' the six context questions in the W5+ context design [149]. These covariates can be defined as a vector of features (e.g., x^{where} can represent gray-scale variation in an image due to illumination conditions), or a binary feature (e.g., x^{why} can indicate whether the observed subject is watching upsetting or humorous videos). Note that although z is linear in the effects modeling the context questions, this is not the case with the response variable y, which is *non-linearly* connected to z via (8.3). Therefore, the estimated intensity is the result of non-linear interactions of the different effects accounting for the context.

12.2.2 Modeling the context questions *who* and *how*

The latent variable model in (12.1) is general enough to address all six context questions. To demonstrate how the proposed model can be applied to the target task (i.e., AU intensity estimation), in what follows, we focus on the context questions *who* and *how*, though, the other context questions can be modeled in a similar way. These two questions are of particular importance, since the first directly accounts for the subject-specific expressivity, while the second accounts for relationships between the observed facial changes and AU intensity that are common for all subjects. To model these two context questions, we introduce context-related covariate effects (CCE) and context-free covariate effects (FCE), which represent the covariates x^{who} and x^{how} in (12.1), respectively. The latter are called the context-free in our work because these covariates coincide with those used in the traditional context-free models (e.g., [102]), where the normalization w.r.t. the first frame in a sequence is performed to 'remove' the differences between subjects. We derive the CCE and FCE components as follows. Given a sequence of ordinal intensities, $\mathbf{y}_i = \{y_{i1}, \ldots, y_{iT_i}\}$, with the corresponding covariate values $\mathbf{x}_i = \{x_{i1}, \ldots, x_{iT_i}\}$, we decompose x_{ij} into CCE ($x_i^u = C^{-1}\sum_{c=1}^{C} x_{ic}$) and FCE ($x_{ij}^r = x_{ij} - x_i^u$) component. The CCE component accounts for effects that are considered constant across the sequence but may vary between sequences (e.g., the facial shapes of different subjects). Here, we estimate it from the first C neutral frames of a sequence[2]. On the other hand, the FCE component accounts for variability *within* the sequence (i.e., the expression intensity). With these newly introduced effects, we write the latent variable model in (12.1) as

$$z_{ij} = \beta_u^T x_i^u + \beta_r^T x_{ij}^r + \epsilon_{ij}. \tag{12.2}$$

By following the same approach as in (8.3), we obtain the context-sensitive cumulative probits as

$$\lambda_{ijk} = \gamma_k - \beta_u^T x_i^u - \beta_r^T x_{ij}^r,\ k = 1, \ldots K, \tag{12.3}$$

where $\sigma = 1$. From (12.3), we can distinguish between (i) an overall effect of the CCE component, as measured by association with the responses, and (ii) the time-varying effects of the FCE component, having a different affect on each response within the sequence. Consequently, the locations of the thresholds γ_k, dividing the ordinal line into the bins corresponding to different intensity levels, are adjusted to the target subject by means of the CCE component ($\beta_u^T x_i^u$). On the other hand, the FCE component ($\beta_u^T x_i^u$) ensures that the intensity-related variation is placed correctly into such adjusted bins. This simultaneous interaction of the CCE and FCE components with the other parameters of the model is at the heart of our approach. If

[2]We set C=5 to obtain a more robust estimate of the target covariates. However, a single frame should suffice.

the CCE component is removed from the model ($\beta_u = 0, \beta_r \neq 0$), the context is lost and it becomes difficult for the model to adapt to different subjects. Conversely, assuming the common effects ($\beta_u = \beta_r$) can lead to very misleading association of covariates with the responses, since they model neither CCE nor FCE covariate effects.

12.2.3 Heteroscedastic noise model

In the previous sections, the latent variable z is defined using the homoscedastic noise model, i.e., the variance σ^2 of the noise term is constant. However, since the CCE component has an additive effect on the locations of the model's thresholds γ_k within a sequence, it accounts only for the mean level of the subject's expressiveness. For the model to be able to fully adapt to different subjects, we also need to allow the scale of the ordinal thresholds to change. This can be attained by allowing the noise level to vary as a function of the covariates, as we did in Chapter 11. For this, we further extend the latent variable model in (12.2) by introducing separate noise terms

$$z_{ij} = \beta_u^T x_i^u + \beta_r^T x_{ij}^r + \delta_i^u + \delta_{ij}^r + \delta_{ij}, \tag{12.4}$$

where $\mathcal{N}(\delta_i^u; 0, \sigma_u(x_i^u))$ and $\mathcal{N}(\delta_{ij}^r; 0, \sigma_r(x_{ij}^r))$. We also keep the constant noise term to account for sources of variation that are not included in the model (e.g., the effects of the other context questions). Here we assume that the three noise terms are independent, so the distribution of the overall noise in the model is a zero-mean Gaussian with the variance

$$\sigma^2(x_{ij}) = \sigma_u^2(x_i^u) + \sigma_r^2(x_{ij}^r) + \sigma_o^2, \tag{12.5}$$

The first two terms on the right represent the CCE and FCE variance, respectively, and are defined as the log-linear function of their covariates, i.e., $\log \sigma_u = \upsilon_u^T x_i^u$ and $\log \sigma_r = \upsilon_r^T x_{ij}^r$. The parameters υ_u and υ_r indicate the level of influence of the CCE and FCE variances, respectively, and the *log* function ensures the positivity of the standard deviation. Using the latent variable model in (12.5), and after the marginalization in (12.3), we obtain the context-sensitive cumulative probits, which also have the changing variance, as

$$\lambda_{ijk} = \gamma_k \sigma^{-1}(x_{ij}) - (\beta_u^T x_i^u + \beta_r^T x_{ij}^r)\sigma^{-1}(x_{ij}), \tag{12.6}$$

where the context-sensitive ordinal likelihood is $P(y_{ij} = k|z_{ij}) = \Phi(\lambda_{ij,k}) - \Phi(\lambda_{ij,k-1})$. From (12.6), we see that both the constant CCE and time-varying FCE covariates influence the scale of the model's thresholds as well as its location, thus, allowing it to adapt to the context above and beyond the contribution of the CCE effects. Note, however, that since we use the same covariates in the location and variance representation, the identification may be fragile. The model can still be identified due to the different functional forms specified for the covariates, but it is necessary to regularize the parameters [183]. This is explained below.

12.2.4 Modeling the context question *when*

The context-sensitive ordinal likelihood introduced in Sec.12.2.3 aims at static classification of ordinal variables. Although the latent variable model in (12.1) can be used to model the context question *when* by encoding temporal correlations between either the input features (x^{when}) or different instances of latent variable z, or both. Another way to model the context question *when* is to encode the temporal correlation directly in the output space of ordinal variables y, as done in the standard CORF model. Recall that CORF employs the linear-chain CRF [108] model that represents the conditional distribution $p(\mathbf{y}_i|\mathbf{x}_i;\theta)$ as

$$p(\mathbf{y}_i|\mathbf{x}_i;\theta) = \frac{\exp(\sum_{j=2}^{T_i}\Psi(y_{i,j-1}, y_{ij}, \mathbf{x}_i;\theta))}{\sum\limits_{\bar{\mathbf{y}}\in\mathcal{Y}^{|T_i|}}\exp(\sum_{j=2}^{T_i}\Psi(\bar{y}_{i,j-1}, \bar{y}_{ij}, \mathbf{x}_i;\theta))}, \tag{12.7}$$

where T_i is the duration of the i-th sequence, and $\mathcal{Y}^{|T_i|}$ is the set of all possible output configurations of the output graph $G = (V, E)$. Furthermore, θ are the parameters of the score function $\Psi(y_{i,j-1}, y_{ij}, x_i;\theta) \equiv \Psi_{ij}(y)$[3] defined on *node* cliques ($r \in V$) and *edge* cliques ($e = (s, r) \in E$) of the graph as

$$\Psi_{ij}(y) = f_n(y_{ij}, \mathbf{x}_i) + f_e(y_{i,j-1}, y_{ij}), \tag{12.8}$$

where $f_n(y_{ij}, x_i)$ and $f_e(y_{i,j-1}, y_{ij})$ are the *node* and *edge* features, respectively. We use the introduced context-sensitive ordinal likelihood $p(y_{ij} = k|z^*_{ij}) = \Phi(\lambda_{ij,k}) - \Phi(\lambda_{ij,k-1})$ to define the *node* features as

$$f_n(y_{ij}, \mathbf{x}_i) = \sum_{k=1}^{K} I(y_{ij} = k)\cdot\log p(y_{ij} = k|z^*_{ij}), \tag{12.9}$$

where $I(\cdot)$ is the indicator function that returns 1 (0) if the argument is true (false). The *edge* features are defined as the first order Markov dependence between the ordinal responses as

$$f_e(y_{i,j-1}, y_{ij}) = \sum_{m,k=1}^{K} I(y_{i,j-1} = m \wedge\ y_{ij} = k)\cdot u_{mk}, \tag{12.10}$$

where $m, k{=}1\ldots K$, and u_{mk} measures the temporal association between the responses. Note that the denominator of (12.7) guarantees that the distribution sums to one, and is computed using (12.9) and (12.10) without the indicator function. Now, given i.i.d. training data $\{\mathbf{y}_i, \mathbf{x}_i\}_{i=1}^N$, the parameters $\theta = \{\{\gamma_k\}_{k=1}^{K-1}, \sigma_o, \beta_u, \beta_r, v_u, v_r, \{u_{mk}\}_{m,k=1}^K\}$ are found by minimizing the regularized conditional log-likelihood.

$$\min_\theta R(\theta) - \sum\nolimits_{i=1}^{N}\log p(\mathbf{y}_i|\mathbf{x}_i;\theta), \tag{12.11}$$

[3] We drop dependence on $j-1$, x_i and θ for notational simplicity.

where $R(\theta)$ is the regularization term that prevents the model from overfitting. We name this model the Context-sensitive Conditional Ordinal Random Field (cs-CORF) model.

12.2.5 Learning and Inference

Weighted Softmax-margin Learning. To deal with skewed distribution of ordinal responses, we relate the large-margin learning approach for sequence classification in [172] to the CRF model in (12.7). However, in contrast to [172], we introduce scaling of the slack variables, which imposes a higher penalty when making errors on minority classes during learning. We start from the standard primal learning approach for max-margin models [196, 90]:

$$\begin{aligned} &\min_{\zeta_{ij},\theta} R(\theta) + \textstyle\sum_{i=1}^{N} \sum_{j=1}^{T_i} \zeta_{ij} \\ &s.t.\ \Psi_{ij}(y) - \Psi_{ij}(\bar{y}) \geq \Delta_{ij}(y,\bar{y}) - \tfrac{\zeta_{ij}}{w_{ij}(y,\bar{y})}, \\ &\forall \bar{y} \in \mathcal{Y}, \zeta_{ij} > 0, i = 1 \ldots N, j = 1 \ldots T_i, \end{aligned} \tag{12.12}$$

where the large-margin set of constraints are applied to the score function defined in (12.8). These constraints enforce the difference between the scores of the correctly labeled cliques ($\Psi_{ij}(y)$) and incorrectly labeled cliques ($\Psi_{ij}(\bar{y}), y \neq \bar{y}$) to be greater than the loss $\Delta_{ij}(y,\bar{y})$. This loss is defined on the temporally neighboring pairs of labels as the weighted Hamming loss, i.e., $\Delta_{ij}(y,\bar{y}) = 1 - [\alpha I(y_{ij},\bar{y}_{ij}) + (1-\alpha) I(y_{ij-1},\bar{y}_{ij-1})]$, for $j{>}1$ and $0 \leq \alpha \leq 1$, while for the first example in the sequence ($j{=}1$), we set $\alpha{=}1$. The weighting of the slack variables ζ_{ij} is attained using the information about a prior distribution of the intensity levels as $p(y) = N_y / \sum_{k=1}^{K} N_k$, leading to $w_{ij}(y,\bar{y}) = w_{ij}(y) = 1/(p(y)+\varepsilon)$. The parameter ε is chosen from the range $[0,1]$ in order to ensure that the overall loss is not dominated by minority classes. The constraints in (12.12) can further be written as

$$w_{ij}(y)\Psi_{ij}(y) - w_{ij}(y)(\Psi_{ij}(\bar{y}) + \Delta_{ij}(y,\bar{y})) \geq -\zeta_{ij}, \tag{12.13}$$

Note that when the weight $w_{ij}(y)$ is set to one, the constraint in (12.13) is equivalent to that used in the conventional n-Slack large-margin learning with margin-rescaling (e.g., [196]). We now re-write the optimization problem in (12.12) in the form that folds the multiple constraints into a single constraint per training sequence, thus

$$\begin{aligned} &\min_{\zeta_i,\theta} R(\theta) + \textstyle\sum_{i=1}^{N} \zeta_i \\ &s.t.\ \textstyle\sum_{j=1}^{T_i} \Big[\Psi_{ij}^{w}(y) - (\Psi_{ij}^{w}(\bar{y}) + \Delta_{ij}^{w}(y,\bar{y}))\Big] \geq -\zeta_i, \\ &\forall \bar{\mathbf{y}}_{ij} \in \mathcal{Y}^{|T_i|},\ i = 1 \ldots N,\ \zeta_i > 0, \end{aligned} \tag{12.14}$$

where we simplify notation by defining $\Psi_{ij}^{w}(y) \equiv w_{ij}(y)\Psi_{ij}(y)$, $\Psi_{ij}^{w}(\bar{y}) \equiv w_{ij}(y)\Psi_{ij}(\bar{y})$ and $\Delta_{ij}^{w}(y,\bar{y}) \equiv w_{ij}(y)\Delta_{ij}(y,\bar{y})$. While the optimization problem (OP) in (12.14) has $N \cdot \mathcal{Y}^{|T_i|}$,

$i = 1...N$, constraints, one for each possible combination of labels $\bar{\mathbf{y}}_i = (\bar{y}_{i1}, ..., \bar{y}_{iT_i}) \in \mathcal{Y}^{|T_i|}$, it has only one slack variable ζ_i per sequence. This is exactly what we need for sequence learning since, in contrast to ζ_{ij} in OP in (12.12), each ζ_i in OP in (12.14) can now be optimized individually for given θ. The smallest feasible ζ_i given θ is then achieved for:

$$\zeta_i = \max_{\bar{\mathbf{y}}_\mathbf{i} \in \mathcal{Y}^{|T_i|}} \sum\nolimits_{j=1}^{T_i} (\Psi_{ij}^w(\bar{y}) + \Delta_{ij}^w(y, \bar{y})) - \sum\nolimits_{j=1}^{T_i} \Psi_{ij}^w(y) \tag{12.15}$$

We next obtain a more workable constraint by replacing the *max* term with the *softmax* upper bound using the inequality $\max_i g_i \leq \log \sum_i e^{g_i}$, which leads to

$$\zeta_i = \log \sum_{\bar{\mathbf{y}}_\mathbf{i} \in \mathcal{Y}^{|T_i|}} \mathrm{e}^{\sum_{j=1}^{T_i} \Psi_{ij}^w(\bar{y}) + \Delta_{ij}^w(y, \bar{y})} - \sum\nolimits_{j=1}^{T_i} \Psi_{ij}^w(y) \tag{12.16}$$

The constraint in (12.16) is more restricted than that in (12.15) since it uses an upper bound on the gap between the scores of the true and model labeling of the sequence. More importantly, in contrast to the *max* constraint, the *softmax* large-margin constraint is a differentiable function of the model parameters. We use this to cast the OP in (12.14) as an unconstrained OP. Specifically, since the constraint in (12.16) has a form similar to that of the negative log of the conditional probability of CRFs defined in (12.7), we can formulate the weighted softmax-margin learning of the CRF/cs-CORF model as the following (unconstrained) OP:

$$\min_{\zeta_i, \theta} R(\theta) + \sum\nolimits_{i=1}^{N} \zeta_i \equiv \min_{\theta} R(\theta) - \sum\nolimits_{i=1}^{N} \log p^w(\mathbf{y}_i | \mathbf{x}_i; \theta), \tag{12.17}$$

where the conditional likelihood-like term p^w is defined as

$$p^w(\mathbf{y}_i | \mathbf{x}_i; \theta) = \frac{\exp(\sum_{j=1}^{T_i} \Psi_{ij}^w(y))}{\sum_{\bar{\mathbf{y}} \in \mathcal{Y}^{|T_i|}} \exp(\sum_{j=1}^{T_i} \Psi_{ij}^w(\bar{y}) + \Delta_{ij}^w(y, \bar{y}))} \tag{12.18}$$

The introduced formulation of the weighted softmax-margin learning allows us to compute the model parameters θ efficiently by the gradient optimization and dynamic programming techniques (e.g., Viterbi algorithm), commonly used for CRFs. Thus, the implementation is straightforward as it only requires applying the weights to the score function $\Psi(\cdot)$ penalized with the loss $\Delta(\cdot)$. On the other hand, the inference is performed by using the unweighted/unpenalized likelihood in (12.7).

Note that OP in (12.17) has a form similar to OPs in other softmax-margin approaches (e.g., [172, 100, 68, 6]). However, none of those approaches addresses the problem of class imbalance. Note also that 'slack-rescaling' in [196, 90] is defined as another way of large-margin structured learning, in addition to 'margin-rescaling', where the slack variables are

scaled using the inverse *loss* $\Delta(y, \bar{y})$. This is different from our approach where the slack variables are scaled with the inverse *weights* $w(y)$ in order to balance the contribution of the loss of the minority and majority classes. Moreover, we include the *loss* $\Delta(y, \bar{y})$ using the 'margin-rescaling' approach because, in contrast to 'slack-rescaling', it allows us to formulate the OP as that of standard CRFs (with the likelihood-like term in (12.18)).

Regularizers. To deal with the order constraints in the parameters γ, we introduce the displacement variables δ_k, where $\gamma_j = \gamma_1 + \sum_{k=1}^{j-1} \delta_k^2$ for $j = 2, \ldots, K-1$. So, γ is replaced by the unconstrained parameters $\{\gamma_1, \delta_1, \ldots, \delta_{K-2}\}$. Another important issue is the regularization of the parameters of the cs-CORF model. We use the L_2 regularizer for the standard CRF parameters, resulting in the regularization term $R(\theta)$ defined as

$$R(\theta) = \rho_1(\|\beta_u\|^2 + \|v_u\|^2) + \rho_2(\|\beta_r\|^2 + \|v_r\|^2) + \rho_3\|u\|^2, \tag{12.19}$$

where (ρ_1, ρ_2, ρ_3) are the regularization parameters, which help to balance the impact of the CCE and FCE effects and the dynamics in the model, in order to avoid the overfitting. With $R(\theta)$, as defined in (12.19), the optimal parameters of the model are found by minimizing the objective in (12.17) using the quasi-Newton LBFGS method. The inference of test sequences is performed using Viterbi decoding, applied to the 'unweighted' conditional likelihood in (12.7).

12.3 Experiments

12.3.1 Datasets and Experimental Procedure

Datasets. Evaluation of the proposed model is performed on the UNBC-MacMaster Shoulder Pain Expression Archive (Shoulder-Pain) [125] and Denver Intensity of Spontaneous Facial Actions (DISFA) [133] datasets. To the best of our knowledge, these are the only two sets of naturalistic data that contain a large number of FACS coded AUs and their intensity. We denote these intensity levels using ordinal scores: 0 (not present) to 5 (maximal intensity).

The Shoulder-pain dataset is described in Chapter 11. In addition to the PSPI scores (i.e., intensity levels for facial expressions of pain), the dataset creators also provide the coding of 11 AUs (4, 6, 7, 9, 10, 12, 20, 25, 26, 27 and 43) and their intensity. As there are only a few examples of higher intensities of AU27, we do not include this AU in our experiments. For similar reasons, we merge examples of levels 5 and 6 of AU12 and AU20. For the ground-truth for intensity of pain expression, we again grouped the PSPI scores into six levels.

The DISFA dataset contains video recordings of 27 subjects watching 'YouTube' video clips. Each image frame was coded in terms of 12 AUs (1, 2, 4, 5, 6, 9, 12, 15, 17, 20, 25

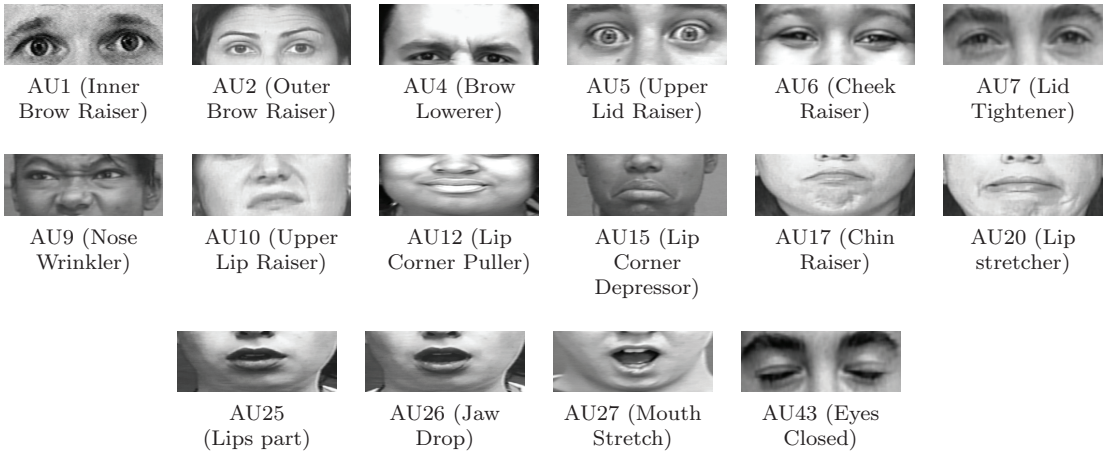


Figure 12.4: Examples of AUs available in Shoulder-Pain and/or DISFA dataset. The images are obtained from http://www.cs.cmu.edu/~face/facs.htm.

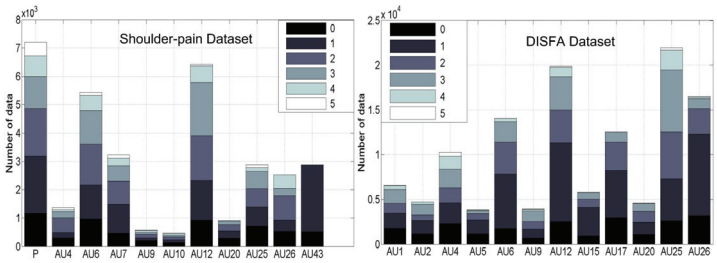


Figure 12.5: Distribution of the intensity levels of AUs used from the Shoulder-pain (*left*) and DISFA (*right*) datasets.

and 26) and their intensity. Since for AU15 and AU20, there are no examples of the intensity level 6 and only a few examples of level 5, we merged levels 4 and 5, resulting in 4 intensity levels for these AUs. For the same reason, we merged examples of the intensity levels 5 and 6 for AU17. Examples of AUs that are present in either the Shoulder-pain or DISFA dataset, or both, are shown in Fig.12.4. Since the recordings contain predominantly expressionless faces (i.e., 0 intensity level for all AUs), the sequences from both datasets were pre-segmented per AU. Specifically, the segments containing non-neutral AU intensity were marked first. Then, the surrounding neutral-intensity frames were added at the beginning and end of these segments. The number of 'neutral' frames was balanced with the second most frequent intensity level of the target AU. Fig.12.5 shows the distribution of the intensity levels after segmentation of the sequences. The sequences made in this way were used to evaluate the models.

Features. As the input to our model, we used the facial representation based on geometric features (i.e., the locations of 66 facial landmarks depicted in Fig.12.9, which were

obtained using a 2D Active Appearance Model (2D-AAM) [125]) as they have already been used successfully for AU recognition tasks (e.g., [126, 200]). Note that in [126, 92] the authors claim that improved recognition performance can be attained by using both the geometric and appearance features (e.g., gray-scale intensity). However, registration of the facial appearance in spontaneous data is still an open problem because of large head movements. For this reason, we limit our analysis to geometric features only. To register the features, we applied an affine transform that maps the facial landmarks from faces in each dataset to those of the corresponding reference face (we used the average face from the target dataset). To reduce the size of features, we applied Principal Component Analysis (PCA) to 132-D feature vectors obtained by concatenation of the (x, y) coordinates of the 66 facial landmarks. On average, this resulted in 18-D features, preserving 97% of variation in the data. These were then used to derive the CCE and FCE covariates as explained in Sec.12.2.2.

Models. We compare the performance of the cs-CORF and standard CORF models, and their variants. Specifically, we compare the maximum-likelihood and the proposed weighted softmax-margin learning of the models, denoted by 'ml' and 'w', respectively. Next, we compare the CORFs with the homoscedastic (σ=1) and heteroscedastic ($\sigma(x)$) noise models, with the latter denoted by 'h'. To compare the ordinal over nominal modeling of the target tasks, we show performance of the standard linear-chain CRF model [108], trained using both 'ml' and 'w' learning. As the baseline model, we use one-vs-all SVM [33]. We also perform comparisons with the state-of-the-art *static* ordinal regression models, Support Vector Ordinal Regression (SVOR) with implicit constraints [38], and Gaussian Process Ordinal Regression (GPOR) with Laplace approximation [37]. In the kernel methods (SVM/SVOR/GPOR), we used a linear kernel function, to have a fair comparison with the linear CRF/CORF-based models. Finally, we include the comparisons with the state-of-the-art models for AU intensity estimation: the RVM approach [92], where continuous estimation of AU intensity is performed, and Spectral Regression [31] combined with one-vs-one SVM (SR+SVM) [130, 133]. The continuous predictions by the RVM-based approach were rounded to the nearest intensity level. For the SR+SVM approach, AU-specific subspaces were selected by running a validation procedure on the training set. In both methods, we used the RBF kernel, as done in the original works [92, 133]. The width of the RBF kernel was set as the median of the (feature) distance set, i.e., $\{\|x_i - x_j\|, i, j = 1, ..., N, i < j\}$ [94]. The hyper/regularization-parameters for different methods were selected by validation on the training set using a grid-search in the range $\rho = \{10^{-4}, 10^{-3}, ..., 1, 2, 5\}$. If not stated otherwise, in all experiments training/testing was performed by running a 5-fold cross

validation procedure, with each fold containing intensity sequences of different subjects.

Evaluation Scores. The performance of the models is reported using: (i) average F-1 computed from F-1 scores for each pain intensity, (ii) mean absolute error (MAE), computed between actual and predicted pain intensities, (iii) Intra-Class Correlation (ICC) [181], and (iv) Ordinal Classification Index (OCI) [32]. Since the distribution of AU intensities is highly imbalanced (in contrast to that of temporal segments of AUs and facial expressions of basic emotions, considered in the previous Chapters), here we use weighted MAE that is defined as

$$\text{MAE} = \frac{1}{K}\sum\nolimits_{j=1}^{K}\frac{1}{N_k}\sum\nolimits_{y_i \in N_k}|y_i - y_i^*|,$$

where N_k is the set of examples of class k, and y_i and y_i^* are the actual and predicted class labels, respectively. The OCI score is obtained directly from the confusion matrix (CM), and is defined as

$$OCI = \min\left\{1 - \frac{\sum\limits_{(r,c)\in path} n_{r,c}}{100\cdot K + \sum\limits_{\forall(r,c)} n_{r,c}\,|r-c|} + \kappa \sum_{(r,c)\in path} n_{r,c}\,|r-c|\right\},$$

where $n_{r,c}$ is the fraction (in %) of examples from the r-th class predicted as being from the c-th class, and the *path* is defined as a sequence of entries where two consecutive entries in the path are 8-adjacent neighbors (see [32] for details). For small values of κ (we set it to 0.25), OCI focuses on measuring ordinal performance from CMs. This score is a *dissimilarity* measure ranging from 0 to 100 (in %).

We use these scores because they capture complementary information about performance of the models. Specifically, F1 focuses on absolute classification, while MAE, as noted in [32], "capture[s] how much the result diverges from the ideal prediction and how *inconsistent* the classifier is in regard to the relative order of the classes". On the other hand, ICC measures the *consistency* of the predictions, and, unlike MAE, is less restrictive in that it disregards the bias that may exist between the true and predicted labels. We also employ OCI as it simplifies comparisons of CMs by different models. Note that all scores defined above, except ICC, are robust to class imbalance, which makes them suitable for the imbalanced learning problems.

12.3.2 Experimental Results

In this section, we first show some qualitative results. We then show the comparisons with the state-of-the-art models using the context and context-free covariates. We continue by showing the results for the intensity estimation of individual AUs and facial expressions of

pain, followed by analysis of performance on two specific AUs (6 and 25). Lastly, we show the results of the cross-dataset experiments.

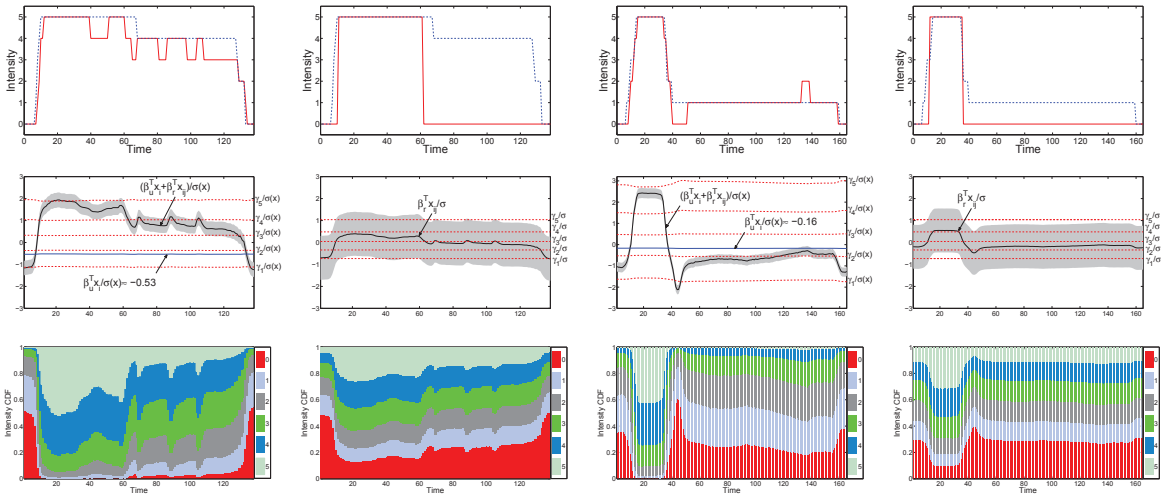


Figure 12.6: The intensity estimation of pain from two example sequences from the Shoulder-pain dataset, attained by cs-CORF(w+h) and base CORF(w). The upper row shows true (*dashed blue*) and predicted (*solid red*) labels by the two models. The middle row shows the ordinal projections of the inputs (*solid black*), with their standard deviation σ (*grey*), and the scaled thresholds (*dashed red*). For cs-CORF(w+h), we also plot the context-induced 'bias' (*solid blue*). The bottom row shows the probability of the pain intensity in each frame.

Qualitative results. To get an insight into the role of the different effects in the proposed model we focus first on comparison between the cs-CORF model (CCE and FCE effects) and the (homoscedastic) CORF model (FCE effects). Both models were optimized using the introduced weighted softmax-margin approach. The performance of the models is demonstrated on the *pain* intensity estimation using two example sequences. As can be seen from Fig.12.6 (top row), the cs-CORF model predicts the intensity levels relatively well, while the CORF model fails to predict level 4 correctly in the first sequence, and level 1 in the second. The middle row of Fig.12.6 shows the values of the corresponding ordinal projections, along with the model parameters. By looking at the ordinal thresholds of the two models, we see that the thresholds of the cs-CORF model achieve a better segmentation of the target signal into discrete intensity levels[4]. This is because (i) their scaling by the heteroscedastic variance and (ii) correction of the subject-specific bias by means of the CCE component. On the other

[4]In a noise-free case, the partitioning of the signal should correspond to the assigned intensity labels. However, as can be seen from Fig.12.6 (middle row), for real data the estimated width of the bins is of the order of the variance, so the segmentation of the latent variable is sometimes inconsistent with the actual predictions of the model.

	F1		MAE		ICC	
	CCE+FCE	FCE	CCE+FCE	FCE	CCE+FCE	FCE
SVM	24.1 (18.1)	24.8 (15.7)	1.13 (20.3)	1.02 (18.5)	34.9 (17.9)	36.9 (16.2)
GPOR	24.4 (17.6)	23.5 (20.4)	0.96 (16.6)	1.06 (19.5)	38.6 (14.9)	37.2 (15.5)
SVOR	26.1 (14.2)	25.2 (15.1)	0.88 (13.3)	0.91 (14.9)	41.8 (13.8)	38.5 (15.2)
RVM	24.5 (17.2)	27.3 (14.8)	0.94 (15.5)	0.93 (15.7)	24.7 (20.4)	31.5 (17.9)
SR+SVM	25.7 (15.8)	29.7 (11.0)	0.94 (15.4)	0.82 (9.1)	27.6 (18.7)	38.7 (15.0)
CRF(ml)	29.3 (11.9)	29.5 (12.1)	0.87 (12.1)	0.86 (11.9)	45.3 (11.7)	46.8 (11.2)
CRF(w)	32.0 (8.1)	31.5 (9.1)	0.84 (10.3)	0.83 (10.3)	49.7 (7.9)	50.2 (7.7)
CORF(ml)	33.2 (6.8)	31.0 (10.4)	0.77 (5.9)	0.79 (8.1)	52.6 (5.9)	48.4 (9.9)
CORF(w)	35.5 (3.9)	33.2 (7.2)	0.73 (3.6)	0.78 (7.4)	54.8 (3.9)	50.2 (8.6)
CORF(ml+h)	35.3 (3.9)	32.8 (7.7)	0.74 (3.9)	0.78 (7.5)	56.0 (2.9)	51.2 (7.6)
CORF(w+h)	**38.7 (1.4)**	34.8 (4.9)	**0.69 (1.9)**	0.76 (5.8)	**59.1 (1.1)**	53.3 (5.5)

Table 12.1: The average performance of the models tested on 23 intensity estimation problems (*pain* expressions + 10 AUs from Shoulder-pain dataset and 12 AUs from DISFA dataset). The numbers in brackets are the average ranks of the models, where the ranking is performed on 46 (=23×2) tasks, as each model is tested using two sets of covariates: the context (CCE+FCE) and context-free (FCE) covariates. The models are ranked separately for each task, the best performing model getting the rank of 1, the second best rank 2, etc. Note that for all three scores, the top ranked model is the proposed context-sensitive CORF(w+h) model (i.e., CORF(w+h) with CCE+FCE).

hand, the base CORF model is far less flexible due to its limited parametrization ($\sigma = 1$ and there is no modeling of the context), resulting in poor estimation of the intermediate intensity levels. Fig.12.6 (bottom row) shows that the maximum probability of the intensity levels is consistent with the actual predictions of the models. From these probabilities, we also conclude that cs-CORF is more discriminative than the standard CORF model.

Comparisons with the state-of-the-art models using the context and context-free covariates. Table 12.1 shows the average results of various models, obtained by a 5-fold cross-validation, for the following 23 intensity estimation problems: *pain* expressions and 10 AUs from the Shoulder-pain dataset, and 12 AUs from the DISFA dataset. The models were evaluated using two sets of covariates: context (CCE+FCE) and context-free (FCE). To ensure that the performance of the models is consistent across the 46 tasks (23 problems×2 sets of covariates), we performed the ranking of the models, as in [53] (see Sec.3.2.2). Specifically, the models were first ranked per task, the best performing model getting the rank of 1, the second best rank 2, etc. In the case of ties, average ranks were assigned. The final ranking was then obtained by averaging the ranks over all tasks. From Table 12.1, we make the following observations. The base SVM model is outperformed by the SR+SVM model when the context-free covariates are used. This is attributed in part to the fact that the latter performs non-linear feature selection by means of SR, and in part to the fact that it uses a non-linear kernel function in the SVM classifier, as well as one-vs-one learning strategy. This differs from the base SVM model, where we used a liner kernel and one-vs-all strategy. On the other hand, both models underperform when the context covariates are used, possibly due to the overfitting of the CCE covariates. This difference in performance is more pronounced in SR+SVM, which is prone to

overfitting of the subspace. The RVM method, although designed for continuous estimation, shows the performance (in terms of F1 and MAE) comparable to that of SVM. However, its ICC scores are lower, which indicates that its estimation of the intensity levels is not always consistent. The static ordinal models, GPOR and SVOR, showed a small improvement in their performance when the context covariates are used. Furthermore, SVOR performed better than the base SVM model across all three scores. The improvement in ICC scores of GPOR and SVOR over nominal static models and RVM, in contrast to the the other two scores, implies that there is a bias in the estimated intensity levels by these ordinal models. Also, the lower performance in terms of F1 and MAE of GPOR is ascribed to its learning being less robust to imbalanced data than that of the max-margin models (i.e., SVOR and SVM). Next, the standard CRF(ml) model performed marginally better than the base SVM in terms of F1. However, its MAE and ICC are much better mainly because of the temporal smoothing of the predicted intensity. On the other hand, the proposed weighted softmax-margin learning improved the performance of the CRF with 'ml' learning. Yet, there is not much difference when using the context or context-free covariates. However, inclusion of the context covariates in the CORF(ml) model results in an improvement in all three scores, compared to the context-free case. CORF(ml) also outperformed the static ordinal models, GPOR and SVOR, which, evidently, remained affected by temporal variability of the data during learning/inference. Then again, the weighted softmax-margin learning (CORF(w)) and the heteroscedastic noise model (CORF(ml+h)) further enhanced the performance of CORF(ml). Moreover, based on the score values and the ranking of the models, the combination of the weighted learning and the heteroscedastic noise model in cs-CORF(w+h) (i.e., CORF(w+h) with FCE+CCE) is the most effective for the target tasks.

Performance on individual AUs and facial expressions of pain. Table 12.2 shows results of the cs-CORF(w+h), CORF(w+h) and CRF(w) models. We also include the results obtained by two state-of-the-art (context-free) models for AU intensity estimation: SR+SVM [133] and RVM [92]. The numbers with bold face in the table indicate that the differences in scores by the proposed cs-CORF(w+h) and the rest of the models are significant, based on the paired t-test ($p = 0.05$). The proposed cs-CORf(w+h) model performs similarly or better than the context-free models in most tasks. Specifically, from Table 12.2 (a), in the case of AU12, cs-CORF(w+h) consistently outperforms the other models. We ascribe this to the fact that AU12 involves activation of an oblique muscle, characterized by curved motion that is usually subject-specific. Therefore, modeling the context question *who*, obviously results in a better performance than that attained by the CORF model. By contrast, AU10 involves activation of vertically set muscles above the upper lip. Similarly, AU9 involves a vertical pull of the

		P	AU4	AU6	AU7	AU9	AU10	AU12	AU20	AU25	AU26	AU43
F1	RVM	22.8	26.7	22.2	22.1	23.5	43.0	27.8	25.5	22.1	22.0	70.7
	SR+SVM	29.4	24.3	23.9	22.3	32.6	43.4	26.7	29.6	36.0	32.4	78.3
	CRF(w)	30.0	27.0	29.0	29.0	33.0	42.0	32.0	32.0	29.0	26.0	76.0
	CORF(w+h)	35.0	32.0	36.0	30.0	41.0	49.0	35.0	34.0	33.0	27.0	78.0
	cs-CORF(w+h)	**41.0**	35.0	**41.0**	**38.0**	**45.0**	50.0	**39.0**	36.0	34.0	30.0	**89.0**
MAE	RVM	1.00	1.05	1.16	1.25	1.30	0.64	0.98	0.99	1.16	1.50	0.18
	SR+SVM	1.00	0.93	0.97	1.13	0.85	0.63	0.81	0.85	0.97	1.39	0.11
	CRF(w)	1.16	0.99	0.98	1.00	0.82	0.53	0.94	0.93	0.99	1.23	0.13
	CORF(w+h)	0.93	0.88	0.79	0.90	0.75	0.41	0.81	0.83	0.95	1.23	0.11
	cs-CORF(w+h)	**0.82**	**0.79**	**0.71**	**0.76**	0.70	0.36	**0.68**	**0.74**	**0.81**	1.19	**0.05**
ICC	RVM	43.1	33.9	18.8	28.9	-0.5	39.1	27.7	16.3	21.7	16.8	46.0
	SR+SVM	44.4	54.6	36.0	27.2	43.4	37.8	34.0	35.2	38.8	18.2	59.1
	CRF(w)	58.0	66.0	52.0	54.0	52.0	49.0	51.0	37.0	43.0	29.0	54.0
	CORF(w+h)	59.0	72.0	60.0	59.0	61.0	65.0	57.0	39.0	50.0	25.0	61.0
	cs-CORF(w+h)	**64.0**	75.0	**67.0**	**68.0**	63.0	66.0	**62.0**	**47.0**	**58.0**	**38.0**	**73.0**

(a) The Shoulder-Pain dataset

		AU1	AU2	AU4	AU5	AU6	AU9	AU12	AU15	AU17	AU20	AU25	AU26
F1	RVM	29.6	29.6	30.7	26.1	27.3	23.3	32.6	24.8	29.7	28.6	34.9	25.9
	SR+SVM	30.7	27.0	28.0	27.1	25.3	30.6	26.5	29.0	29.3	34.3	40.4	24.9
	CRF(w)	30.0	34.0	30.0	33.0	28.0	29.0	34.0	33.0	37.0	35.0	36.0	25.0
	CORF(w+h)	35.0	38.0	34.0	33.0	31.0	33.0	34.0	33.0	40.0	37.0	39.0	27.0
	cs-CORF(w+h)	**39.0**	41.0	**37.0**	**37.0**	**36.0**	36.0	**38.0**	**37.0**	**44.0**	41.0	**45.0**	**32.0**
MAE	RVM	0.94	0.82	1.07	0.77	0.72	1.02	0.63	0.64	0.72	0.84	0.68	0.70
	SR+SVM	0.88	0.77	1.07	0.60	0.65	0.74	0.69	0.52	0.59	0.77	0.63	0.51
	CRF(w)	0.92	0.95	1.02	0.62	0.70	0.91	0.57	0.50	0.60	0.74	0.68	0.56
	CORF(w+h)	0.85	0.75	0.90	0.63	0.63	0.78	0.60	0.54	0.60	0.83	0.64	0.52
	cs-CORF(w+h)	0.80	0.70	**0.82**	0.58	0.60	0.78	0.61	0.50	**0.51**	0.72	**0.57**	0.53
ICC	RVM	33.9	53.7	44.7	9.8	33.1	35.1	57.3	25.1	26.7	30.9	66.4	31.0
	SR+SVM	53.2	46.8	51.9	26.5	26.1	52.2	40.2	20.7	25.5	47.7	69.0	21.4
	CRF(w)	52.0	55.0	60.0	49.0	48.0	53.0	65.0	44.0	38.0	50.0	72.0	28.0
	CORF(w+h)	56.0	63.0	63.0	47.0	49.0	55.0	63.0	49.0	38.0	41.0	72.0	30.0
	cs-CORF(w+h)	**61.0**	**68.0**	**67.0**	51.0	**57.0**	58.0	66.0	51.0	**46.0**	49.0	**78.0**	**40.0**

(b) The DISFA dataset

Table 12.2: The performance of the models on intensity estimation of *pain* expression (P) and 11 AUs from the Sholder-Pain dataset, and 12 AUs from the DISFA dataset. The results are the averages of the 5-fold cross-validation procedure. We use bold face to indicate that the proposed cs-CORF(w+h) performs significantly better than the rest of the models, based on the paired t-test with $p = 0.05$.

muscles around the nose, which wrinkles the nose and pulls the nostril wings straight up. Due to the subtlety of these facial movements in naturalistic data and the involvement of vertically set muscles (rather than oblique ones), no strong personal characterization is expected in these AUs. Thus, modeling the context does not much improve the intensity estimation of AU9 and AU10. On the other hand, although AU20 involves horizontal motion (elongating the mouth), it often occurs in combination with other AUs (e.g., 10+20+25 or 20+26). Since these combinations are additive, cs-CORF(w+h) separates the effects of the target AU and those which co-occur by means of the CCE effects, resulting in the better performance by the proposed model. The impact of the context is also reflected in the intensity estimation of AU6. The activation of this AU wrinkles the skin around the outer corners of the eyes and raises the cheeks, so its detection/intensity estimation using the facial landmarks only is not possible in isolation from other AUs. However, the context of AUs makes it still possible to estimate

this AU since it usually appears in combination with other AUs (e.g., 6+12 is very common in naturalistic data and it represents a genuine smile). Although we do not explicitly model co-occurrences of different AUs, they are implicitly included in the CCE and FCE components.

It is also interesting that in the case of AU43, intensity estimation is better attained by using cs-CORF(w+h) than CRF(w), since there is no ordinal information as AU43 has only two levels (eyes open/closed). We ascribe this to modeling of the context and noise heteroscedasticity in the cs-CORF(w+h) model. Similarly, in the case of the DISFA dataset (Table 12.2 (b)), the proposed cs-CORF(w+h) achieves results that are similar or better than those of the other models. Nevertheless, compared to the Shoulder-pain dataset, some of the differences (e.g., AU12 and AU20) are not significant when $p = 0.05$ is used in the t-test. On the other hand, the intensity estimation of AU4 (brow lowerer) is significantly improved. We attribute this to the fact that AU4 is more subtle in the Shoulder-pain than in DISFA dataset, mainly because of the different context stimulus, resulting in fewer examples of the higher intensity levels of this AU in the Shoulder-pain dataset.

Analysis of the models' performance on AU6 and AU25. We choose these two AUs as examples to further demonstrate the performance of the models. Note that intensity estimation of AU6 is particularly challenging because it cannot be detected from facial landmarks alone as its inference relies on co-occurring AUs. On the other hand, AU25 can be detected from facial landmarks alone and is one of the most common facial actions that occurs involuntary in spontaneous facial displays. Fig.12.7 shows the confusion matrices (CMs) for different models. For each CM, we computed the OCI score, which low values indicate good performance (see Sec.12.3.1). In both cases, the cs-CORF(w+h) estimated the highest intensity levels more accurately compared to the CORF(w+h). By carefully inspecting the CMs, we note that in both of the ordinal models most confusion occurred between the neighboring intensity levels. This is in contrast to the other models, which exhibit a more 'dispersed' confusion of the intensity levels due to the lack of the ordinal monotonicity constraints. This is also reflected in their OCI scores. However, in some cases, the ordinal models confused the higher intensity levels with the neutral (the first column in the CMs), which occurred mainly when the input features were corrupted by the errors in facial landmark localization and/or registration. These were treated as outliers by the models, and therefore classified into the first bin on the ordinal line, corresponding to neutral intensity. However, cs-CORF(w+h) is more robust to such cases due to the use of the contextual information. It is also worth noting that since there is a small number of training examples for the highest intensity of AU25 from the DISFA dataset (i.e., less than 30), in this case all models failed to generalize to the unseen subjects (see Fig.12.7(b),

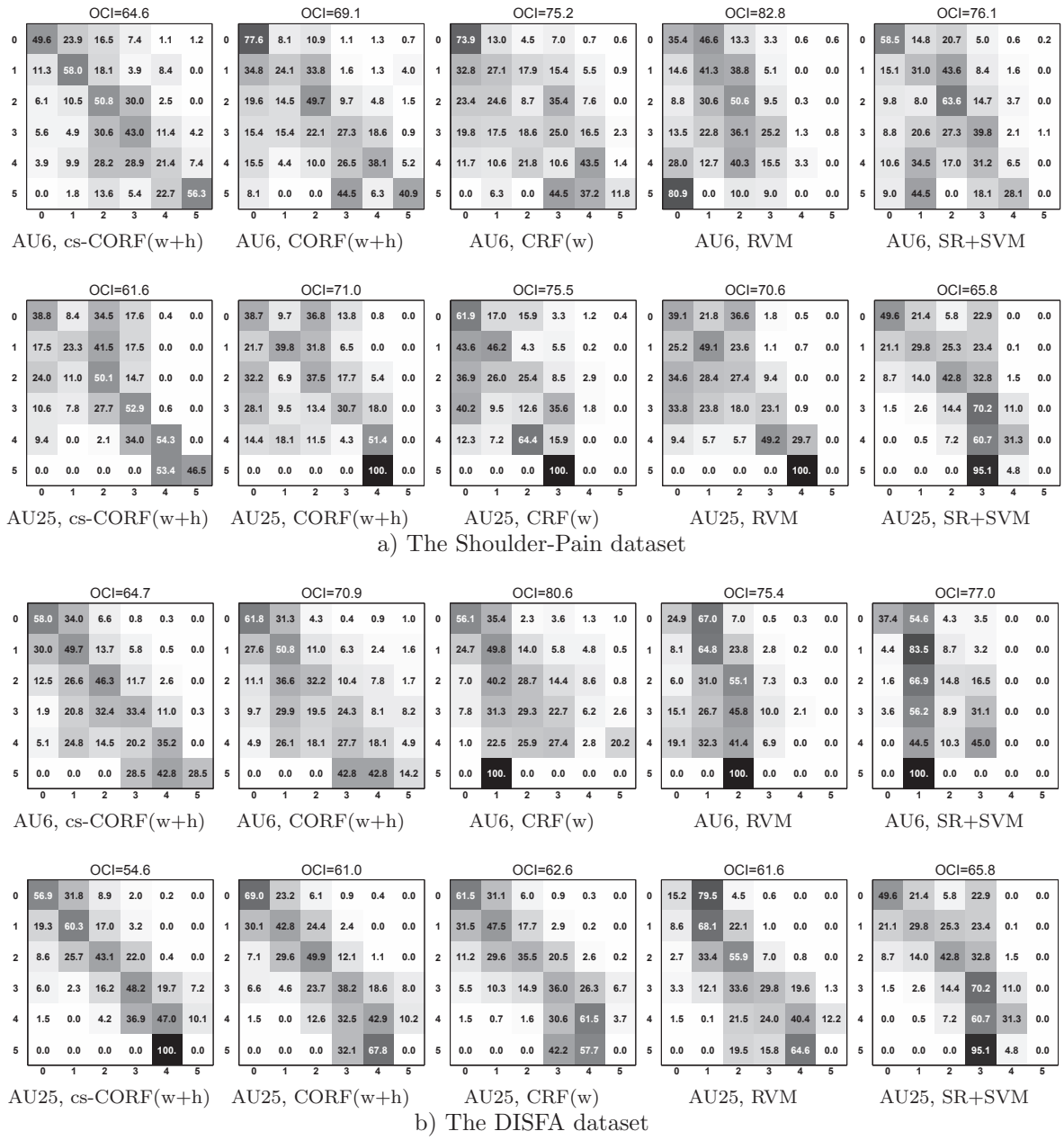


Figure 12.7: The (normalized) confusion matrices (CMs) computed from the true and predicted labels obtained by the denoted models applied for intensity estimation of AU6 and AU25 from the two datasets. The lower OCI score, the better performance.

bottom row).

Fig.12.8 shows examples of the intensity estimation at the sequence level for AU6 and AU25. The scores shown are computed from the depicted sequences. We see that the RVM model

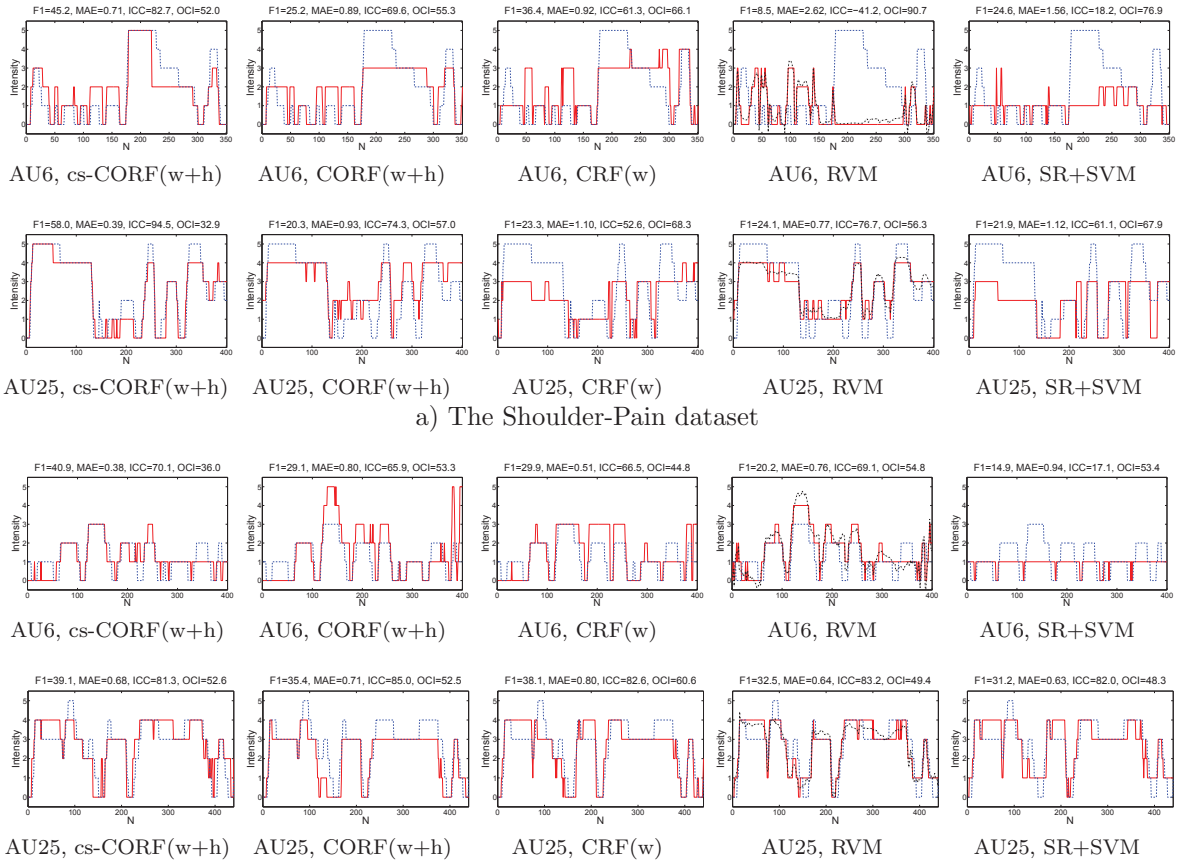


Figure 12.8: The true (*dashed blue*) and predicted (*solid red*) intensity of AU6 and AU25 from the two datasets. The sequences shown are obtained by concatenation of several exemplary sequences corresponding to different test subjects. The scores shown at the top of each figure are computed from the depicted sequences. For RVM, we also include the continuous estimation of AU intensity (*dashed black*).

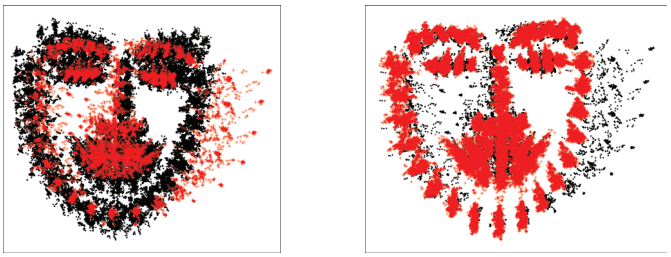
estimates the slope of the true signal well, but it misses its scale, which is a consequence of assuming an equal interval scale for the outputs. Note also from Fig.12.8(a) (RVM, AU6, frames 180–300) that this model estimates the whole sequence as having neutral intensity. This is because the input features were far from its kernel bases, which were selected during training. We also observe that SR+SVM underestimates the true intensity levels, which is possibly because of its bias toward the majority classes in the learned subspace. Based on the F1 scores for CRF(w), it outperforms CORF(w+h), however, CORF(w+h) achieves better MAE and ICC. This is expected because of the nature of the feature functions used in these two models (nominal vs. ordinal).

Cross-dataset evaluation. To test robustness of the models, we perform a cross-dataset evaluation. For this, the models were trained on DISFA dataset and tested on the Shoulder-pain dataset, and the other way round. Attaining a good performance in this setting is challenging mainly due to: (i) the difficulty of aligning the features of the two datasets, (ii) the bias in the ground-truth annotations of the two datasets, and (iii) the difference in the context stimulus (the pain inducing exercises vs. YouTube videos), which affects the frequency and co-occurrence of AUs, and thus the features to be selected. For this experiment, we used examples of 7 AUs (i.e., AU4, AU6, AU9, AU12, AU20, AU25 and AU26) that are present in both datasets. Registration of the facial landmarks between datasets was performed as explained in Fig.12.9.

From Table 12.3, we see that the performance of all models is lower for most of the AUs compared to that attained on the datasets used to train the models (see Table 12.2). This is expected because of the reasons (i)-(iii) mentioned above. From Fig.12.9 we also see that there is a different level of variation in the registered training/test points from the two datasets. This, in turn, negatively affects the performance of the models. Also, we note that in the case of AU6, cs-CORF(w+h) performs similarly to the other models. This is because the context stimulus in the two datasets is quite different, and so are the AUs co-occurrences that affect the features of this AU. As none of the models accounts for this difference in dynamics of AUs, they all achieve low performance. Furthermore, as we saw before, the estimation of AU20 was not significantly improved with the context modeling, and it is neither here. In the case of AU9 (nose wrinkler), modeling the context helps when training is conducted on the DISFA dataset and testing on the Shoulder-pain dataset, but not the other way round. This is caused by the inaccurate registration of the facial points around the nose area in the latter case (see Fig.12.9 on the left). Nevertheless, in the case of the other AUs (4, 12, 25 and 26), cs-CORF(w+h) consistently outperforms the other models. This is also reflected in the average results (over all AUs).

12.4 Conclusions

The results obtained indicate the benefits of the cs-CORF model for intensity estimation of AUs and expression of pain. Introducing the context and heteroscedastic effects in the probit function, used to define the node features in cs-CORF, is critical for the model's performance. In particular, modeling the context question *who* by inclusion of the CCE component substantially raises the performance of the traditional CORF across all three scoring measures. This is because the FCE component alone is unable to account for the presence of the context but



(a) (b)

Figure 12.9: Cross-dataset registration: (a) DISFA to Shoulder-pain, and (b) Shoulder-pain to DISFA. The reference face is calculated as the average of the points registered within the datasets (*red*) that are used to train the models. The registered points of the test dataset (*black*) are obtained by using an affine transform that maps the test points to the reference face of the training set. Note that in both cases the registration is imperfect, mainly because of large head-pose variation in the Shoulder-pain dataset, which cannot sufficiently be accounted for by using the affine transform.

Cross-dataset evaluation			AU4	AU6	AU9	AU12	AU20	AU25	AU26	Av.
F1	cs-CORF(w+h)	SP-D	**28.0**	29.0	25.0	**36.0**	27.0	**37.0**	**19.0**	**28.7**
	CORF(w+h)		25.0	**30.0**	**28.0**	31.0	**31.0**	35.0	14.0	27.7
	CRF(w)		22.0	22.0	**28.0**	34.0	29.0	28.0	16.0	25.5
	RVM		20.0	20.0	21.0	28.0	21.0	33.0	13.0	22.3
	SR+SVM		27.0	19.0	24.0	19.0	16.0	30.0	14.0	21.3
	cs-CORF(w+h)	D-SP	**26.0**	24.0	**39.0**	27.0	38.0	**43.0**	**29.0**	**32.3**
	CORF(w+h)		24.0	24.0	36.0	26.0	**41.0**	38.0	21.0	30.0
	CRF(w)		23.0	22.0	30.0	27.0	33.0	29.0	16.0	25.7
	RVM		22.0	17.0	0.09	24.0	17.0	19.0	16.0	16.4
	SR+SVM		21.0	**26.0**	33.0	**29.0**	31.0	30.0	20.0	27.1
MAE	cs-CORF(w+h)	SP-D	**1.24**	1.25	**1.14**	**0.72**	**0.92**	**0.80**	**1.34**	**1.05**
	CORF(w+h)		1.41	1.24	1.40	0.79	1.08	0.86	1.39	1.17
	CRF(w)		1.59	**1.21**	1.30	0.97	1.06	0.84	1.47	1.21
	RVM		1.57	1.44	1.47	1.05	1.07	0.81	1.62	1.29
	SR+SVM		1.53	1.78	1.54	1.12	1.36	1.13	1.38	1.41
	cs-CORF(w+h)	D-SP	**1.11**	**1.16**	**0.77**	**1.18**	1.04	**0.75**	**1.16**	**1.02**
	CORF(w+h)		1.26	1.25	0.87	1.33	0.95	0.82	1.40	1.13
	CRF(w)		1.20	1.44	1.06	1.34	**1.03**	1.02	1.40	1.21
	RVM		1.24	2.11	2.50	1.38	1.20	1.73	1.42	1.65
	SR+SVM		1.41	1.31	0.99	1.29	1.03	1.31	1.39	1.25
ICC	cs-CORF(w+h)	SP-D	**52.0**	47.0	49.0	**66.0**	**46.0**	**69.0**	27.0	**50.9**
	CORF(w+h)		48.0	**48.0**	**53.0**	62.0	38.0	65.0	28.0	48.8
	CRF(w)		37.0	37.0	44.0	58.0	40.0	57.0	28.0	43.0
	RVM		32.0	34.0	27.0	56.0	25.0	51.0	22.0	35.3
	SR+SVM		41.0	13.0	34.0	44.0	12.0	44.0	**30.0**	31.1
	cs-CORF(w+h)	D-SP	**42.0**	**45.0**	**74.0**	**55.0**	36.0	**62.0**	**27.0**	**48.7**
	CORF(w+h)		37.0	41.0	68.0	50.0	**37.0**	55.0	17.0	43.6
	CRF(w)		37.0	37.0	62.0	41.0	35.0	51.0	15.0	39.7
	RVM		37.0	0.07	0.00	39.0	15.0	25.0	-0.03	16.6
	SR+SVM		25.0	33.0	60.0	39.0	34.0	37.0	26.0	36.3

Table 12.3: Cross-datasets evaluation of the models on 7 AUs present in both datasets. The models are trained using the data of the target AUs from the Shoulder-pain (SP) dataset, and tested on the data from the DISFA (D) dataset (SP-D), and the other way round (D-SP). We use bold face to highlight the scores of the best performing models for the given task. On average, the proposed cs-CORF model outperforms the rest of the models on all tasks.

also cannot result in its full removal. This is also true because of the heteroscedastic nature of the data, encoded both in variance and the offset, and the modeling of which is important for a proper adaptation of the model to different subjects. On the other hand, we conclude that 'naive' inclusion of the CCE covariates in the non-ordinal models does not improve their overall performance. The main reason for this lies in their lack of parameter tying, i.e., the influence of the CCE and FCE components on each response (i.e., the intensity level) is modeled independently. By contrast, the CCE/FCE-related parameters and the ordinal thresholds in the cs-CORF model act in concert, with the former helping to adjust the location and scale of the latter depending on the input. This, in turn, allows the model to distinguish between distinct motion patterns of AUs, some of which exhibit strong personal characterization. Also, in situations where the feature registration has not been fully achieved, and where only a small amount of data is available for training, as in the case of the Shoulder-pain dataset, the inclusion of the CCE component increases robustness of the model.

It is also important to mention that while the CRF nominal model, which is commonly used for context modeling in other domains (e.g., [89, 215]), performs rather well, it fails to reach full performance level of the cs-CORF model. This is in part due to the lack of the ordinal monotonicity constraints and in part due to the to the increased parameter dimensionality. Regarding the former, the misclassification away from true labels incurs higher cost in ordinal regression compared to the label-distance agnostic classification of CRFs. Similar reasoning can be applied to analysis of performance of the static nominal models such as multi-class SVM. Likewise, standard regression models like RVM are unfit for modeling ordinal responses due to their implicit assumption of an interval scale [214]. Furthermore, the traditional methods for sequence classification and AU intensity estimation are designed for balanced data. Yet, because of the imbalanced nature of our data, proper scaling of the loss during training is necessary. The most frequent low intensity levels that would otherwise dominate performance scores are properly balanced using the proposed weighted softmax-margin learning for CRFs. This is reflected in improvements of the weighted models (**w**) over their unweighted counterparts (**ml**). Finally, while standard static ordinal models such as GPOR and SVOR provide a solid framework for modeling ordinal data, the class imbalance and the lack of temporal constraints adversely affect their learning and inference. Consequently, they cannot take full advantage of the CCE component. This is all successfully remedied by the proposed cs-CORF model.

To conclude, in this Chapter we have proposed a novel method for intensity estimation of AUs and pain from spontaneously displayed facial expressions. We have addressed the lim-

itations of the state-of-the-art approaches that do not leverage the ordinal structure in the expression intensity, and also fail to account for influence of the context as well as heteroscedastic and imbalanced nature of the expression intensity data. We have shown on the data of spontaneously displayed facial expressions that our approach substantially outperforms the state-of-the-art methods for intensity estimation of AUs and pain.

Chapter 13

Analysis of Facial Expression Dynamics: Conclusions and Future Work

In this part of the thesis, we have proposed different models for analysis of facial expression dynamics from image sequences. In particular, we have focused on two aspects of facial expression dynamics: temporal segments and intensity of facial expressions. We based our models on the Conditional Random Field (CRF) framework for structured learning of image sequences. In these models, we encoded the spatio-temporal structure in image sequences of facial expressions. We achieved this by accounting for ordinal relationships between temporal segments and also intensity levels of target expressions, as well as their dependencies in the temporal domain. We also explored several means of addressing the subject variability in the data by simultaneously exploiting various priors, and the effects of heteroscedasticity and context of target facial expressions. In this way, we solved some of the important challenges of automated analysis of facial expression dynamics. The proposed models can discriminate successfully between different temporal segments and also intensity levels of spontaneously displayed facial expressions and AUs of different subjects. In contrast to the existing models, which do not account for the effects mentioned above, our models achieve substantially better temporal segmentation and intensity estimation of facial expressions and AUs. In what follows, we discuss the proposed contributions and give directions for future research.

Chapter 9 While most of the state-of-the-art dynamic methods focus on classification of facial expressions only, in the proposed MCORF model, we also modeled their underlying dynamics, driven by temporal segments of emotion expression. We accomplished this within

the unified framework that performs simultaneous classification and temporal segmentation of facial expressions. In this way, we not only facilitated classification of emotion categories but also identified their temporal segments within sequences. We also showed that by employing the MAP strategy to learn the parameters of our MCORF model resulted in the model that is largely invariant to subject differences. This is mainly attributed to the graph Laplacian prior that we designed based on our domain knowledge, and placed over the model parameters. This resulted in the task-specific regularizer, the role of which turned out to be crucial for performance of the model.

Chapter 10 By using the regularizer mentioned above, the explicit mappings in the ordinal feature functions of the MCORF model become a linear approximation of the otherwise non-linear mappings functions. However, in the case of high-dimensional inputs (e.g., the appearance-based features), learning of the explicit mappings inevitably results in a large number of the model parameters. This can easily lead to overfitting. We addressed this in the KCORF model, where we introduced fully non-linear feature mappings, which permit the use of implicit feature spaces through Mercer kernels. We showed on the task of temporal segmentation of AUs that this model outperforms the MCORF and CORF models with linear mappings in the ordinal feature functions. Furthermore, for the KCORF model, we proposed the composite kernel function that allowed the model to automatically select regions of a face that are highly relevant for temporal segmentation of target AUs. This resulted in the model being less prone to overfitting, and, thus, better able to generalize to test subjects, compared to the existing methods. In comparison to other kernel-based models, KCORF employs the best of static kernel models for ordinal data, such as GPOR and SVOR, and dynamic kernel models for nominal data, such as KCRF and the GP formulation of discriminative learning of sequences [5]. This is because the former perform static modeling of data, while the latter fail to account for their ordinal structure - two types of structure that we successfully accounted for in our model.

Chapter 11 The KCORF$_h$ model that we proposed for intensity estimation of spontaneous facial expressions of pain further generalizes our KCORF model by relaxing its assumption of having constant variance in the ordinal feature functions. This allowed the inputs (i.e., facial features of various subjects) to exert a different influence on the location and thresholds of the ordinal feature functions. This, in turn, resulted in the KCORF$_h$ model being able to adapt better to the varying facial expressiveness levels of different subjects, compared to its homoscedastic counterparts. Our experimental results evidence that this extra degree of freedom in the KCORF$_h$ model is important for capturing subtle changes in spontaneously

displayed facial expressions, and in particular, those caused by their intensity variation.

Chapter 12 Finally, the cs-CORF model for intensity estimation of spontaneous AUs and facial expressions of pain, generalizes our contributions mentioned above by also exploiting the context in which target facial expressions occur. In contrast to KCORF$_h$, this model accounts for subject variability in the data by also allowing the subject-specific biases, in addition to the changing variance, to influence the model parameters. This is achieved by an explicit modeling of the context factors *who* (the observed subject), *how* (changes in facial expressions), and *when* (timing of facial expression intensity). In particular, we showed that modeling the context factor *who* is important for raising the performance of the CORF model and its heteroscedastic counterpart, on the target tasks. This evidences that applying the Laplacian regularization and/or using the changing variance is not as effective for attenuating the subject differences as when both the subject-specific biases and the changing variance are allowed to directly influence the model parameters, as in cs-CORF. This is especially true when estimating the higher intensity levels, the facial features of which vary greatly across subjects. Also, in contrast to the models mentioned above, and the state-of-the-art models for the intensity estimation, our cs-CORF model performs robust parameter learning from a skewed distribution of the intensity levels using the introduced weighted softmax-margin learning. This also contributed to improving estimation of the less frequently occurring intensities, i.e., the higher intensity levels.

Future work. The proposed methods exploit modeling strategy of static ordinal regression models and manifold learning techniques within the CRF framework, in order to model different types of data structure. While these methods have rich representational power, as for most discriminative models based on CRFs, their performance on the target tasks relies heavily on parameter regularization. For this, we used the standard validation procedure. However, this may be time consuming as the number of regularization parameters to be tuned increases. A way to address this is to perform the parameter coupling in order to reduce their search space, e.g., as in [20]. Nevertheless, this is an open problem in machine learning that poses a bottleneck of many existing machine-learning algorithms.

One important issue that we have not investigated in the methods proposed is how to model higher-level temporal dependencies in data in order to better capture underlying dynamics of target tasks. For this, the CRF modeling approach in [35], for instance, can be explored to re-define the energy function of our models so that it entails infinitely-long time dependencies between the data. Also, since manual annotation of temporal segments and intensity levels of facial expressions is labor intensive, we may end up with partially labeled data. In this case,

the graph Laplacian regularization that we introduced can be extended as in [21] in order to carry out semi-supervised learning, and thus make the use of unlabeled data for improving the generalization power of the models. While these are only a few of many possible extensions, they are applicable to all the methods proposed. In what follows, we briefly comment on limitations and possible extensions of each method.

In the method for simultaneous classification and temporal segmentation of facial expressions in Chapter 9, the size of the emotion manifold on which the MCORF parameters are learned is determined using a validation procedure. However, this can be accomplished automatically by minimizing the rank of the manifold, as in [165]. Also, since we optimize classification of emotions and their temporal segments simultaneously, using the LBFGS method, we can easily end up in a local minimum. An alternative to this is to design an Expectation-Maximization-like algorithm where learning of the top layer (i.e., emotions) and the intermediate layer (i.e., temporal segments) is divided into two steps. This method can further be improved by including context-factors such as the observed subject in its node features, as we did in our cs-CORF model. Similarly, the explicit mappings we used to find the emotion manifold can be kernelized using the approach presented in Chapter 10.

The main challenge in the KCORF model is selection of the kernel bases for the target task. Although for simplicity we randomly selected a number of kernel bases, more sophisticated approaches can be used. For instance, the kernel selection approach used in KCRFs [109] can be adapted to our model to incrementally select kernels by greedily reducing the regularized cost. Another approach to achieve the kernel sparsity is to explore the kernel structure (e.g., as in [5]). Note also that we independently optimized the weights of the kernel bases and those of our CHI kernel, used to weight face regions based on their relevance for each AU. However, by carefully analyzing this kernel structure (e.g., as in [5]), more efficient and robust learning of the kernel weights could be achieved. To attain robustness with respect to changes in illumination, the face occlusions (e.g., by hand), and other sources of noise and outliers in the data, the robust kernels (e.g., [197, 119]) can be exploited within the KCORF model.

While the extensions mentioned above are also applicable to the KCORF$_h$ model, it would be particularly interesting to see how different functional forms (and noise distributions) for modeling hetersocedasticity affect performance of this model. In the current method we used the same functional forms for the location and scale model in the ordinal feature functions of KCORF$_h$, and we observed that the parameter estimation can sometimes be fragile, as pointed out in [214]. A thorough analysis of behavior of KCORF$_h$ when different noise assumptions and functional forms (or the kernel function) for the ordinal scale are used would help to take

full advantage of this approach. This could help to capture subtle differences in the facial expressions of different subjects even better.

Lastly, we found in our experiments that the cs-CORF model is sometimes prone to overfitting of the context question *who*. Thus, a careful regularization of its parameters, as mentioned above, is needed to ensure that the model generalizes well to test subjects. Furthermore, modeling of the context question *who* can be improved by exploiting the subject's attributes such as the gender or age, which can be estimated using independently trained models (e.g., see [97]). Similarly, modeling of the context question *when* can be improved by accounting for co-occurrences of different AUs and their intensities in time. A simple approach would be to train independent SVM models for detection of each AU, and use their outputs as additional input features (x^{when}) to our AU-specific context models. Another alternative is, for instance, to model intensities of co-occurring AUs within our context model by using the notion of factorial CRFs [185]. It is also important to investigate the impact of the other context questions (i.e., *what*,*where* and *why*) on the target tasks. For instance, the context question *what* can be answered by using the SVM classifier to determine whether the subject's focus of attention is another person or a computer. The binary output of this classifier can be then used to form x^{what} in our cs-CORF model. Likewise, the context question *where* can be answered by determining the subject's head-pose (or his/her location in a scene), which estimate can be used as x^{where}. The context question *why* is perhaps the most complex as it depends highly on the other context questions. A way to account for this question is to determine whether the displayed facial expression is posed or spontaneous (e.g., as in [199]), and use an indicator of this to form x^{why}. Nevertheless, there are many possible ways to answer the above mentioned context questions, and while we suggested only simple few, the future research should explore more compelling ways of doing this. Finally, since the proposed cs-CORF model is linear, this can pose a limitation when dealing with high-dimensional input features (i.e, the appearance features). This can be addressed by kernelizing cs-CORF using the same approach as in KCORF, and by defining its kernel function as a weighted sum of kernels designed particularly for each context question.

Chapter 14

Final Conclusions

In this thesis, we proposed various machine learning algorithms for addressing two important problems of automated analysis of facial expressions. The first problem that we addressed is pose-invariant facial expression recognition. For this, we proposed novel models for pose normalization that achieve decoupling of head pose and expression in the case of large out-of-plane head rotations. This is followed by classification of the pose normalized facial expressions into target expression categories. We explored different types of spatial structure of facial expression data by means of priors and constraints, which we efficiently incorporated into the Gaussian Process framework to obtain our models. These models solve some of the most important challenges of pose-invariant facial expression classification by being able to generalize to various poses and expressions from a small amount of training data, while also being largely robust to corrupted image features and imbalanced examples of different facial expression categories. We showed that these models perform accurate pose-invariant facial expression classification of the six basic emotions, considerably outperforming the existing approaches, which fail to address the challenges mentioned above.

The second problem that we addressed in this thesis is automated analysis of dynamics of facial expressions and AUs, in terms of their temporal segments and intensity levels. For this, we proposed novel models that are based on the Conditional Random Field framework for structured learning of image sequences. In these models, we encoded the spatio-temporal structure in image sequences of facial expressions. We achieved this by accounting for ordinal relationships between temporal segments and also intensity levels of target expressions, as well as their dependencies in the temporal domain. We also explored several means of addressing the subject variability in the data by simultaneously exploiting various priors, and the effects of heteroscedasticity and context of target facial expressions. All this resulted in the models that

are able to capture subtle variation in spontaneously displayed facial expressions and AUs of different subjects, and thus, discriminate successfully between different temporal segments and also intensity levels of their facial expressions. In contrast to other existing models, which do not account for the effects mentioned above, our models achieve substantially better temporal segmentation and intensity estimation of target facial expressions and Action Units.

Taken together, the methods proposed in this thesis solve some of the most important challenges in pose-invariant facial expression recognition and analysis of facial expression dynamics. Looking into the future, it is evident that this research can serve as a basis for further work on automated analysis of facial expressions. For instance, addressing pose-invariant analysis of facial expression dynamics is a natural step forward. Exploring more effective ways of facial feature extraction and selection for target tasks is another direction to pursue. This would facilitate learning and improve generalization of the models for facial expression analysis. Also, evaluation of these models using data recorded in natural environments, where more realistic illumination conditions and head movements are present, would help to make practical use of the models but also to identify new modeling challenges. Fortunately, the field of machine learning is progressing rapidly, allowing us to deal effectively with all these. However, when designing target models, it should be remembered that facial expressions and their dynamics are bound by different physical constraints and environmental factors. For this reason, context-sensitive modeling of facial expressions is perhaps the most promising way to achieve fully automated facial expression analysis. We believe that the research presented in this thesis represents a significant step towards accomplishing that goal.

Bibliography

[1] A. Agresti. *Analysis of ordinal categorical data*. Wiley Series in Probability and Statistics, 1984. 117, 118

[2] N. Ahmed, T. Natarajan, and K. R. Rao. Discrete cosine transform. *IEEE Transactions on Computers*, 23:90–93, 1974. 34

[3] T. Ahonen, A. Hadid, and M. Pietikainen. Face description with local binary patterns: Application to face recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(12):2037–2041, 2006. 104, 147

[4] J. Alabort, G. Tzimiropoulos, S. Zafeiriu, and M. Pantic. Generic active appearance models revisited. In *Asian Conference on Computer Vision (ACCV)*, pages 650–663, 2012. 31

[5] Y. Altun. *Discriminative methods for label sequence learning*. PhD thesis, Brown University, 2005. 130, 188, 190

[6] Y. Altun and T. Hofmann. Large margin methods for label sequence learning. *Proc. European Conf. on Speech Communication and Tech. (EuroSpeech)*, 2003. 171

[7] M. Alvarez and N. Lawrence. Sparse convolved multiple output gaussian processes. In *Neural Inf. Proc. Systems (NIPS)*, pages 57–64, 2008. 86, 95

[8] Z. Ambadar, J. F. Cohn, and L. I. Reed. All smiles are not created equal: Morphology and timing of smiles perceived as amused, polite, and embarrassed/nervous. *J. Nonverbal Behavior*, 33:17–34, 2009. 12

[9] Z. Ambadar, J. Schooler, and J.F. Cohn. Deciphering the enigmatic face: The importance of facial dynamics to interpreting subtle facial expressions. *Psychological Science*, 16(5):403–410, 2005. 16, 125

[10] A. M. Amin, N. V. Afzulpurkar, M. N. Dailey, V. Esichaikul, and D. N. Batanov. Fuzzy-c-mean determines the principle component pairs to estimate the degree of emotion from facial expressions. In *Fuzzy Systems and Knowledge Discovery*, volume 3613, pages 484–493, 2005. 46

[11] A. Asthana, S. Cheng, S. Zafeiriou, and M. Pantic. Robust discriminative response map fitting with constrained local models. In *Int'l Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2013. 31

[12] A. Asthana, R. Goecke, N. Quadrianto, and T. Gedeon. Learning based automatic face annotation for arbitrary poses and expressions from frontal images only. In *Int'l Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1635–1642, 2009. 44, 45

[13] S. Baccianella, A. Esuli, and F. Sebastiani. Evaluation measures for ordinal regression. *Int'l Conf. on Intell. Syst. Design and Applications*, pages 283–287, 2009. 136

[14] A. Barla, F. Odone, and A. Verri. Histogram intersection kernel for image classification. *Int'l Conf. on Image Processing (ICIP)*, 3:513–16, 2003. 147

[15] M.S. Bartlett and J. Whitehill. *Automated Facial Expression Measurement: Recent Applications to Basic Research in Human Behavior, Learning, and Education.* Handbook of Face Perception. Oxford University Press., 2010. 18

[16] M.S. Bartlett, G. Littlewort, M. G. Frank, C. Lainscsek, Ian R. Fasel, and Javier R. Movellan. Automatic recognition of facial actions in spontaneous expressions. *Journal of Multimedia*, 1(6):22–35, 2006. 43

[17] M.S. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, and J. Movellan. Fully automatic facial action recognition in spontaneous behavior. *Automatic Face and Gesture Recognition (FG)*, pages 223 –230, 2006. 14, 32, 34, 37, 48

[18] M.S. Bartlett, Gwen Littlewort, M. Frank, C. Lainscsek, Ian Fasel, and J. Movellan. Recognizing facial expression: machine learning and application to spontaneous behavior. In *Int'l Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 568–573 vol. 2, 2005. 37

[19] J.J. Bazzo and M.V. Lamar. Recognizing facial actions using gabor wavelets with neutral face average difference. In *Automatic Face and Gesture Recognition (FG)*, pages 505–510, 2004. 37

[20] M. Belge, M.E. Kilmer, and E.L. Miller. Efficient determination of multiple regularization parameters in a generalized l-curve framework. *Inverse Problems*, 18:2002, 2002. 189

[21] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006. 146, 190

[22] J.C. Bezdek and R.J. Hathaway. Some notes on alternating optimization. In *Advances in Soft Computing — AFSS 2002*, volume 2275, pages 288–300. 2002. 111

[23] C.M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., 2006. 37, 58, 59, 60, 63, 69, 85, 100, 102, 104, 131, 135

[24] M.J. Black and A.D. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *Int'l Journal of Computer Vision*, 26(1):63–84, 1998. 31

[25] M.J. Black and Y. Yacoob. Recognizing facial expressions in image sequences using local parameterized models of image motion. *Int'l J. of Computer Vision*, 25:23–48, 1997. 12, 37

[26] V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25:1063–1074, 2003. 44

[27] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proc. of the 11th Annual Conf. on Comp. Learning Theory*, pages 92–100, 1998. 112

[28] L. Bo and C. Sminchisescu. Twin gaussian processes for structured prediction. *Int'l Journal of Computer Vision*, 87(1-2):28–52, 2010. 61, 72, 86, 89

[29] E.V. Bonilla, K.M. Chai, and C. Williams. Multi-task Gaussian process prediction. In *Neural Information Processing Systems (NIPS)*, 2008. 72

[30] P. Boyle and M. Frean. Dependent gaussian processes. In *Neural Information Processing Systems (NIPS)*, pages 217–224, 2005. 86, 95

[31] D. Cai, X. He, and J. Han. Spectral regression for efficient regularized subspace learning. *Proc. Int'l Conf. Computer Vision (ICCV)*, pages 1–8, 2007. 48, 174

[32] J.S. Cardoso and R. Sousa. Measuring the performance of ordinal classification. *Int'l Journ. of Pattern Recognition and Artificial Intell.*, 25(8):1173–1195, 2011. 136, 175

[33] C. Chang and C. Lin. Libsvm: A library for support vector machines. In *ACM Transactions on Intelligent Systems and Technology*, pages 1–27, 2011. 72, 159, 174

[34] K. Chang, T. Liu, and S. Lai. Learning partially-observed hidden conditional random fields for facial expression recognition. In *Int'l Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 533–540, 2009. 38, 39

[35] S.P. Chatzis and Y. Demiris. The infinite-order conditional random field model for sequential data modeling. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 35(6):1523–1534, 2013. 189

[36] S.W. Chew, P. Lucey, S. Lucey, J. Saragih, J.F. Cohn, I. Matthews, and S. Sridharan. In the pursuit of effective affective computing: The relationship between features and registration. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 42(4):1006–1016, 2012. 37

[37] W. Chu and Z. Ghahramani. Gaussian processes for ordinal regression. *Journal of Machine Learning Research*, 6:1019–1041, 2005. 46, 116, 117, 118, 119, 136, 159, 174

[38] W. Chu and S. Sathiya Keerthi. New approaches to support vector ordinal regression. *Int'l Conf. on Machine Learning (ICML)*, pages 145–152, 2005. 46, 116, 117, 118, 119, 159, 174

[39] W. Chu, F. De la Torre, and J.F. Cohn. Selective transfer machine for personalized facial action unit detection. In *Int'l Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3515–3522, 2013. 14, 37

[40] F R.K. Chung. Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, No. 92). *American Mathematical Society*, 1996. 101, 128

[41] I. Cohen, N. Sebe, L. Chen, A. Garg, and T. Huang. Facial expression recognition from video sequences: Temporal and static modelling. In *Computer Vision and Image Understanding*, pages 160–187, 2003. 33, 34, 37, 38, 45

[42] J.F. Cohn and P. Ekman. Measuring facial actions. In *The New Handbook of Methods in Nonverbal Behavior Research, Harrigan, J.A., Rosenthal, R. & Scherer, K., Eds.*, pages 9–64. Oxford University Press, 2005. 9, 11

[43] J.F. Cohn and K.L. Schmidt. The timing of facial motion in posed and spontaneous smiles. *Int'l Journal of Wavelets, Multiresolution and Inf. Processing*, 2(2):121–132, 2004. 16

[44] T.F. Cootes and C.J. Taylor. Active shape models - smart snakes. In *British Machine Vision Conf. (BMVC)*, pages 266–275, 1992. 73, 85

[45] T.F. Cootes and C.J. Taylor Statistical models of appearance for computer vision. *World Wide Web Publication*, 2001. 44

[46] T.F Cootes, G.V Wheeler, K.N Walker, and C.J Taylor. View-based active appearance models. *Image and Vision Computing*, 20(9-10):657 – 664, 2002. 44

[47] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292, 2001. 119

[48] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Int'l Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 886–893. IEEE, 2005. 34

[49] C. Darwin. *The Expression of the Emotions in Man and Animals*. John Murray, 1872. 9

[50] F. De la Torre and M.J. Black. A framework for robust subspace learning. *Int. J. Comput. Vision*, 54(1-3):117–142, 2003. 86

[51] F. De la Torre and J.F. Cohn. *Guide to Visual Analysis of Humans: Looking at People*, chapter Facial Expression Analysis. Springer, 2011. 9, 31, 33, 36

[52] J.R. Delannoy and J. McDonald. Automatic estimation of the dynamics of facial expression using a three-level model of intensity. In *Automatic Face and Gesture Recognition (FG)*, pages 1–6, 2008. 47, 48

[53] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, December 2006. 177

[54] G. Donato, M.S. Bartlett, J.C. Hager, P. Ekman, and T.J. Sejnowski. Classifying facial actions. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 21(10):974–989, 1999. 37

[55] F. Dornaika and F. Davoine. Simultaneous facial action tracking and expression recognition in the presence of head motion. *Int'l Journal of Computer Vision*, 76(3):257–281, 2008. 43

[56] F. Dornaika and J. Orozco. Real time 3d face and facial feature tracking. *J. Real-Time Image Processing*, 2(1):35–44, 2007. 31, 71, 73, 79, 80, 81, 90, 93, 94, 135, 140

[57] G.B. Duchenne de Bologne. *The Mechanism of Human Facial Expression*. Cambridge University Press, 1990. 15

[58] C.H. Ek. *Shared Gaussian Process Latent Variable Models*. PhD thesis, Oxford Brookes University, 2009. 25, 56, 98, 99, 100, 112

[59] P. Ekman. Darwin, deception, and facial expression. *Annals of the New York Academy of Sciences*, 1000:205–221, 2003. 12, 13, 16

[60] P. Ekman, W.V. Friesen, and J.C. Hager. *Facial Action Coding System (FACS): Manual*. A Human Face, 2002. 9, 11, 12, 14, 15, 16, 143, 164

[61] P. Ekman and L. E. Ronsenberg. *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System*. Oxford University Press, 2005. 9, 12, 13, 16

[62] P. Ekman and W.V. Friesen. Constants across cultures in the face and emotion. *Journal of Personality and Social Psychology*, 17:124–129, 1971. 9, 11

[63] P. Ekman and W.V. Friesen. *Unmasking the face: A guide to recognizing emotions from facial clues*. Prentice Hall, Oxford, 2003. 9

[64] B. Fasel and J. Luettin. Recognition of asymmetric facial action unit activities and intensities. In *Int'l Conf. on Pattern Recognition (ICPR)*, volume 1, pages 1100–1103 vol.1, 2000. 37

[65] M. G. Frank and P. Ekman. The ability to detect deceit generalizes across different types of high-stakes lies. *Journal of Personality and Social Psychology*, 72(6):1429–1439, 1997. 12

[66] B. Gholami, W.M. Haddad, and A.R. Tannenbaum. Agitation and pain assessment using digital imaging. In *Int'l Conf. of the Engineering in Medicine and Biology Society*, pages 2176–2179, 2009. 37

[67] D. Gill, O.G. Garrod, R.E. Jack, and P.G. Schyns. From facial gesture to social judgment: a psychophysical approach. *J. Nonverbal Behav.*, 3(6):395, 2012. 16

[68] K. Gimpel and N.A. Smith. Softmax-margin crfs: training log-linear models with cost functions. *Human Language Technologies: Annual Conf. of the North American Chapter of the Association for Comp. Linguistics*, pages 733–736, 2010. 171

[69] M. Gönen and E. Alpaydin. Multiple kernel learning algorithms. *Journal of Machine Learning Research*, 12:2211–2268, 2011. 147

[70] W.H. Greene. *Econometric Analysis*. Prentice Hall, 2002. 119

[71] K. Grochow, S.L. Martin, A. Hertzmann, and Z. Popović. Style-based inverse kinematics. In *Proc. ACM Int'l Conf. on Computer Graphics and Interactive Techniques*, pages 522–531, 2004. 65

[72] R. Gross, I. Matthews, and S. Baker. Generic vs. person specific active appearance models. *Image and Vision Computing Journal*, 23:1080–1093, 2005. 44

[73] R. Gross, I. Matthews, J.F. Cohn, T. Kanade, and S. Baker. Multi-pie. *Image and Vision Computing Journal*, 28(5):807–813, 2010. 42, 70, 89, 104

[74] A. Gunawardana, M. Mahajan, A Acero, and Platt J. C. Hidden conditional random fields for phone classification. *Int'l Conf. on Speech Comm. and Techn.*, 2005. 122

[75] H. Gunes and M. Piccardi. Automatic temporal segment detection and affect recognition from face and body display. *IEEE Trans. on Systems, Man, and Cybernetics*, 39(1):64–84, 2009. 45

[76] J. Hamm, C.G. Kohler, R.C. Gur, and R. Verma. Automated facial action coding system for dynamic analysis of facial expressions in neuropsychiatric disorders. *Journal of Neuroscience Methods*, 200(2):237 – 256, 2011. 37

[77] Z. Hammal and J.F. Cohn. Automatic detection of pain intensity. *Int'l Conf. on Multimodal Interfaces (ICMI)*, pages 47–52, 2012. 47

[78] X. He and P. Niyogi. Locality preserving projections. *Neural Inf. Proc. Systems (NIPS)*, 2004. 41, 103, 104, 129, 132, 148, 159, 160

[79] M. Heller and V. Haynal. The faces of suicidal depression. *Kahiers Psychiatriques Genevois*, 16:107–117, 1994. 12

[80] U. Hess, S. Blairy, and R.E. Kleck. The relationship between the intensity of emotional facial expressions and observers' decoding. *Journal of Nonverbal Behavior*, 21(4):241–257, 1997. 15

[81] N. Hesse, T. Gehrig, H. Gao, and H.K. Ekenel. Multi-view facial expression recognition using local appearance features. In *Int'l Conf. on Pattern Recognition (ICPR)*, pages 3533–3536, 2012. 34, 35, 44

[82] C. Hu, Ya Chang, R. Feris, and M. Turk. Manifold based analysis of facial expression. In *Int'l Conf. on Computer Vision and Pattern Recognition (CVPR'W)*, pages 81–81, 2004. 33, 40

[83] Y. Hu, Z. Zeng, L. Yin, X. Wei, J. Tu, and T.S. Huang. A study of non-frontal-view facial expressions recognition. In *Int'l Conf. on Pattern Recognition (ICPR)*, pages 1–4, 2008. 42, 77, 78, 104

[84] Y. Hu, Z. Zeng, L. Yin, X. Wei, Xi Zhou, and T.S. Huang. Multi-view facial expression recognition. In *Automatic Face and Gesture Recognition (FG)*, pages 1–6. IEEE, 2008. 34, 35, 42, 77, 78

[85] S. Jain, C. Hu, and J.K. Aggarwal. Facial expression recognition with temporal modeling of shapes. In *Proc. Int'l Conf. Computer Vision (ICCV'W)*, pages 1642–1649, 2011. 38

[86] L.A. Jeni, J.M. Girard, J.F. Cohn, and F. De La Torre. Continuous au intensity estimation using localized, sparse facial feature space. *Automatic Face and Gesture Recognition (FG)*, pages 1–7, 2013. 47, 48

[87] L.A. Jeni, A. Lőrincz, T. Nagy, Z. Palotai, J. Sebők, Z. Szabó, and D. Takács. 3d shape estimation in video sequences provides high precision evaluation of facial expressions. *Image and Vision Computing*, 2012. 44

[88] B. Jiang, M. F. Valstar, and M. Pantic. Action unit detection using sparse appearance descriptors in space-time video volumes. *Automatic Face and Gesture Recognition (FG)*, pages 314–321, 2011. 37, 147

[89] W. Jiang, S. Chang, and A.C. Loui. Context-based concept fusion with boosted conditional random fields. *Int'l Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2007. 185

[90] T. Joachims, T. Finley, and C.J. Yu. Cutting-plane training of structural svms. *Journal of Machine Learning*, 77(1):27–59, 2009. 165, 170, 171

[91] S.J. Julier and J.K. Uhlmann. A non-divergent estimation algorithm in the presence of unknown correlations. In *Proc. American Control Conf.*, pages 2369–2373, 1997. 67

[92] S. Kaltwang, O. Rudovic, and M. Pantic. Continuous pain intensity estimation from facial expressions. *Int'l Symposium on Visual Computing (ISVC)*, 7432:368–377, 2012. 32, 33, 34, 35, 36, 47, 48, 158, 174, 178

[93] M. Kan, S. Shan, H. Zhang, S. Lao, and X. Chen. Multi-view discriminant analysis. In *European Conf. on Computer Vision (ECCV)*, pages 808–821. 2012. 103

[94] T. Kanamori. Statistical models and learning algorithms for ordinal regression problems. *Journ. of Inf. Fusion*, 14(2):199–207, 2013. 118, 174

[95] A. Kapoor, H. Ahn, and R.W. Picard. Mixture of gaussian processes for combining multiple modalities. In *Multiple Classifier Systems*, volume 3541 of *Lecture Notes in Computer Science*, pages 86–96. 2005. 111

[96] A. Kapoor, Y. Qi, and R.W. Picard. Fully automatic upper facial action recognition. In *Automatic Face and Gesture Recognition (FG'W)*, pages 195–202, 2003. 33, 34

[97] V. Karimi and A. Tashk. Age and gender estimation by using hybrid facial features. In *Telecommunications Forum (TELFOR)*, pages 1725–1728, 2012. 191

[98] D. Keltner and P. Ekman. *Facial Expression of Emotion*, pages 236–249. Guilford Press, New York, USA, 2000. 9

[99] M. Khademi, M. Manzuri-Shalmani, M. Kiapour, and A. Kiaei. Recognizing combinations of facial action units with different intensity using a mixture of hidden markov models and neural network. In *Proc. of the 9th Int'l Conf. on Multiple Classifier Systems*, pages 304–313, 2010. 38

[100] M. Kim. Large margin cost-sensitive learning of conditional random fields. *Pattern Recognition*, 43(10):3683 – 3692, 2010. 171

[101] M. Kim and V. Pavlovic. Hidden conditional ordinal random fields for sequence classification. *Machine Learning and Knowledge Discovery in Databases*, 6322:51–65, 2010. 25, 39, 46, 117, 122, 126, 136, 138, 139

[102] M. Kim and V. Pavlovic. Structured output ordinal regression for dynamic facial emotion intensity prediction. *European Conf. on Computer Vision (ECCV)*, pages 649–662, 2010. 16, 34, 35, 45, 50, 116, 117, 121, 136, 167

[103] G. Kimeldorf and G. Wahba. Some results on Tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, 33:82–95, 1971. 146

[104] S. Kimura and M. Yachida. Facial expression recognition and its degree estimation. In *Int'l Conf. on Pattern Recognition (ICPR)*, pages 295–300, 1997. 15, 46

[105] S. Koelstra, M. Pantic, and I. Patras. A dynamic texture based approach to recognition of facial actions and their temporal models. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 32:1940–1954, 2010. 45

[106] S. Kumano, K. Otsuka, J. Yamato, E. Maeda, and Y. Sato. Pose-invariant facial expression recognition using variable-intensity templates. *Int'l J. of Computer Vision*, 83(2):178–194, 2009. 43, 44

[107] S. Kumar and M. Hebert. Discriminative random fields. *Int'l J. of Comp. Vision*, 68:179–201, 2006. 120, 121

[108] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Int'l Conf. on Machine Learning (ICML)*, pages 282–289, 2001. 37, 50, 116, 117, 119, 120, 121, 159, 169, 174

[109] J. Lafferty, X. Zhu, and Y. Liu. Kernel conditional random fields: representation and clique selection. *Int'l Conf. on Machine Learning (ICML)*, 2004. 145, 146, 190

[110] N.D. Lawrence. Probabilistic non-linear principal component analysis with gaussian process latent variable models. *Journal of Machine Learning Research*, 6:1783–1816, 2005. 99

[111] N.D. Lawrence and J.Q. Candela. Local distance preservation in the gp-lvm through back constraints. In *Int'l Conf. on Machine Learning (ICML)*, volume 148, pages 513–520, 2006. 102

[112] C.S. Lee and A. Elgammal. Facial expression analysis using nonlinear decomposable generative models. In *Automatic Face and Gesture Recognition (FG)*, pages 17–31, 2005. 40, 41

[113] K.K. Lee and Y. Xu. Real-time estimation of facial expression intensity. In *Int'l Conf. on Robotics and Automation (ICRA)*, volume 2, pages 2567–2572 vol.2, 2003. 46

[114] T.S. Lee. Image representation using 2d gabor wavelets. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 18:959–971, 1996. 34

[115] Y. Li, S. Wang, Y. Zhao, and Q. Ji. Simultaneous facial feature tracking and facial expression recognition. *IEEE Trans. on Image Processing*, page In Press, 2013. 40

[116] J. Lien, T. Kanade, J.F. Cohn, and C. Li. Detection, tracking, and classification of action units in facial expression. *J. of Rob. and Aut.Systems*, 1999. 135

[117] J.J. Lien, T. Kanade, J.F. Cohn, and C.C. Li. Automated facial expression recognition based on facs action units. In *Automatic Face and Gesture Recognition (FG)*, pages 390–395, 1998. 38

[118] G.C. Littlewort, M.S. Bartlett, and K. Lee. Automatic coding of facial expressions displayed during posed and genuine pain. *Image and Vision Computing*, 27:1797–1803, 2009. 34

[119] S. Liwicki, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. Efficient online subspace learning with an indefinite kernel for visual tracking and recognition. *IEEE Transactions on Neural Networks and Learning Systems*, 23:1624–1636, 2012. 190

[120] F. Long, T. Wu, J.R. Movellan, M.S. Bartlett, and G. Littlewort. Learning spatio-otemporal features by using independent component analysis with application to facial expression recognition. *Neurocomputing*, 93:126 – 132, 2012. 34

[121] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *Int'l Journal of Computer Vision*, 60(2):91–110, 2004. 35

[122] D.G. Lowe. Object recognition from local scale-invariant features. In *Automatic Face and Gesture Recognition (FG)*, volume 2, pages 1150–1157, 1999. 43

[123] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision (darpa). In *Proc. of the DARPA Image Understanding Workshop*, pages 121–130, 1981. 31

[124] P. Lucey, J.F. Cohn, K.M. Prkachin, P.E. Solomon, S. Chew, and I. Matthews. Painful monitoring: Automatic pain monitoring using the unbc-mcmaster shoulder pain expression archive database. *Image and Vision Computing*, 42:197–205, 2012. 153

[125] P. Lucey, J.F. Cohn, K.M. Prkachin, P.E. Solomon, and I. Matthews. Painful data: The unbc-mcmaster shoulder pain expression archive database. *Automatic Face and Gesture Recognition (FG)*, pages 57–64, 2011. 32, 33, 37, 47, 153, 154, 158, 164, 172, 174

[126] S. Lucey, A.B. Ashraf, and J.F. Cohn. Investigating spontaneous facial action recognition through aam representations of the face. *Face Recognition Book*, 2007. 174

[127] S. Lucey, I. Matthews, C. Hu, Z. Ambadar, and J.F. Cohn. Aam derived face representations for robust facial action recognition. In *Automatic Face and Gesture Recognition (FG)*, pages 155–160, 2006. 36

[128] M.J. Lyons, J. Budynek, and S. Akamatsu. Automatic classification of single facial images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 21(12):1357–1362, 1999. 37

[129] B. Mahasseni and S. Todorovic. Latent multitask learning for view-invariant action recognition. In *Proc. Int'l Conf. Computer Vision (ICCV)*, 2013. 103

[130] M.H. Mahoor, S. Cadavid, D.S. Messinger, and J.F. Cohn. A framework for automated measurement of the intensity of non-posed facial action units. *Int'l Conf. on Computer Vision and Pattern Recognition (CVPR'W)*, pages 74–80, 2009. 15, 47, 159, 165, 174

[131] B. Martinez, M. F. Valstar, X. Binefa, and M. Pantic. Local evidence aggregation for regression based facial point detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 35(5):1149–1163, 2013. 31

[132] I. Matthews and S. Baker. Active appearance models revisited. *Int'l Journal of Computer Vision*, 60(2):135–164, 2004. 30, 31, 32, 33

[133] S. Mavadati, M. Mahoor, K. Bartlett, P. Trinh, and J.F. Cohn. Disfa: A spontaneous facial action intensity database. *IEEE Trans. on Affective Comp.*, 4(2):151–160, 2013. 14, 15, 34, 35, 47, 48, 159, 172, 174, 178

[134] P. McCullagh. Regression models for ordinal data. *Journal of the Royal Stat. Society. Series B*, 42:109–142, 1980. 118, 119

[135] G. McKeown, M.F. Valstar, R. Cowie, and M. Pantic. The semaine corpus of emotionally coloured character interactions. In *Proc. International Conf. on Multimedia and Expo*, pages 1079–1084, 2010. 70, 80

[136] S. Moore and R. Bowden. Local binary patterns for multi-view facial expression recognition. *Computer Vision and Image Understanding*, 115(4):541–558, 2011. 34, 42, 77, 78, 97, 106, 107, 112

[137] K.P. Murphy. The bayes net toolbox for matlab. *Computing Science and Statistics*, 33:2001, 2001. 159

[138] M. A. Nicolaou, H. Gunes, and M. Pantic. Output-associative rvm regression for dimensional and continuous emotion prediction. *Image and Vision Computing*, 30:186–196, 2012. 12

[139] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002. 35, 158

[140] N. Oliver, A. Pentland, and F. Berard. Lafter: a real-time face and lips tracker with facial expression recognition. *Pattern Recognition*, 33(8):1369–1382, 2000. 38

[141] J. Orozco, B. Martinez, and M. Pantic. Empirical analysis of cascade deformable models for multi-view face detection. In *Int'l Conf. on Image Processing (ICIP)*, pages 1–5, 2013. 30

[142] T. Otsuka and J. Ohya. Recognizing multiple persons' facial expressions using hmm based on automatic extraction of significant frames from image sequences. In *Int'l Conf. on Image Processing (ICIP)*, volume 2, pages 546–549 vol.2, 1997. 38

[143] C. Padgett and G.W. Cottrell. Representing face images for emotion classification. In *Neural Inf. Proc. Systems (NIPS)*, pages 894–900. MIT Press, 1996. 37

[144] M. Pantic. Machine analysis of facial behaviour: Naturalistic and dynamic behaviour. *Philosophical Transactions of Royal Society B*, 364:3505–3513, 2009. 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 143

[145] M. Pantic and M.S. Bartlett. *Machine Analysis of Facial Expressions*, pages 377–416. I-Tech Education and Publishing, 2007. 29, 30, 31, 33, 125

[146] M. Pantic, A. Nijholt, A. Pentland, and T.S. Huanag. Human-centred intelligent human? computer interaction (hci^2): how far are we from attaining it? *International Journal of Autonomous and Adaptive Communications Systems*, 1(2):168–187, 2008. 11

[147] M. Pantic and I. Patras. Detecting facial actions and their temporal segments in nearly frontal-view face image sequences. *Proc. of IEEE Int'l Conf. Systems, Man and Cybernetics*, pages 3358–3363, 2005. 45

[148] M. Pantic and I. Patras. Dynamics of facial expression: Recognition of facial actions and their temporal segments from face profile image sequences. *IEEE Trans. on Systems, Man and Cybernetics - Part B*, 36(2):433–449, 2006. 33, 36, 43, 45, 112

[149] M. Pantic, A. Pentland, A. Nijholt, and T.S. Huang. *Human Computing and Machine Understanding of Human Behavior: A Survey*, volume 4451, pages 47–71. 2007. 17, 22, 26, 164, 165, 166

[150] M. Pantic and L.J.M. Rothkrantz. Facial action recognition for facial expression analysis from static face images. *Trans. Sys. Man Cyber. Part B*, 34(3):1449–1461, 2004. 33, 36, 37

[151] M. Pantic, M. F. Valstar, R. Rademaker, and L. Maat. Web-based database for facial expression analysis. *Int'l Conf. on Multimedia and Expo*, pages 317–321, 2005. 149

[152] M. Pantic and G. Caridakis. Image and video processing for affective applications. In *Emotion-Oriented Systems*, Cognitive Technologies, pages 101–114. 2011. 13, 17

[153] M. Pantic and L.J.M. Rothkrantz. Automatic analysis of facial expressions: The state of the art. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22:1424–1445, 2000. 12

[154] I. Patras and M. Pantic. Particle Filtering with Factorized Likelihoods for Tracking Facial Features. *Automatic Face and Gesture Recognition (FG)*, pages 97–102, 2004. 135, 140

[155] K.M. Prkachin and P.E. Solomon. The structure, reliability and validity of pain expression: Evidence from patients with shoulder pain. *Pain*, 139(2):267–274, 2008. 154, 158

[156] A. Quattoni, M. Collins, and T. Darrell. Conditional random fields for object recognition. *Neural Inf. Proc. Systems (NIPS)*, pages 1097–1104, 2004. 117, 122

[157] L.R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. Readings in speech recognition, pages 267–296. 1990. 37, 117

[158] C.E. Rasmussen and Z. Ghahramani. Infinite mixtures of gaussian process experts. In *Neural Inf. Proc. Systems (NIPS)*, pages 881–888, 2001. 111

[159] C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2005. 56, 57, 58, 59, 60, 65, 66, 72, 85, 87, 102

[160] J. Reilly, J. Ghent, and J. McDonald. Investigating the dynamics of facial expression. *Lecture Notes in Computer Science*, 4292:334–343, 2006. 47, 48, 165

[161] O. Rudovic, V. Pavlovic, and M. Pantic. Kernel conditional ordinal random fields for temporal segmentation of facial action units. *European Conf. on Computer Vision (ECCV'W)*, 2012. 158

[162] O. Rudovic, V. Pavlovic, and M. Pantic. Automatic pain intensity estimation with heteroscedastic conditional ordinal random fields. *Int'l Symposium on Visual Computing (ISVC)*, 2013. 47

[163] J.A. Russell. Is there universal recognition of emotion from facial expression? a review of the cross-cultural studies. *Psychological Bulletin*, 115:102–141, 1994. 12

[164] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. A semi-automatic methodology for facial landmark annotation. In *Int'l Conf. on Computer Vision and Pattern Recognition (CVPR'W)*, 2013. 104

[165] M. Salzmann, C.H. Ek, R. Urtasun, and T. Darrell. Factorized Orthogonal Latent Spaces. *Journal of Machine Learning Research*, 9:701–708, 2010. 111, 190

[166] M. Salzmann and R. Urtasun. Implicitly constrained gaussian process regression for monocular non-rigid pose estimation. *Neural Inf. Proc. Systems (NIPS)*, pages 2065–2073, 2010. 88

[167] G. Sandbach, S. Zafeiriou, M. Pantic, and L. Yin. Static and dynamic 3d facial expression recognition: A comprehensive survey. *Image and Vision Computing*, 30(10):683–697, 2012. 3D Facial Behaviour Analysis and Understanding. 44

[168] A. Savrana, B. Sankur, and M.T. Bilgeb. Regression-based intensity estimation of facial action units. *Image and Vision Computing*, 2012. 47, 48, 165

[169] K. Scherer and P. Ekman. *Handbook of methods in nonverbal behavior research*. Cambridge University Press, 1982. 14

[170] B. Schölkopf, R. Herbrich, and A.J. Smola. A generalized representer theorem. In *Proc. of the 14th Annual Conf. on Comp. Learning Theory*, COLT '01, pages 416–426. Springer-Verlag, 2001. 26, 50, 146

[171] B. Schölkopf, A.J. Smola, R.C. Williamson, and P.L. Bartlett. New support vector algorithms. *Neural Comput.*, 12:1207–1245, 2000. 59, 70

[172] F. Sha and L.K. Saul. Large margin hidden markov models for automatic speech recognition. *Neural Inf. Proc. Systems (NIPS)*, pages 1249–1256, 2007. 170, 171

[173] C. Shan. *Inferring facial and body language*. PhD thesis, Queen Mary, University of London, 2007. 15, 34, 37, 46

[174] C. Shan and R. Braspenning. Recognizing facial expressions automatically from video. In *Handbook of Ambient Intelligence and Smart Environments*, pages 479–509. 2010. 12

[175] C. Shan, S. Gong, and P.W. Mcowan. Appearance manifold of facial expression. *Lecture Notes in Comp. Science*, 3766:221–230, 2005. 46, 129

[176] C. Shan, S. Gong, and P.W. Mcowan. Dynamic facial expression recognition using a bayesian temporal manifold model. In *British Machine Vision Conf. (BMVC)*, pages 297–306, 2006. 40, 41, 49

[177] C. Shan, S. Gong, and P.W. McOwan. Facial expression recognition based on local binary patterns: A comprehensive study. *Image and Vision Comp.*, 27(6):803–816, 2009. 35, 37

[178] L. Shang and K.P. Chan. Nonparametric discriminant hmm and application to facial expression recognition. In *Int'l Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2090–2096, 2009. 38, 45

[179] A. Sharma, A. Kumar, H. Daume, and D.W. Jacobs. Generalized multiview analysis: A discriminative latent space. In *Int'l Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2160–2167, 2012. 103, 104

[180] J.Q. Shi, R. Murray-Smith, M. Titterington, and J.Q. Shi. Bayesian regression and classification using mixtures of gaussian processes. In *Int'l Journal of Adaptive Control and Signal Processing*, pages 149–161, 2003. 111

[181] P.E. Shorut and J.L. Fleiss. Intraclass correlations: Uses in assessing rater reliability. *Psychological Bulletin*, 86(2):420–428, 1979. 159, 175

[182] T. Simon, M.H. Nguyen, F. De la Torre, and J.F. Cohn. Action unit detection with segment-based svms. In *Int'l Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2010. 39

[183] A. Skrondal and S. Rabe-Hesketh. *Generalized Latent Variable Modeling Multilevel, Longitudinal, and Structural Equation Models*. Chapman & Hall, 2004. 168

[184] Y. Sun and L. Yin. Facial expression recognition based on 3d dynamic range model sequences. In *European Conf. on Computer Vision (ECCV)*, pages 58–71, 2008. 44

[185] C.A. Sutton, A. Mccallum, and K. Rohanimanesh. Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. *Journal of Machine Learning Research*, 8:693–723, 2007. 191

[186] H. Tang, M. Hasegawa-Johnson, and T.S. Huang. Non-frontal view facial expression recognition based on ergodic hidden markov model supervectors. In *Proc. Int'l Conf. on Multimedia and Expo*, pages 1202–1207, 2010. 77, 78

[187] U. Tariq, J. Yang, and T.S. Huang. Multi-view facial expression recognition analysis with generic sparse coding feature. In *European Conf. on Computer Vision (ECCV'W)*, pages 578–588, 2012. 34, 42, 43, 106, 107

[188] Y. Tian. Evaluation of face resolution for expression analysis. In *Int'l Conf. on Computer Vision and Pattern Recognition (CVPR'W)*, pages 82–82, 2004. 37

[189] Y. Tian, T. Kanade, and J.F. Cohn. Facial expression analysis. In *Handbook of Face Recognition*, pages 247–275. 2005. 10, 11, 13

[190] M.E. Tipping. Sparse bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244, 2001. 47, 59, 60

[191] Y. Tong, J. Chen, and Q. Ji. A unified probabilistic framework for spontaneous facial action modeling and understanding. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 32(2):258–273, 2010. 14, 16, 32

[192] Y. Tong, W. Liao, and Q. Ji. Facial action unit recognition by exploiting their dynamic and semantic relationships. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(10):1683–1699, 2007. 14, 40

[193] C. Trad, H. Hajj, W. El-Hajj, and F. Al-Jamil. Facial action unit and emotion recognition with head pose variations. In *Advanced Data Mining and Applications*, volume 7713, pages 383–394. 2012. 43

[194] V. Tresp. A bayesian committee machine. *Neural Computing*, 12(11):2719–2741, 2000. 68

[195] V. Tresp and M. Taniguchi. Combining estimators using non-constant weighting functions. In *Neural Inf. Proc. Systems (NIPS)*, pages 419–426, 1995. 67

[196] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453–1484, 2005. 165, 170, 171

[197] G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. Subspace learning from image gradient orientations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(12):2454–2466, 2012. 190

[198] R. Urtasun and T. Darrell. Discriminative gaussian process latent variable model for classification. In *Int'l Conf. on Machine Learning (ICML)*, pages 927–934, 2007. 25, 98, 100, 101, 104, 107

[199] M.F. Valstar, H. Gunes, and M. Pantic. How to distinguish posed from spontaneous smiles using geometric features. *Int'l Conf. on Multimodal Interfaces (ICMI)*, pages 38–45, 2007. 14, 191

[200] M.F. Valstar and M. Pantic. Fully automatic recognition of the temporal phases of facial actions. *IEEE Trans. on Systems, Man, and Cybernetics*, 42(1):28–43, 2012. 14, 16, 32, 33, 36, 38, 45, 70, 123, 149, 150, 151, 152, 174

[201] M.F. Valstar, B. Martinez, X. Binefa, and M. Pantic. Facial point detection using boosted regression and graph models. In *Proc. Int'l Conf. Computer Vision and Pattern Recognition*, pages 2729–2736, 2010. 31

[202] L. van der Maaten and E. Hendriks. Action unit classification using active appearance models and conditional random fields. *Cognitive Processing*, 13(2):507–518, 2012. 38

[203] A. Varol, M. Salzmann, E. Tola, and P. Fua. Template-free monocular reconstruction of deformable surfaces. In *Proc. Int'l Conf. Computer Vision (ICCV)*, pages 1811–1818, 2009. 111

[204] P. Viola and M. Jones. Robust real-time object detection. In *Int'l Journal of Computer Vision*, 2004. 30

[205] S Vishwanathan, N. Schraudolph, M. Schmidt, and K. Murphy. Accelerated training of conditional random fields with stochastic meta-descent. *Int'l Conf. on Machine Learning (ICML)*, 2006. 121

[206] D. Vukadinovic and M. Pantic. Fully automatic facial feature point detection using gabor feature based boosted classifiers. In *IEEE Int'l Conf. Systems, Man and Cybernetics (SMC'05)*, pages 1692–1698, 2005. 30, 31, 32, 33

[207] J.M. Wang, D.J. Fleet, and A. Hertzmann. Gaussian process dynamical models for human motion. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 30(2):283–298, 2008. 112

[208] J. Wang, L. Yin, X. Wei, and Yi Sun. 3d facial expression recognition based on primitive surface feature distribution. In *Int'l Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1399–1406, 2006. 42, 44, 70, 89

[209] S. Wang, A. Quattoni, L.P. Morency, D. Demirdjian, and T. Darrell. Hidden conditional random fields for gesture recognition. *Int'l Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1097–1104, 2006. 39, 122

[210] T. Wang and J. James Lien. Facial expression recognition system based on rigid and non-rigid motion separation and 3d pose estimation. *Pattern Recognition*, 42(5):962 – 977, 2009. 43, 44

[211] J. Whitehill and C.W. Omlin. Haar features for FACS AU recognition. In *Automatic Face and Gesture Recognition (FG)*, 2006. 34

[212] A.C. Williams. Facial expression of pain: An evolutionary account. *Behavioral and Brain Sciences*, 25(4):439–488, 2002. 16

[213] A.G. Wilson, D.A. Knowles, and Z. Ghahramani. Gaussian process regression networks. In *Int'l Conf. on Machine Learning (ICML)*, 2012. 95

[214] R. Winkelmann and S. Boes. *Analysis of microdata*. Springer, 2006. 117, 118, 120, 156, 185, 190

[215] Y. Xiang, X. Zhou, Z. Liu, T.S. Chua, and C. Ngo. Semantic context modeling with maximal margin conditional random fields for automatic image annotation. *Int'l Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3368–3375, 2010. 185

[216] X. Xiong and F. De la Torre. Supervised descent method and its applications to face alignment. In *Int'l Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2013. 31

[217] P. Yang, Q. Liu, and D.N. Metaxas. Boosting encoded dynamic features for facial expression recognition. *Pattern Recognition Letters*, (2):132 – 139, 2009. 37

[218] P. Yang, Q. Liu, and D.N. Metaxas. Rankboost with l1 regularization for facial expression recognition and intensity estimation. *Proc. Int'l Conf. Computer Vision (ICCV)*, pages 1018–1025, 2009. 34, 35, 43, 46, 47

[219] M. Yeasin, B. Bullot, and R. Sharma. Recognition of facial expressions and measurement of levels of interest from video. *IEEE Trans. on Multimedia*, 8(3):500–508, 2006. 38

[220] L. Yin, X.n Chen, Y. Sun, T. Worm, and M. Reale. A high-resolution 3d dynamic facial expression database. *Automatic Face and Gesture Recognition (FG)*, pages 679–684, 2008. 135

[221] Z. Zeng, M. Pantic, G.I. Roisman, and T.S. Huang. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 31(1):39–58, 2009. 19, 55, 70

[222] C. Zhang and Z. Zhang. A survey of recent advances in face detection. Technical Report MSR-TR-2010-66, Microsoft Research, 2010. 30

[223] Y. Zhang and Q. Ji. Active and dynamic information fusion for facial expression understanding from image sequences. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(5):699–714, 2005. 40

[224] Z. Zhang, M. Lyons, M. Schuster, and S. Akamatsu. Comparison between geometry-based and gabor-wavelets-based facial expression recognition using multi-layer perceptron. In *Automatic Face and Gesture Recognition (FG)*, pages 454–459, 1998. 33, 34

[225] W. Zheng, H. Tang, Z. Lin, and T.S. Huang. Emotion recognition from arbitrary view facial images. *European Conf. on Computer Vision (ECCV)*, pages 490–503, 2010. 34, 42, 43, 78

[226] W. Zheng, H. Tang, Z. Lin, and T.S. Huang. A novel approach to expression recognition from non-frontal face images. In *Proc. Int'l Conf. Computer Vision (ICCV)*, pages 1901–1908, 2009. 77, 78

[227] G. Zhong, W. Li, D. Yeung, X. Hou, and C. Liu. Gaussian process latent random field. *Int'l Conf. on Artificial Intelligence (AAAI)*, 2010. 25, 100, 101, 104, 107

[228] X. Zhu and D. Ramanan. Face detection pose estimation, and landmark localization in the wild. In *Int'l Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2879 –2886, 2012. 30, 31

[229] X. Zhu, J. Lafferty, and Z. Ghahramani. Semi-supervised learning: From gaussian fields to gaussian processes. Technical report, School of Comp. Science, Carnegie Mellon University, 2003. 101, 129

[230] Z. Zhu and Q. Ji. Robust real-time face pose and facial expression recovery. In *Int'l Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 681–688, 2006. 44, 73, 90