Imperial College London
Department of Computing

Gaussian Processes for Modeling of Facial Expressions

Stefanos Eleftheriadis

September, 2016

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.

Acknowledgements

I would like to thank my supervisor Prof. Maja Pantic for trusting me with a challenging PhD topic and for her support over the past years. I would also like to thank Dr. Ognjen Rudovic for his valuable help and guidance, especially during the first years of my PhD. He was the one who had to put up with my eager enthusiasm and was assigned with the difficult task of initiating me into the cruel world of academic research. I feel extremely lucky for meeting with Dr. Marc Deisenroth, and I am grateful to him for our collaboration and our fruitful discussions during the final stage of my research. I couldn't, of course, forget to mention Amani and Teresa, for their invaluable help and their relentless willingness to manage all the less sexy administrative stuff for me. Thank you for making my job easier.

A big thanks goes to all my friends from the i-BUG group for the great moments that we have shared, and for being such a great company throughout the years. However, I feel obliged to explicitly express my deepest gratitude to my dear friends Christos*, Nondas*, Thanos* and Simos* (you all know what the star means!), for putting up with me all these years. If it wasn't for you guys, my grumpiness wouldn't have been developed to its absolute maximum! Special thanks to Simos (again), Stavros and Yannis for proofreading parts of this thesis. I would also like to thank Sanjay, for happily serving as my on-the-spot English dictionary, and for our joyful long-lasting chats during our parallel writing-up period. A special mention to my old lads from Greece, Dimitris, Stamatis and Vagelis, for being around to support and advice me. Don't worry Kosta, I haven't forgotten your high-pitching 'ela rei'!

I couldn't forget to thank the constant variables in my life. As in every optimization problem, they are always there to remind you the original cost of your decisions and choices. I'm referring to my life-time brotherhood friends, Evripidis, Giannis and Thanos. I still haven't learned to live without you being part of my daily routine.

I owe a special mention to my family in Greece, Rania, Makis, yiayia Europe and Sakis. You have always been there for me and I could always feel the warmth of your love even through our shortest phone calls.

I would like to express my sincere gratitude to my sister and roommate Ioanna. It is for her that my moving to a new country was so smooth. It is for her that a strange flat feels like home. Most importantly, it was her that had to deal with my mood on this long and exhausting journey. I just thank you! George, you joined a bit late, as usual, but it is a common knowledge that your decision to put up with both the siblings is courageous!

Finally, no words can express my gratitude to the latent variables that I will always condition on, the ones that govern my random process, my parents, Stefanos and Zoi. Thank you for letting me take all the wrong decisions in this journey, and thank you for making my problems yours. Without your love and unconditional support it wouldn't be possible for me to achieve anything.

"The people you love the most are the ones you are more comfortable to share the silence with..."

This thesis is dedicated to my family, Stefanos, Zoi and Ioanna.

No Journey is complete without reaching the destination!
To Ithaka and what it represents...
Because it needs a god's plea for Calypso letting you free...
and in your fights with Cyclopes and Laestrygonians you will be remembered as 'Nobody'.
Because Scylla and Charybdis won't let you pass through...
and the Sirens' enchanting song will get you lured.
Because Penelope's faith won't last for long...
and the suitors are too many to slaughter them all.

Abstract

Automated analysis of facial expressions has been gaining significant attention over the past years. This stems from the fact that it constitutes the primal step toward developing some of the next-generation computer technologies that can make an impact in many domains, ranging from medical imaging and health assessment to marketing and education. No matter the target application, the need to deploy systems under demanding, real-world conditions that can generalize well across the population is urgent. Hence, careful consideration of numerous factors has to be taken prior to designing such a system. The work presented in this thesis focuses on tackling two important problems in automated analysis of facial expressions: (i) view-invariant facial expression analysis; (ii) modeling of the structural patterns in the face, in terms of well coordinated facial muscle movements. Driven by the necessity for efficient and accurate inference mechanisms we explore machine learning techniques based on the probabilistic framework of Gaussian processes (GPs). Our ultimate goal is to design powerful models that can efficiently handle imagery with *spontaneously* displayed facial expressions, and explain in detail the complex configurations behind the human face in real-world situations.

To effectively decouple the head pose and expression in the presence of large out-of-plane head rotations we introduce a manifold learning approach based on multi-view learning strategies. Contrary to the majority of existing methods that typically treat the numerous poses as individual problems, in this model we first learn a discriminative manifold shared by multiple views of a facial expression. Subsequently, we perform facial expression classification in the expression manifold. Hence, the pose normalization problem is solved by aligning the facial expressions from different poses in a common latent space. We demonstrate that the recovered manifold can efficiently generalize to various poses and expressions even from a small amount of training data, while also being largely robust to corrupted image features due to illumination variations. State-of-the-art performance is achieved in the task of facial expression classification of basic emotions.

The methods that we propose for learning the structure in the configuration of the muscle movements represent some of the first attempts in the field of analysis and intensity estimation of facial expressions. In these models, we extend our multi-view approach to exploit relationships not only in the input features but also in the multi-output labels. The structure of the outputs is imposed into the recovered manifold either from heuristically defined hard constraints, or in an auto-encoded manner, where the structure is learned automatically from the input data. The resulting models are proven to be robust to data with imbalanced expression categories, due to our proposed Bayesian learning of the target manifold. We also propose a novel regression approach based on product of GP experts where we take into account people's individual expressiveness in order to adapt the learned models on each subject. We demonstrate the superior performance of our proposed models on the task of facial expression recognition and intensity estimation.

Contents

Chapter 1

Introduction

"*Every face could become spiritually beautiful through the accurate rendering of his or her emotions*"

Duchenne de Boulogne

Contents

Facial expressions convey emotions, provide clues about people's personality and intentions, reveal the state of pain, weakness or hesitation, among others. The study and understanding of human facial expressions has been a long standing problem. The first reported scientific research on the analysis of facial expressions can be tracked back to as early as 1862, when Duchenne de Boulogne published the 'Mécanisme de la physionomie humaine' [49]. In his study, influenced by the beliefs of physiognomy of the 19th century, Duchenne wanted to determine how the muscles in the human face produce facial expressions. He believed that the reading of the expressions alone could reveal an accurate rendering of the soul's emotions. Directly related to this belief is also the seminal work of Charles Darwin, who studied facial expressions and body gestures in mammals [40]. Darwin explored the importance of facial expressions for communication and described variations in facial expressions of emotions. The

goal of his work was to show how human expressions link human movements with emotional states, and are genetically determined from purposeful animal actions. He was one of the first who studied complex emotional states including self-attention, shame, shyness, modesty and blushing, setting the foundations of the study of affect.

An influential milestone in the facial expression analysis, is the work of Paul Ekman [53]. According to Ekman, there exists a set of six basic emotions (anger, fear, disgust, happiness, sadness and surprise) that can be globally encountered across populations of different cultures. The latter suggests that these six basic emotions are not only universal in terms of expressing, but also in terms of understanding them. This findings encouraged Ekman & colleagues to deepen their studies in various works [55, 51, 56, 54, 52], which can be regarded as the beginning of what we now call affect analysis. These works set the basis for describing and analyzing facial expressions not only of emotions, but also of cognitive states, such as interest, boredom, confusion, stress, etc. Emerging from these studies, there has been noted an ever growing research attention towards the analysis of human affect in the past years, spanning the fields of psychology, cognitive science and computer science. This increasing interest in recognizing and interpreting the human emotion resulted in the birth of *affective computing* [145], which focuses on the development of autonomous systems and devices, capable of simulating and analyzing the human affect. The applicability of affective computing expands in various domains, from medicine and psychology to security, covering numerous applications, such as human-computer interaction, analysis of social behavior, pain monitoring and entertainment, among others [137, 190, 139, 13, 67].

In this thesis, inspired by the works in the field of affective computing, we explore and propose various techniques based on machine learning and pattern recognition for analyzing the human affect, and in particular, facial expressions. The remainder of the introductory chapter is organized as follows. Firstly, in Section 1.1 we refer in more detail to the problem space on which the thesis builds on, and in Section 1.2 we introduce the most commonly encountered modeling challenges. We then describe in more detail our main contributions in Section 1.3, and list the publications that stemmed from this work in Section 1.4. Finally, in Section 1.5 we give the outline of the thesis.

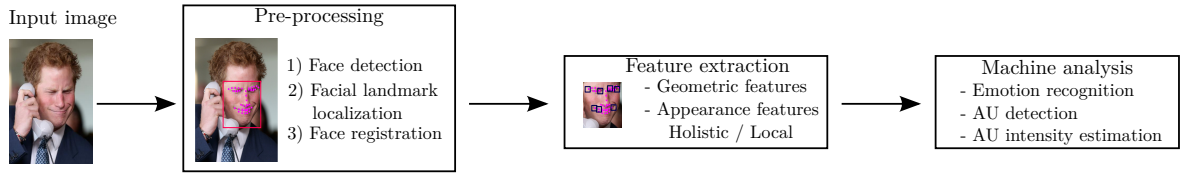


Figure 1.1: A typical system for automated analysis of facial expressions. Given an input image, the first step consists of pre-processing of the target image(s). Subsequently, we proceed to the feature extraction. Different geometric and/or appearance features can be used, which are usually chosen depending on the target task. The final step is machine interpretation of facial expressions.

1.1 Problem Space

1.1.1 Automated Analysis of Facial Expressions

The ultimate goal of affective computing is to build automated systems for analyzing and simulating the human affect. This is usually attempted by trying to train the computer to interpret facial motions and cues from visual information (*i.e.*, video streams or images). Although this seems a relatively easy task given the human's ability to analyze facial expressions with little effort, development of such a system is quite challenging and requires careful design [139]. A typical system aimed toward automated analysis of facial expressions follows the pipeline depicted in Fig. 1.1. This architecture is comprised of three basic steps. First we need to determine whether a face exists and in which location in an input image. This is the pre-processing step. After having correctly located the face, we proceed to the feature extraction step. Finally, the extracted features serve as input to a machine learning algorithm for analyzing the facial expression. Note that recently, a new kind of learning paradigm based on deep learning, comes to challenge the above framework, *e.g.*, deep convolutional neural networks [100]. In such systems, the tree individual steps are normally combined into a single learning procedure, in which, image registration, feature extraction and classification can be performed jointly. Nevertheless, such an approach is out of the scope of this thesis, and hence, in the following we give a brief description of each step from the pipeline depicted in Fig. 1.1.

Pre-processing

In order to extract features from facial images, the first step consists of three parts: (i) detecting the face in a given image; (ii) determining the actual location of the face; (iii) registering all faces in a common coordinate system. In what follows we briefly describe each part.

Face Detection. As we have already mentioned above, the primal step toward achieving auto-

mated facial expression analysis is the detection of the face in a given input image. This is proven to be rather challenging, especially when dealing with imagery from real-world conditions where we encounter numerous faces depicted in varying illumination conditions, variations in the head pose, occlusions of key parts of the face, etc. The most widely used algorithm for face detection is the Viola-Jones [191], which exhibits reliable performance on close to frontal images. Extensions of [191] to multi-view facial detectors are reported in [213, 134].

Facial Landmark Localization. After having located the existence of a face in an input image, a set of points has to be localized on the face. These facial landmarks are defined as distinctive face locations, such as the corners of the eyes and mouth, contours of the eyebrows or tip of the nose. The landmark points, when combined together in sufficient numbers define the face shape. The process of landmark localization is quite complex and remains an active research topic. It usually requires performing statistical analysis on well defined shape and texture models in order to explain variations in both facial shapes and appearances. Based on these models, novel shape instances can be generated and fitted in new face images. Well studied methods for this purpose are the active appearance model (AAM) [32] and the constrained local model (CLM) [10]. Note that the landmark localization step may be omitted in the case only the texture of the face is required for the task at hand. However, it is most often required since in most applications faces need to be spatially aligned and registered.

Face Registration. Prior to the feature extraction step we need to eliminate unwanted variations between the faces, such as differences scale / pose and location. This is achieved via registering all facial images in a common coordinate frame. First, registration of the facial points is performed, usually by applying the Procrustes analysis [73] to the set of face shapes, in order to find a global affine transform. 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. The registration of the texture follows. This can be performed by applying the learned global affine transform to the whole facial texture. An alternative is to learn a piece-wise affine transform for the different facial parts and then warp the facial texture to the reference frame. While the former may better preserve facial expression details, the latter is better for reducing the subject differences.

Feature Extraction

Having processed the facial images, the next step consists of extracting the desired features. The most common employed features can be categorized into geometric and appearance based [201, 42].

Geometric features. As the name implies, geometric features are usually a collection of information regarding the morphology of the face. The most widely used geometric representation is the 2D Cartesian coordinates of certain points in the face, *i.e.*, the aforementioned facial landmarks. They are readily interpretable, and thus, they are especially attractive for behavioral scientists, who can use them to derive rules for studying the meaning of expressions. The set of facial landmarks can be enhanced by including angle- and distance-based representations, in order to encode the configuration and geometric deformations of the human face.

Appearance features. Contrary to the geometric features, appearance-based features encode the textural information of the face. Therefore, they can effectively capture changes in the face caused by wrinkles, bulges, and furrows [87]. The original pixels of the facial image can be used as an appearance descriptor. However, more advanced features have been proposed throughout the years, which are more suitable for the of facial expression analysis. A set of commonly used appearance features include the gradient-based descriptors, such as histograms of oriented gradients (HOGs) [35] and scale invariant feature transform (SIFT) [113]. Another widely used descriptor is the local binary patterns (LBPs) [131], which quantifies the relative information between neighboring pixels. Feature sets borrowed from the signal processing community have also been applied on expression analysis, such as the Gabor wavelets [108] and the discrete cosine transform (DCT) [4]. In general, some of these features are better suited to represent global appearance (*e.g.*, Gabors and DCT), and hence, are extracted holistically from the entire image. On the other hand, local descriptors, such as gradient-based features and LBPs, are usually extracted from patches centered around the facial landmarks. Despite the plethora of available appearance-based features, none of them can be regarded as a universal descriptor that performs well in a variety of applications. Thus, the choice is usually made by weighting the trade-off between accuracy, complexity and robustness to various transformations and noise.

Machine Analysis of Facial Expression

After the extraction of the desired facial features, the final step involves the design and application of machine learning algorithms in order to facilitate the analysis of facial expressions. Different models and learning strategies have been proposed throughout the years, varying from simple classifiers applied on the extracted features, to learning low-level dependencies among the features based on some form of statistical analysis. A detailed overview of the methods proposed for facial expression analysis is given in Chapter 2.

The work presented in this thesis falls in the final step of the system in Fig. 1.1. In particular,

we propose different machine learning algorithms with the aim of addressing some of the most commonly encountered problems in automated analysis of facial expressions. In the remaining of the section, we elaborate on the particular characteristics of the facial expressions, which should be considered prior to designing a machine learning algorithm. We then continue to the following sections, by introducing the most commonly encountered modeling challenges, and listing of our contributions.

1.1.2 Facial Expression Analysis: Emotions vs. Facial Action Units

In the pipeline described above, facial expressions can be described at different levels [177]. The more prevalent approaches focus on identifying either the exact facial affect, or the activations of facial muscles, named action units (AUs). According to [31] these orthogonal approaches – referred to as message and sign judgment, respectively – are just different measurements for facial expressions.

Automated analysis of facial expressions based on the message judgment tries to decode the conveyed meaning, normally in terms of the six basic emotions, as described by Ekman [53]. The simplicity of this approach has attracted the interest of the majority of the works proposed in the field [142]. However, in practice, categorizing all facial expressions as basic emotions is of limited applicability. Displaying of a certain facial expression does not necessarily mean that the person is actually experiencing the associated emotion. An illustrative example is the smiley expressions which can appear in moments of both happiness and embarrassment [6]. Apart from the ambiguity between the facial expressions and the underlying emotion there is another discouraging factor for the use of the message judgment measurement. According to the recent study of [47] people usually display compound emotions. Compound emotions are those that can be constructed by combining basic component categories to create new ones. For instance, a happily surprised expression, which we frequently use when we randomly bump into someone loved on the street, is a combination of happiness and surprise. There are many more compound categories that involve combination of different emotions. This suggests that the message judgment approach, which assumes the existence of a set of mutually exclusive classes of emotions, is not well suited for the particular task. Perhaps a more viable option would be to examine the facial muscle movements that are observed in the underlying facial expression.

Toward this direction, the approach that employs the sign judgment relies on identifying the correct facial muscle configuration that is responsible for producing the displayed expression. To describe the possible configurations, the facial action coding system (FACS) [54] defines 32

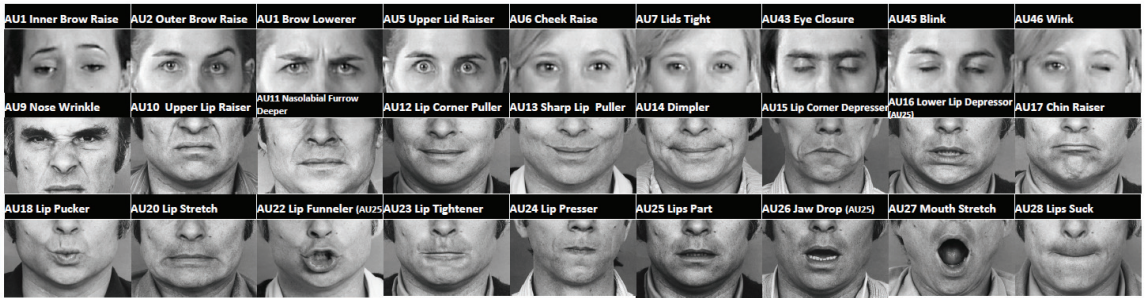


Figure 1.2: Facial action units (AUs), with 9 AUs for the upper face and 18 for the lower, containing images from [54]. Figure adapted from [150].

unique, non-overlapping, visually detectable facial muscle activations, *i.e.*, the AUs. 9 Out of the 32 AUs are defined for describing the upper face, 18 for the lower face, while the rest cannot be exclusively attributed to either. A list of facial AUs can be found in Fig. 1.2. Furthermore, FACS encodes several categories of head/eye positions and other movements, which can be used to describe miscellaneous actions. FACS also defines rules for scoring the intensity of each AU in the range from absent to maximal intensity on a six-point ordinal scale. This, in turn, is critical for high-level interpretation of facial expressions.

Up until recently, the dominant approach toward the facial expression analysis was the message judgment. However, since every possible facial expression can be described as a combination of different AUs, the research trend has been shifted toward automated analysis of AUs, *i.e.*, the sign judgment approach. For instance, the FACS has been used to teach children on the autism spectrum to produce facial expressions [72], to demonstrate differences between polite and amused smiles [6], as well as voluntary and evoked expressions of pain [56]. In this thesis we first start by employing the message judgment approach, and then continue by proposing techniques for the analysis of AUs.

1.1.3 Posed vs. Spontaneous Expressions

An important factor that should also be considered during the design of an automated system for facial expression analysis is the difference in the elicitation and the origin of the expressions. Based on these criteria, facial expressions can be described as either *posed* or *spontaneous* expressions. Posed expressions are usually collected from trained actors or random subjects that were requested to exhibit a particular facial expression, *e.g.*, disgust, while being recorded in a well constrained laboratory environment. On the other hand, spontaneous expressions are captured on real-world conditions, and appear involuntary on the subject's face to communicate the elicited emotion.

The core differences between posed and spontaneous expressions have been extensively studied in psychology and cognitive science [52, 56], where it has been found that they are controlled from different areas in the brain. In particular, deliberate facial activities originate mostly in the motor strip of the neocortex, whereas the less voluntary facial movements are initiated in the sub-cortical part of the brain. These neuroanatomical indications suggest that different activation patterns of the facial muscles are involved in the formation of posed and spontaneous expressions. A well studied paradigm is the different AUs that are present in facial expressions of spontaneous and fake smiles. In genuine smiles (or 'Duchenne smiles') the associated facial expression is composed by the combination of AU12 ('lip corned puller') and AU6 ('cheek raiser'). On the other hand, expressions of deliberate smiles can be usually described only from the presence of AU12. Apart from the AU co-occurrence patterns, the nature of the facial expressions significantly affects the intensity and the duration of them. In general, spontaneously displayed facial expressions are characterized by synchronized and smooth muscle movements, contrary to the less smooth posed expressions [139]. Moreover naturalistic expressions are usually more subtle and involve large out-of-plane head movements [116]. Hence, it is not surprising that the performance of automated systems that are developed based on posed facial displays is expected to downgrade when applied to spontaneous expressions, a fact which constitutes them inapplicable to real-world situations.

Lately, due to the availability of appropriate datasets, the research community have shifted their attention toward designing systems for automated analysis of spontaneous facial expressions [15, 122]. In particular, a lot of studies focus on discriminating spontaneous from posed facial behavior, such as in expressions of smile [187] or pain [111]. Despite the significant progress that has been made, there is still space for improvement. This can be achieved by developing new methodologies and learning strategies that specifically tackle the challenges that arise when dealing with spontaneous facial expressions (*e.g.*, variations in head pose, illumination conditions, co-occurrence patterns etc.). The models that we propose in this thesis try to address some of those challenges.

1.2 Challenges

The machine analysis of facial expressions is challenging mainly due to the complexity and subtlety of human facial behavior, as well as individual differences in expressiveness and variations in head-pose, illumination, occlusions, and so on [139]. In this section, we introduce a set of rising challenges in the field, in order to facilitate later discussions on the practical contributions of our work.

1.2.1 Multiple Views

The focus of the research during the past years was on imagery in which the depicted persons are relatively still and exhibit posed expressions in a nearly frontal pose [201]. However, many real-world applications relate to spontaneous interactions (*e.g.*, meeting summarization, political debates analysis, etc.), in which people tend to move their head while being recorded. Furthermore, depending on the camera position, facial images can be captured from multiple views. These variations in head pose and/or view angle have an adversary impact on the analysis of facial expressions. First of all the head pose is responsible for violating the symmetry of the face, and under extreme rotations it can lead to self-occlusions, *i.e.*, certain parts of the face are not visible. Thus, the characteristics of the face, on which we rely in order to perform facial expression analysis, are being distorted in the presence of arbitrary views. Hopefully, this can be rectified due to the well-known symmetry of the face. However, the main challenge is to decouple the rigid facial changes – due to the head pose – and the non-rigid facial changes – due to the expression – as they are non-linearly coupled in 2D images [216]. Another factor that needs to be addressed when dealing with multi-view, and especially corresponding data (*e.g.*, security cameras in a monitored environment), is the redundancy of the information, and the variations in the illumination conditions. For instance, according to recent studies in the field [138], it is shown that the left hemisphere of the face is more informative when it comes to expressing negative emotions (*e.g.*, Disgust), while the right hemisphere is more informative for positive emotions (*e.g.*, Happiness). However, such assumptions are not expected to strictly hold when the two hemispheres are exposed to different illumination. The lighting conditions may significantly affect the appearance of the face, *e.g.*, light shadows may be confused with wrinkles, which imply the presence of facial deformations associated with certain expressions. This can lead, not only automated systems but humans also, to falsified deductions regarding the displayed expression. The above challenges exemplify the need for effectively exploiting the information from multiple views in order to facilitate the expression analysis. Thus, accounting for the fact that each view is just a different manifestation of the same underlying facial expression related content, multi-view analysis is expected to result in more effective models for the task at hand.

1.2.2 Multiple Modalities

As we have seen the analysis of facial expression can be carried out by using either geometric or appearance-based features. The geometric features capture changes in the location of specific salient facial points caused by facial muscles activity (*e.g.*, facial points displacement between

expressive and expressionless faces [114]). On the other hand, the appearance-based features capture transient differences in the facial appearance such as wrinkles, bulges and furrows. While the former are more robust to illumination and pose changes, not all AUs can be detected solely from the geometric features [189]. For example, the activation of AU6 wrinkles the skin around the outer corners of the eyes and raises the cheeks, which makes it difficult, if not impossible, to detect this AU from facial landmarks only. On the other hand, raising of the eye brows, *i.e.*, AU1,2, can effectively be explained from the geometric deformations of the face shape. Apart from the different characteristics of the employed features there are also other factors that have to be taken into consideration. Geometric features are highly dependent on the underlying tracking algorithm, and hence, different algorithms can produce inconsistent face shapes. Appearance-based features are typically high-dimensional and contain subject-specific information, both of which can adversely affect the performance of the expression analysis. A fused model should be able to effectively handle two different situations. First, to isolate corrupted data, commonly arising in spontaneous, real-world scenarios. This should apply even in cases where the corruptions are not spread evenly across the modalities, *e.g.*, noisy appearance features due to illuminations or occlusions, compared to the unaffected geometric features. Second, the relevance of each modality should be automatically determined by the model, regarding the target task.

1.2.3 Structural Patterns in Expressions

Interpreting the facial expression in terms of basic emotions is straightforward since a single label can be assigned to an input facial image. On the other hand, AUs rarely appear in isolation, and thus, determining the AU configuration in a facial expression is a far more difficult task due to the large number of possible combinations (more than 7,000, especially in spontaneous data) [159]. This constitutes the AU analysis a multi-label problem, in the sense that multiple AUs can be active, and in different intensities, within a single image. For this reason, contrary to the prevalent approach of treating each AU independently [189, 139], the AU analysis can be improved at the model level by exploiting the 'semantics' of AUs, in terms of their co-occurrences. An illustrating example is the case where the activations of certain AUs are driven based on latent factors, such as emotions. For instance, the co-occurrence of AU12 and AU6 signals the facial expression of smiles that are related to joy. On the other hand, expressions where AU12 occurs alone are associated with fake smiles, as in situations of sarcasm. Also, the co-occurring AUs can be non-additive, in the case of which one AU masks another, or a new and distinct set of appearances is created [54]. For example, 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. This, in turn, significantly affects the appearance features of the target AUs.

Modeling of the co-occurrences is also beneficial when scoring the intensity of the AUs, apart from their presence/absence. For instance, the criteria for intensity scoring of AU7 (lid tightener) are changed significantly if AU7 appears with a maximal intensity of AU43 (eye closure), since this combination changes the appearance as well as timing of these AUs [151]. These co-occurrences are usually driven by the context in which the target facial behavior occurs (*e.g.*, pain or joy). Encoding this type of information during the joint AU analysis helps to reduce the space of possible AU combinations in target data, resulting in simpler and more effective models for the joint prediction. The importance of this can be understood better by comparing the joint analysis with individual AU models. In the latter case, to effectively address the problem, one needs to train separate models not only for each AU, but also for each non-additive combination at different intensity levels. However, joint AU analysis is not always expected to be superior to the individual modeling. For instance, different co-occurrence patterns can be encountered among the data, depending on the participants and the context of the employed datasets. This gives rise to another important challenge, which is related to the contextual information in the data, and, should be carefully examined individually.

1.2.4 Context-Specific Attributes

People do not follow a universal pattern when trying to interpret the facial expressions of others. Normally, the human brain analyzes various factors (not only the displayed facial expression), prior to making the decision. Perhaps the most influential factor is the knowledge of the person that performs the particular facial expression. It is well understood that people gesticulate in different ways. For instance, extrovert people are often smiling at higher intensities compared to an introvert person. Apart from the personality traits, age can also affect the appearance of the face. Elder people normally carry wrinkles around the corner of the eyes, without necessary performing the action of cheek raising, *i.e.*, AU6. Furthermore, knowledge of the stimulus (*e.g.*, whether someone is watching a comedy film or a football game) is another of many factors that can influence the meaning of the displayed facial expressions. To summarize the key aspects of the context in which the facial expressions occur, the authors in [141] suggested the W5+ context model. In such a model all the contextual factors can be considered by answering the questions: *who* (the observed subject), *when* (the timing

of the phenomenon), *where* (the environmental characteristics, *e.g.*, view angle, illumination etc.), *why* (the stimulus), *what* (the task related cues), and *how* (they way the expression is conveyed, *e.g.*, by means of intensity levels or activated AUs). Thus, by accounting for (some of) these factors we can achieve a more reliable analysis of facial expressions. However, the majority of the existing works in the literature rely on generic models. These models are expected to generalize well when applied to data recorded within specific contexts. Nevertheless, due to possible variations in these contextual dimensions, especially when dealing with uncotrolled spontaneous data, the performance of these generic approaches is expected to downgrade largely when applied to previously unseen data [68]. Ideally, a proper model for facial expression analysis should take into account all the above contextual factors during training. However, due to the lack of appropriate data, such an approach is not feasible. A more reasonable solution would be to develop mechanisms that can adapt the learned models to the context of the examined situation. As a first step toward this direction, in this thesis we propose a domain adaptation approach that can be used to adapt the context questions *who* (subject) and *where* (view) during test time.

1.3 Contributions

In this section we describe in more detail the main technical contributions of our thesis and we relate them to the aforementioned challenges. For all our proposed models we build upon the Gaussian process framework [146] and introduce novel extensions and learning strategies in order to efficiently deal with the analysis of facial expressions. 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. Although the proposed methodologies have been developed having a specific task in mind, they can be applied to various problems with similar settings, without loss of generality.

- **Chapter 4. Multi-view analysis of facial expressions.** The first problem that we address in this thesis is the multi-view analysis of facial expressions. For this purpose we introduce the discriminative shared Gaussian process latent variable model (DS-GPLVM) for multi-view and view-invariant facial expression classification of basic emotions. The proposed DS-GPLVM is the first approach that exploits the multi-view learning strategy in order to align facial expressions from multiple poses on a common non-linear manifold. To achieve this, we use the notion of Shared GPs [167, 50] to generalize discriminative GP latent variable models [183, 212] to multiple observation

spaces. Hence, in DS-GPLVM the discriminative information is shared among the views. Consequently, classification of facial expressions from under-performing poses is largely improved on the shared manifold. In the proposed DS-GPLVM we can efficiently handle large number of views due to our proposed learning scheme. We first split the training into different sub-problems (one for each view), and then optimize each sub-problem separately. Finally, we demonstrate that the proposed DS-GPLVM is applicable to a variety of tasks (multi-view classification, multiple-feature fusion, pose-wise classification, etc.), a fact which makes it a complete framework for multi-view analysis of facial expressions.

- **Chapter 5. Joint feature fusion and AU detection.** Although the method proposed above is quite general, it has two main limitations: (i) the emotion classifier is learned independently from the manifold; (ii) it cannot handle multiple labels in the output, hence, it is not appropriate for facial expression analysis based on AU detection. To ameliorate this, we propose a multi-conditional latent variable model (MC-LVM) that performs simultaneously the fusion of different facial features and joint detection of AUs. One of the key novelties of the proposed model is that the MC-LVM is derived in a fully Bayesian multi-conditional formulation, and combines the merits of *both* the generative and discriminative probabilistic models, by merging the framework of shared GPs (feature fusion) with logistic classifiers (AU detection). This property, makes the MC-LVM more flexible on generalizing to new data, while also being less susceptible to overfitting. The structure from the output labels is integrated into the manifold through newly introduced constraints during the model learning. *Topological* constraints encode local dependencies (from image pairs) among multiple AUs, while *relational* constraints, enforce the AU co-occurrences of the model predictions to match those of the target labels. We experimentally show that such constraints play an important role in increasing the discriminative power of the learned manifold, resulting in improved (average) detection performance. MC-LVM is one of the first approaches for multiple AU recognition that jointly performs facial feature fusion and AU detection, via manifold learning.

- **Chapter 6. Feature fusion and AU intensity estimation.** Although the MC-LVM can effectively deal with multiple labels in the output, it cannot model the intensity of the facial AUs. Moreover, the structure of the co-occurring AUs is learned from heuristic constraints. We address these limitations by: (i) explicitly modeling the ordinal nature of the AUs and (ii) learning the desired structure directly from the data. Specifically, we propose the variational GP auto-encoder (VGP-AE), which is composed of a probabilistic

GP *encoder*, used to fuse multiple observed features onto a latent space, and a GP *decoder*, used for their reconstruction. Inference of the proposed VGP-AE is performed in a fully Bayesian framework, where the recovered latent representations are further endowed with the ordinal output labels. In this way, we seamlessly integrate the ordinal structure into the recovered manifold while attaining robust fusion of the input features. The fully probabilistic nature of our auto-encoder allows us to explicitly model the uncertainty in the projections onto the learned manifold. This results in learning a well regularized latent space with good generalization abilities. Furthermore, VGP-AE is the first approach that achieves simultaneous fusion of multiple input features and joint AU intensity estimation in the context of facial behavior analysis.

- **Chapter 7. Domain adaptation for facial expression analysis.** The last challenge that we introduced in the previous section is the modeling of context-specific attributes, such as the different levels of expressiveness encountered across the population. To address this challenge, we use the notion of *domain adaptation* to perform view and subject adaptation, for expression classification of basic emotions and AUs. In particular, we generalize prior work on GP experts [43, 25], and introduce domain-specific GPs as local experts for the task of facial expression analysis. We facilitate the adaptation of the classifier in a probabilistic fashion by conditioning the target expert on the predictions from multiple source experts. Our proposed GP domain experts (GPDE) is the first approach that exploits the variance in the predicted expression in order to utilize a measure of confidence for weighting the contribution of each expert. This results in learning a *confident* classifier that minimizes the risk of potential negative transfer (*i.e.*, the adapted model performing worse than the model trained using the target data only). Furthermore, this is the first work in the field of facial behavior modeling that can simultaneously perform adaption to multiple AUs. As we demonstrate in the experiments in Chapter 7 the proposed GPDE can effectively perform adaptation of 12 AUs simultaneously, and outperforms generic and person-specific classifiers, while using as few as 50 target examples. The latter is of remarkable importance, since the annotation of several AUs is a time demanding and tedious task, which can be performed only from well trained personnel.

1.4 Publications

The work presented in this thesis has resulted in the following list of publications:

- **International Conferences**

[1] **S. Eleftheriadis**, O. Rudovic, M. P. Deisenroth, M. Pantic. Variational Gaussian Process Auto-Encoder for Ordinal Prediction of Facial Action Units. In *Asian Conf. on Computer Vision (ACCV). Taipei, Taiwan.* 2016.

[2] **S. Eleftheriadis**, O. Rudovic, M. P. Deisenroth, M. Pantic. Gaussian Process Domain Experts for Model Adaptation in Facial Behavior Analysis. In *Proceedings of* IEEE *Int'l Conf. on Computer Vision and Pattern Recognition (CVPR-W), Workshop on Context Based Affect Recognition (CBAR). Las Vegas, Nevada, USA.* 2016.

[3] **S. Eleftheriadis**, O. Rudovic, M. Pantic. Multi-conditional Latent Variable Model for Joint Facial Action Unit Detection. In *Proceedings of* IEEE *Int'l Conf. on Computer Vision (ICCV), pp. 3792-3800. Santiago, Chile.* 2015.

[4] **S. Eleftheriadis**, O. Rudovic, M. Pantic. View-constrained Latent Variable Model for Multi-view Facial Expression Recognition. In *Advances in Visual Computing:* 10^{th} *Int'l Symposium (ISVC), pp. 292-303. Las Vegas, Nevada, USA.* 2014.

[5] **S. Eleftheriadis**, O. Rudovic, M. Pantic. Shared Gaussian Process Latent Variable Model for Multi-view Facial Expression Recognition. In *Advances in Visual Computing:* 9^{th} *Int'l Symposium (ISVC), pp. 527-538. Rethymnon, Crete, Greece.* 2013.

- **Journal Articles**

[1] **S. Eleftheriadis**, O. Rudovic, M. P. Deisenroth, M. Pantic. Gaussian Process Domain Experts for Modeling Facial Affect. *IEEE Transactions on Image Processing (TIP).* Submitted – under revision.

[2] **S. Eleftheriadis**, O. Rudovic, M. Pantic. Joint Facial Action Unit Detection and Feature Fusion: A Multi-conditional Learning Approach. *IEEE Transactions on Image Processing (TIP). 25(12): pp. 5727-5742.* 2016.

[3] **S. Eleftheriadis**, O. Rudovic, M. Pantic. Discriminative Shared Gaussian Processes for Multi-view and View-invariant Facial Expression Recognition. *IEEE Transactions on Image Processing (TIP). 24(1): pp. 189-204.* 2015.

1.5 Thesis Outline

The rest of the thesis is structured as follows. In Chapter 2 we review the related literature and pay particular attention to the existing machine learning models that have been proposed

for facial expression analysis. In Chapter 3 we briefly present the basics behind the framework of GPs. Subsequently, in Chapter 4 we introduce the proposed discriminative shared Gaussian process latent variable model (DS-GPLVM) to address the problems of multi-view and view-invariant facial expression classification of basic emotions. Chapter 5 introduces the multi-conditional latent variable model (MC-LVM) for joint facial action unit detection and feature fusion. In Chapter 6 we introduce the variational Gaussian process auto-encoder (VGP-AE) for intensity estimation of facial action units. Chapter 7 introduces our Gaussian process domain experts (GPDE) for view and subject adaptation for analysis of facial expressions. Finally, we conclude the thesis in Chapter 8.

Chapter 2

Machine Analysis of Facial Expressions: state-of-the-art

Contents

To date, the majority of the works in the area of facial expression analysis deal with imagery where the subjects are depicted in a (nearly) frontal head pose. Depending on whether they take into account the temporal information of the expression, they can be divided into static and dynamic approaches. The former, typically employ static multi-class classifiers such as rule-based classifiers [20, 143], artificial neural networks (ANN) [136, 176], support vector machine (SVM) [14, 163], Bayesian networks (BN) [30], k-nearest neighbors (kNN) [117], among others. Their main goal 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), on frame-by-frame basis. The approaches that deal with the dynamic classification of the facial expressions are mainly based on hidden Markov models (HMMs) [135, 133, 197, 110, 189, 164, 30]. Their main goal is to isolate the segments in the video sequence that contain a facial expression and perform the emotion recognition within these segments. The common drawback of the aforementioned methods (both static and dynamic) is their inability to operate on off-frontal poses. This modeling practice –mainly driven due to data unavailability in the past– can lead to effective classifiers, yet with limited applicability. Their usage is constrained to applications where the subject is *always* facing towards a camera, *e.g.*, video conferences,

online gaming, etc. However, in real-world scenarios (video summarization, security and surveillance, etc.), we frequently observe spontaneous human-to-human interactions, where the participating subjects perform large out-of-plane head rotations. Hence, in the aforementioned situations, learning only from frontal images would result in degraded performance.

The same modeling practice (*e.g.*, frontal/single view analysis) is also observed in the AU-related literature. Again, the main reason for this is the complete lack of data with AU annotations from corresponding views. However, there is another source of variation in the input facial images that can be treated in a multi-view manner. The term view can, more broadly, refer to any possible descriptor of a given image. Hence, different feature representations, *i.e.*, geometric- and appearance-based features, can be regarded as multiple views. By combining the information of the various features (*i.e.*, feature fusion), within the notion of multi-view learning, we can possibly derive more powerful feature representation. Note that different AUs are better explained from different type of features. For instance, the deformations on the shape caused by the raising of the eyebrows (*i.e.*, AU1,2) can effectively described from geometric features. On the other hand, bulges and wrinkles that appear in the face due to the action of cheek raising (*i.e.*, AU6) are better captured from appearance-based features. Despite that, most of existing approaches for AU analysis use a single type of features; either representing the geometry [151, 189, 90, 12, 179, 140] or the appearance [91, 156, 28, 26, 158, 119, 111, 16] of the face deformations. Lately, some works proposed to combine the information from various features by either concatenating them into one single vector, *i.e.*, early fusion [114, 194, 118, 215], or by combining the results of separate classifiers trained on each modality, *i.e.*, decision-level fusion, [115, 92]. More appropriate solutions for fusing the input features have also been proposed under the framework of multiple kernel learning (MKL) [162, 125]. In all these works it was shown that the fusion of the features was beneficial for the detection of the majority of the AUs. Thus, the improvement in the results suggests that the analysis of facial AUs could further benefit by following a proper multi-view learning strategy.

Another worth exploring area for advancing the facial expression analysis is the relations among the AUs. Several AUs commonly co-occur in a facial expression in order to compose a single basic emotion (*e.g.*, the co-occurrence of AU6+12 or AU6+12+25 in full-blown spontaneous smiles). This implies that the analysis of facial AUs is a multi-label problem compared to the multi-class nature of the basic emotions. However, the majority of the existing works, so far, attempted to recognize AUs or certain AU combinations independently [114, 115, 14, 119, 189, 99]. Hecne, they resolved to construct indepedent classifiers

for each input featue, while ignoring completely the multi-label nature of the problem. For instance, [14] applied independent Adaboost classifiers on the extracted Gabor features from the facial images. Similarly, the authors in [119] encoded the Gabor appearance features into a sparse dictionary of facial images. Yet, this work focused on the detection of certain AU combinations as different classes, instead of recognizing the activations of independent AUs. The authors in [114, 115] employed the SVM classifier to evaluate the performance of the AU detection task for different input features. For the purposes of such comparison, the authors used geometric features based on the landmark locations of a 2D AAM, and appearance features based on raw pixel intensities from the warped facial images. There are also works that employed variants of dynamic Bayesian networks (DBN) (mainly applied on appearance Gabor features) in order to account for the temporal dynamics during the AU detection task. Representative are [99, 189] that used HMMs in combination with GentleBoost and SVM classifiers.

The same strategy is followed even in the more recent works that study the problem of AU intensity estimation. The AU intensity is modeled independently via classification [118, 122, 151, 125, 186] or regression [158, 92, 89, 93] techniques. While the classification methods (normally based on support vector classification (SVC) [33] and conditional random fields (CRF) [104]) seem to be a natural choice to handle the problem, they often struggle from inconsistent results, since they completely ignore the discrete, yet ordinal, state of the labels (missclassification between different states are equally penalized). On the other hand, modeling the intensity levels on a continuous scale, like the regression based methods (*e.g.*, support vector regression (SVR) [169]), is sub-optimal due to the fact that the various intensities span on a different range [54].

Regardless of the addressed problem (*i.e.*, detection or intensity estimation) or the modeling technique (*e.g.*, regression, classification, temporal modeling), none of the above methods takes into account the dependencies among the AUs. Hence, they ignore to model any co-occurrence structure between the outputs, which may result in low performance when data from certain AUs are scarce. Another common limitation of all these works is that they rely on *generic* classifiers. With the term generic we refer to simple classifiers that are trained on all available data which are assumed to encode all possible variations of the population. Hence, the performance of these classifiers is expected to degrade when applied to previously unseen data [68]. Such a scenario is the case when we try to infer the facial expression of a new subject, whose level of expressiveness varies significantly compared to the training subjects. These individual differences among subjects have mainly been tackled by accounting for the

subject information at the training stage. Specifically, the original feature set is extended by adding the subject-specific features [151], or by building person-specific classifiers [188]. Although these approaches showed improvement over generic classifiers, their main limitation is that for building personalized classifiers, access to an adequate collection of images of the target person is essential.

In the remaining of this chapter, we first review existing approaches for multi-view facial expression recognition of the basic emotions, and then proceed to methods for joint AU detection and intensity estimation. We then review approaches for personalized analysis of facial expressions. Lastly, we relate those works to the methods proposed in this thesis.

2.1 Multi-view and View-invariant Facial Expression Recognition

The first step towards rectifying the limitations of the frontal-based analysis of facial expressions had been achieved by the collection of more appropriate data, acquired from multiple views (*e.g.*, the BU-3DFE [193] and MultiPIE [76] datasets). Subsequently, several approaches have been proposed recently for the multi-view facial expression recognition of six basic emotions. Based on how they deal with variation in the head-pose (view) and expressions in 2D images, they can be divided into: (i) *pose-wise*, (ii) *pose-independent* and (iii) *pose-normalized* models. The methods from the first category treat each view as a separate problem. Hence, different models are trained independently per view. On the other hand, the approaches from the second group operate on a completely orthogonal direction. A universal model is learned from data from multiple views. Finally, the methods from the third group attempt to learn a mapping between frontal and non-frontal images, in order to normalize the pose before the classification task.

Pose-wise facial expression recognition. A representative of the first group is [127], where the authors used local binary patterns (LBPs) [131] (and its variants) to perform a two-step facial expression classification. In the first step, the colsest head-pose to the (discrete) training pose was selected via the SVM classifier. Once the view was obtained, the task of facial expression recognition was handled via another set of pose-specific SVM classifiers. This approach was evaluated on synthetic images generated from BU-3DFE at five yaw angles ($0° - 90°$), and posed expressions from MultiPIE at seven yaw angles ($0° - 90°$). In [86], the performance of different appearance features (SIFT, HOG, and LBPs), extracted from synthetic images from BU-3DFE, was tested under 5 yaw angles ($0° - 90°$). The

various features were extracted around the locations of characteristic facial points, and were used as input to train pose-specific kNN classifiers. An important outcome of [86] was the experimental proof that the two-stage multi-view facial expression recognition performed better than considering all the combinations between available views and emotions as separate classes. Motivated by these results, the authors in [85] evaluated the performance of different classifiers on the same yaw angles from BU-3DFE, and found that the SVM performed better, on average. In a similar study [81], the authors used per-view trained 2D AAMs to locate a set of characteristic facial points over thirteen yaw angles ($-90°$ – $90°$). The obtained points were used as landmarks to extract LBP, SIFT and DCT features around them. Pose-specific SVM classifiers showed that the combination of the geometric features from the AAMs with the DCT appearance-based features, achieved the best average performance. Nonetheless, the main limitation of the *pose-wise* classifiers is that they treat each view as an independent problem. Hence, they require a sufficient amount of training data per view, in order to learn effective classifiers. Furthermore, by learning view-specific classifiers, these approaches fail to model possible correlations between the features from the various poses, which can result in lower average performance.

Pose-independent facial expression recognition. As mentioned above, the methods of this group attempt to learn a single classifier by combining the available data from multiple poses. Specifically, [210] used variants of dense SIFT [113] features extracted from expressive images over seven yaw angles ($0°$ – $90°$), and five pitch angles ($-30°$ – $30°$). A universal linear classifier was then trained on the concatenation of the SIFT features from all thirty five views. It is worth noting that dimensionality reduction based on Gaussian mixture models (GMM) [19] was proposed from the authors, in order to facilitate an efficient training of the classifier. Likewise, [175] used the generic sparse coding scheme (GSC) [195] to learn a dictionary that sparsely encodes the SIFT features extracted from the same twelve views. After obtaining the relevant dictionary for a given test image, linear classification was used again to perform the facial expression recognition. Although the methods of this category seem to deal effectively with arbitrary views, they have certain limitations. Due to the high variation in appearance of facial expressions in different views and of different subjects, the complexity of the learned classifier increases significantly with the number of views and expressions. This can easily lead to overfitting, and, in turn, poor generalization of the classifier to unseen data.

Pose-normalized facial expression recognition. The approaches that fall in this category rely on known correspondences between facial images from various poses. Given that

correspondence, a regression function can be trained in order to pair the input features between any pair of poses. Representatives of this *pose-normalization* approach are [148, 149]. In these methods, the authors first perform the view normalization, and then apply facial expression classification in the canonical view. The latter is usually chosen to be the frontal view. For the view normalization, the authors employed the coupled GP (CGP) regression model that exploits pair-wise correlations between the views, in order to learn robust mappings for projecting facial features (*i.e.*, a set of facial points) from an arbitrary pose to the frontal. In a similar work, the authors in [9] used again the GP regression, yet for modeling the opposite mapping. The pose-specific facial points are obtained from the frontal ones. Subsequently, these points were used as landmarks to produce virtual images, by warping the appearance from the frontal to the desired view. The resulting facial image is used for the emotion classification task. Likewise, the authors in [88] encoded different appearance-based features (HOG and LBPs) in a sparse dictionary using k-singular value decomposition (k-SVD) [3]. A linear regression was learned to map the dictionaries between an arbitrary view and the frontal. The facial expression recognition of six basic emotion was performed on the reconstructed facial features from the normalized dictionary. A common limitation of all the *pose-normalized* approaches is that the view normalization and learning of the expression classifier are done independently. Thus, the classification's accuracy is bounded by that of the view normalization, since any errors in the latter can adversely affect the performance of the recognition task. Furthermore, the canonical view has to be selected in advance. This can further limit the accuracy of the expression classification, as such view may not be the most discriminative for classification of certain facial expression categories.

2.2 Joint Action Unit Detection and Intensity Estimation

The works mentioned above focus solely on the classification of the basic emotion categories from multiple poses. When it comes to the analysis of facial AUs, due to the lack of available annotated multi-view data, the area remains still unexplored. However, as we already explained, the different image descriptors, *i.e.*, geometric- and appearance-based features, can be regarded as multiple views. Hence, multi-view learning can be employed to model the variation between the input features. Furthermore, an even more important source of variation in the analysis of AUs stems from the nature of the output. The analysis of facial AUs is a multi-label problem compared to the multi-class nature of the basic emotions. Thus, a holistic analysis would need to consider also the correlations among the multiple outputs. In what follows we review the works that placed the AU analysis on the frames of multi-view learning

(feature fusion) and multi-label learning (joint modeling of the outputs).

2.2.1 Joint Facial AU Detection

As we have explained above, a holistic analysis of facial AUs suggests jointly modeling of the relations among the input features and the highly correlated outputs. However, the majority of the existing works, so far, attempted to recognize AUs or certain AU combinations independently [114, 115, 14, 119, 189, 99]. While the former approach ignores the dependencies among the AUs, the latter results in a prohibitively large space of possible combinations. To the best of our knowledge, there are only few works that perform joint detection of AUs [180, 215, 194, 207, 205, 206, 171, 209]. Towards this direction, the authors in [180] proposed a two stage strategy. First they applied independent Adaboost classifiers for each AU on Gabor features extracted from the facial images. Then a generative DBN is employed to model the dependencies among the various AUs and refine the classifiers' predictions. Due to the Markov assumptions while learning the network of the co-occurred AUs, this model can handle only local dependencies between pairs of AUs. The same two stage approach was also followd by [215], yet, the authors considered the information from both geometric and appearance features (2D landmark points and Gabor wavelets). Specifically, the logistic classifiers for multiple AUs was first learned on the concatenated features, by using the notion of multi-task feature learning [8]. Then, a similar pre-trained BN was employed to refine the predictions. Hence, the same limitation as of [180] also apply to [215]. Nontheless, the main drawback of both [180, 215] lies on their two-stage training scheme. The independent modeling of the discriminative classifiers and the generative DBN could result in inconsistent learned dependencies across inputs/outputs, and hence, produce contradictory predictions.

On the other hand, the models in [194, 207, 205, 206, 171, 209] are defined in a fully discriminative framework. More specifically, [194] employed the restricted Boltzmann machine (RBM) to overcome the pair-wise AU modeling limitation of the DBN [180, 215]. The authors proposed a parametric model, in which *discrete* latent variables account for correlations among discrete outputs that are directly connected to the image features. The latter are comprised again from a combination of 2D landmark points and raw pixel intensities, obtained from the warped images. Since the latent variables are not connected to the feature space, they cannot model correlations between the inputs, hence, *concatenation* of the input features is used for the fusion task. [207] combined multi-task learning with MKL techniques, in order to jointly learn the AU-specific SVM classifiers for different appearance features (LBP and HOG). This work has been extended in [205, 206], where the authors introduced an l_p-norm regularization

to the MKL problem, in order to obtain a more robust solution with possible sparse structure among the AUs. However, all three MKL methods, *i.e.*, [207, 205, 206] due to their expensive learning complexity, can only deal with a small subset of AUs (typically less than 4) in the output.

In a more recent work, [209] used again the notion of multi-task learning in order to learn multiple logistic classifiers for each AU. The learned dependencies among the AUs were additionally constrained to be sparse, via appropriate regularizations based on positive and negative AU co-occurrences. Simultaneously to the AU detection task, [209] performed feature selection in order to preserve a sparse subset of SIFT appearance features, extracted from patches around the face, that are more relevant to each AU. Yet, the feature fusion task was not addressed. More importantly, the learned AU-dependencies were regarded only between predefined pairs of AUs. Likewise, [171] proposed a probabilistic framework, based on Bayesian compressed sensing (BCS), in order to encode the co-occurrence structure and the (group) sparsity patterns of the AUs to the compressed signal (latent variables). HOG features extracted from different pyramid levels served as the input features, and were mapped to the latent variables via a linear regression. Hence, neither this work addresses the problem of fusing different input features.

2.2.2 Joint AU Intensity Estimation

The works described on the previous section focus solely on the detection of AU activations (*i.e.*, presence/absence). However, the true nature of the AUs is not binary, since they appear in various levels. The AU intensity analysis is relative new problem in the field, and most of the proposed works focus on independent modeling of the AUs [151, 118, 122, 125, 186, 158, 92, 89, 93]. Hence, they fail to account for the structured relations among the AUs. Moreover, except for [158, 125] none of these works can naturally handle the case where we have different modalities in the input (*e.g.*, fusion of geometric and appearance features). This can adversely affect the models' performance, since different AUs are better described from different modalities (*e.g.*, AU1+2 from geometric and AU6 from appearance features).

Only recently, the joint estimation of the intensity levels has been addressed [109, 156, 94, 129, 77, 126]. This is motivated by the fact that intensity annotations are difficult to obtain (due to the tedious process of manually coding) and that AU levels are highly imbalanced. Thus, by imposing the structure on the output in terms of AU co-occurrences, a more robust intensity estimation is expected. Towards this direction, [109] employed the same two-stage learning strategy as the one encountered on the works in AU detection. The authors first

trained a multi-class SVM and then infered a DBN in order to capture the semantic relationships among the various AUs. Likewise, the authors in [156] followed a similar approach. First, they trained individual SVR for estimating the intensity of each AU over appearance based features. Then, they fed the predictions into a Markov random field (MRF) in order to model the dependencies between the AUs and improve the performance. Again, similarly to [180, 215], this two-stage approach followed in both [109, 156] limits their recognition performance, since learning of the regressors/classifiers and the AU relations are handled independently. Moreover, both [109, 156] use information only from appearance features, which makes them more susceptible to subject and illumination variations.

To overcome the limitatios of the previous works, the authors in [94] proposed to learn latent representations which encode the information of the input features and the output labels. The structure of the latent variables is governed by the relations among the AUs, and it is constrained to form a tree graph. However, in the presence of high-dimensional inputs and multiple AUs, this method becomes prohibitively expensive. Moreover, the authors show that with this approach the fusion of different features does not benefit the estimation of AU intensity, achieving similar performance to when individual modalities are used. More recently, a learning method based on sparse representation has been proposed in [126]. Specifically, the authors use the notion of robust principal component analysis [24] to decompose the expression from facial identity. Then, joint intensity estimation of multiple AUs is performed via a regression model based on dictionary learning. Yet, this approach uses only appearance features, thus, it cannot benefit from the information of illumination invariant geometric features. The authors in [129] cast the joint AU intensity estimation in a multi-task formulation. They employed metric learning for kernel regression (MLKR) in order to find an optimal subspace where the error from all tasks is reduced. The main drawback of [129] is that the use of MLKR becomes prohibitive when dealing with high dimensional features, let alone when using features of different modalities (*e.g.*, fusion of geometric and appearance features).

Finally, the recent developments in the deep networks literature inspired the authors in [77] to train deep convolutional neural networks (CNN) for automatic feature extraction and AU intensity estimation. This work showed some promising results in the recent FERA2015 challenge [186]. However, deep networks normally require large amount of annotated data for their effective training. This is a burden since, AU-coded data are still scarcely available.

2.3 Domain Adaptation for Personalized Analysis of Facial Expressions

The methods that we have encountered in Sec. 2.1 and Sec. 2.2, no matter their employed strategy, and whether they model the relations between the inputs/outputs, suffer from a common limitation. That is their poor generalization to new unseen data, an artifact which especially appears when dealing with spontaneous data of facial expressions. A reasonable and cost-efficient way to deal with this problem is to normalize the data based on some person-specific attributes. For instance, the authors in [11] suggested to perform a dynamic normalization of the expression phenomena. They achieved so by removing the global neutral expression, per subject, from each available sequence. Hence, the resulted geometric and appearance descriptors hold only the relevant information regarding the individual facial deformations that are responsible for every expression, and not general face variations. However, the problem with this technique is that it does not take into account the different levels of expressiveness between the subjects. Thus, the normalized features may still suffer from inconsistencies, especially in the cases of subtle expressions. Lately, in order to ameliorate this effect, recent advances in the field focus on employing standard domain adaptation techniques for building personalized classifiers.

In the domain adaptation literature, normally the data between the phases of training and testing are treated as data from different domains. Thus, all subjects that are present during the training of a classifier are considered to belong to the source domain, while the data from the test target belong to the target domain. The ultimate goal of domain adaptation is to bridge the gap between the two domains. A widely used algorithm for adaptation is the kernel mean matching (KMM) [74], which directly infers resampling weights by matching training and test distributions. Towards this attempt, the authors in [28] employed the KMM to learn person-specific, independent AU detectors. This is attained by modifying the SVM's cost function to account for the mismatch in the distribution between source and target domain, while also adjusting the SVM's hyper-plane to the target test data. Although proven to be effective, this transductive learning approach is inefficient. This is due to the fact that for each target subject a new classifier has to be relearned during inference. Likewise, the authors in [124] proposed a supervised extension to the KMM. More specifically, they used the provided labeled examples from both domains in order to align the source and target distributions in a class-to-class manner. The reweighted source data along with the target data, form the input features that are used to train several classifiers, *e.g.*, SVM, for facial expression recognition of basic emotions.

Apart from KMM, adaptation can be also attained by combining the knowledge from multiple classifiers or by sharing the parameter space between source and target classifiers. In [26], a two-step learning approach was proposed for person-specific pain recognition and AU detection. First, the input data of each subject were regarded as different source domains, and were used to train weak Adaboost classifiers. Then, the weak classifiers were weighted based on their classification performance on the available target data. A second boosting was performed on the best performing source classifiers in order to derive the final set of weak classifiers for the target data. In [157, 199], the Adaboost classifiers were replaced with the linear SVMs. First, independent AU classifiers were trained from the source domain data. Then, the SVR framework was employed to associate the input features with the classifiers' parameters. Finally, the unlabeled target domain data were fed into the learned regressors, in order to obtain the target-specific classifier's parameters. Recently, a more suitable approach has been proposed in [200]. The authors suggested to train target-specific classifiers by exploiting the confidence in the predictions from the source classifiers. In their approach, the confidence is represented by the agreement in the predictions between a pair of SVM classifiers, which were trained to distinguish the easy-positive and easy-negative samples in the source data. The confident classifiers are then employed to obtain 'virtual' labels for a portion of the target data, which can be used to train a target-specific detector.

Note that, apart from [26], all the works mentioned above perform in the unsupervised adaptation setting. While this requires less effort in terms of obtaining the labels for the target sub-sample, its underlying assumption is that target data can be well represented as a weighted combination of the source data. However, in real-world data, this assumption can easily be violated, resulting in poor performance of the adapted classifier. A further limitation of the aforementioned methods is that none of them exploits the multi-label nature of the AU detection problem. Hence, not only they fail to model the relations among the various AUs, but they also need to adapt each AU-specific classifier independently.

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. 2.1–2.3. In what follows, we discuss similarities and differences of existing approaches to the methods proposed in this thesis. We relate/contrast these methods in the context of the target problems that we address.

Multi-view and view-invariant classification of facial expressions. In Chapter 4 we propose a method for multi-view and view-invariant facial expression recognition of basic emotion categories. In contrast to the approaches from Sec. 2.1 that operate either on a pose-wise manner or normalize the pose to a canonical view, the multi-view method that we propose performs the pose normalization implicitly on a discriminative manifold shared among multiple views of facial expressions. The classification of an observed facial expression can be carried out either in the view-invariant manner (using only a single view of the expression) or in the multi-view manner (using multiple views of the expression). However, instead of learning independent classifiers, as in the pose-wise classification methods, we learn a single classifier in the low-dimensional manifold. Compared to the pose-independent methods, the complexity of our classifier is significantly reduced. This is due to the fact that we account for the underlying structure of the data (*i.e.*, the correspondences between the views) via the shared latent variables. Thus, within the proposed learning strategy we can directly relate the performance of our proposed method to both pose-wise and pose-independent approaches from Sec. 2.1, in a unified framework. As we show in the experimental analysis of Chapter 4, modeling of the dependencies among the views in the shared subspace, not only results in improved performance compared to state-of-the art, but also improves the accuracy in underperforming views. The latter implies that more robust classification can be attained via our proposed method.

Joint feature fusion and multiple AU detection. The method that we propose for this task advances the existing work from Sec. 2.2.1 in many aspects. First of all, both the problems of feature fusion and joint AU detection are addressed, simultaneously, within a single latent variable model. Specifically, the method that we propose in Chapter 5 performs feature fusion in a generative fashion via a low-dimensional shared subspace, while simultaneously performing AU detection using a discriminative classification approach. The learned low-dimensional manifold allows the model to capture dependencies among multiple AUs at both feature and model level. Contrary to the methods from Sec. 2.2.1, which are purely generative or discriminative, as we show in Chapter 5, our joint formulation takes the best of both approaches and successfully combines them in a multi-conditional likelihood function. Due to the latter, the proposed model is less susceptible to overfitting compared to purely discriminative models, since the generative part acts as an efficient regularizer during parameter learning. Additional regularizations are also considered during the training of our method, in order to constrain the latent variables to preserve the local structure in the outputs. This has not been addressed from the latent variable approaches [194, 171] that solely model the AU dependencies only via

their multi-task formulation. Note that the proposed fusion technique has some similarities to the MKL-based fusion methods [205, 206]. The latter perform the feature fusion implicitly via the kernel-induced space, while our manifold-based approach does it explicitly via the fixed point estimate of the shared low-dimensional latent projections. Finally, the complexity of the proposed approach scales linearly to the number of AUs in the output. Consequently, we can efficiently model relations among a relatively large number of outputs, without the requirement to *a priori* define groups of highly correlated AUs as done in [206, 209].

Joint feature fusion and AU intensity estimation. In the method for intensity estimation of facial AUs, introduced in Chapter 6, we generalize the latent variable model mentioned above to account for the ordinal labels in the output. The work presented in Chapter 6 advances the current state-of-the-art in several aspects. First of all, our approach can efficiently perform the fusion of multiple modalities by means of a shared manifold, while simultaneously dealing with the problem of joint AU intensity estimation. This is in contrast to most of the works from Sec. 2.2.2 that either do not address the feature fusion problem or fail to attain an improved performance when both modalities are used, *e.g.*, [94]. The automatic feature selection is implicitly attained via a latent space, which is learned in an auto-encoding manner. Thus, oposing to the expensive dimensionality reduction apprach of [129], we can automatically perform feature selection via the manifold in an efficient probabilistci approach. Furthermore, for the AU intensity estimation part, we employ the more appropriate framework of *ordinal* regression [2]. The recovered latent representations are used as input to multiple ordinal regressors [2], which are concurrently learned in a joint Bayesian training. Finally the use of the kernel-based GPs allow us to efficiently deal with high-dimensional input and output variables without (significantly) affecting the model's complexity.

Context adaptation for facial expression analysis. Apart from [26], all the works mentioned in Sec. 2.3 perform in the unsupervised adaptation setting. While this requires less effort in terms of obtaining the labels for the target sub-sample, it can have a negative impact on the final classification when the distribution between the source and target data vary significantly. On the other hand, in the method we propose in Chapter 7 we adopt a supervised approach that needs only few annotated data from target domain to perform the adaptation. This, in turn, allows us to define both target and source experts, by means of individual GP regressors, assuring that the performance of the resulting classifier is not constrained by the distribution of the source data. Hence, contrary to the works from Sec. 2.3 that perform the adaptation by adjusting the classifiers' parameters and minimizing the error between the dis-

tributions of the original source and target domain data, we follow a different approach. We achieve domain adaptation in a Bayesian fashion, and explain the target data by conditioning on the learned source experts. Note that except for [200], none of the methods from Sec. 2.3 provide a measure of confidence in the predicted labels. Yet, even in [200] the confidence is obtained in a heuristic manner and is not directly related to the prediction of the classifier. On the contrary, in our probabilistic approach, we model the confidence by means of predicted variance. Finally, oppsing to transductive adaptation approaches (*e.g.*, [28]) that need to be re-trained completely, the adaptation in our proposed method is efficient and requires no re-training of the source model.

Chapter 3

Gaussian Processes: Background Overview

Contents

In many real-world applications in the fields of computer vision and pattern recognition, the practical problem consist of learning the underlying function f, that can associate some observed inputs (*e.g.*, geometric or appearance-based facial features in our case), with the corresponding outputs (*e.g.*, facial expressions or facial muscle configuration). The dominant approach so far is to train *parametric* models, *i.e.*, assume that the underlying function can be adequately described by some parameters, normally a set of weights that measure the interactions between the inputs. The main limitation of the parametric approach is the original assumption regarding a *finite* set of parameters. This practically means that given the parameters, any future predictions are independent of the observed data. Hence, the complexity of any parametric model is bounded even if the amount of data is unbounded. A more flexible approach would be to learn *non-parametric* models, *i.e.*, assume that the distribution of the data can only be defined in terms of an *infinite* dimensional set of parameters. We normally think of this infinite set as the mapping function f, which can be naturally modeled within the framework of Gaussian processes (GPs) [146].

In what follows, we first give more intuition regarding the importance of GPs to our line

of research. Then we present the basics behind the framework of GPs, in order to provide the reader with the appropriate methodological background, prior to presenting our proposed models in the upcoming chapters.

3.1 Why Gaussian Processes?

The main goal in our approach to automated analysis of facial expressions is to learn high-dimensional mappings between the corresponding facial features and the associated output labels. We can tackle this problem by following either a supervised learning approach (*i.e.*, regression/classification), or an unsupervised learning approach (*i.e.*, dimensionality reduction). In the former case, we aim to map directly the facial features to the output labels, while in the latter we aim to find a low-dimensional manifold where the facial features and the output labels are coupled together. In what follows, we outline the key strengths of GPs that make them particularly suitable for the target tasks.

- GPs, as a fully probabilistic framework, can naturally provide a well calibrated uncertainty in their predictions. The importance of modeling the uncertainty is twofold: (i) Latent variables can be learned as random variables with known probability distributions. Hence, latent samples can be efficiently collected in order to facilitate a fully Bayesian training of the models (see Chapters 5&6). Consequently, the automatic regularization from the Bayesian framework, allows us to learn models that are robust to overfitting, and also capable of generalizing well to new settings; (ii) The learned uncertainty can be used to design gating functions for combining predictions from different mapping functions learned with GPs. We use this mechanism to perform domain adaptation during the analysis of facial expressions (Chapter 7).

- 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 model the interactions among the features, which are responsible for preserving the facial-expression-specific details during the learning of either the low-dimensional manifold or the direct mapping function.

- Prior knowledge can be easily introduced during the learning of latent variables using GPs. We use this property of GPs to incorporate two types of priors: (i) 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 4); (ii) The structured output prior, defined again via the Laplacian matrix, which now encodes the information regarding co-occurring patterns in the multiple outputs. This results in a model with structured output that we use for multi-label classification (Chapter 5).

3.2 Gaussian Processes

A Gaussian process [146] is a generalization of the multivariate Gaussian distribution to an infinite number of dimensions (random variables). A sample from a Gaussian process is a random function f that models the relationship between two variables, *i.e.*, $f : \boldsymbol{X} \rightarrow \boldsymbol{Y}$. $\boldsymbol{X}$ and $\boldsymbol{Y}$ are usually corresponding multivariate instances $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^N$ and $\boldsymbol{Y} = \{\boldsymbol{y}_i\}_{i=1}^N$, with $\boldsymbol{x}_i \in \mathbb{R}^q$ and $\boldsymbol{y}_i \in \mathbb{R}^D$. Hence, a Gaussian process can be regarded as a collection of functions, any finite number of which have a jointly Gaussian distribution. This definition highlights the expressive power of Gaussian processes, which along with the tractable marginalization of Gaussian distributions allow us to only work with a finite set of function instantiations $\boldsymbol{f}_{:,j} = f_{:,j}(\boldsymbol{x}_{i,:}) = [f_{:,j}(\boldsymbol{x}_1), f_{:,j}(\boldsymbol{x}_2), \cdots, f_{:,j}(\boldsymbol{x}_n)]$[1], which constitute our observed data and jointly follow a marginal Gaussian distribution. This implies that all other (possibly infinite) function values corresponding to unseen inputs are just marginalized.

More formally, we consider the case where f operates as a mapping function between two variables $\boldsymbol{X}, \boldsymbol{Y}$. We assume that each dimension of the observed output $\boldsymbol{y}_i$ is a noisy observation of the function instantiation $\boldsymbol{f}_{:,j}$ corrupted with Gaussian noise $\epsilon_{i,j} \sim \mathcal{N}(0, \sigma_n^2)$, where σ_n^2 is the variance of the noisy process, so that

$$y_{i,j} = f_{:,j}(\boldsymbol{x}_i) + \epsilon_{i,j}. \tag{3.1}$$

Here, all mapping functions are assigned a GP prior, and hence, the process can be parameterized by its mean $\mu(\boldsymbol{x})$ and covariance function $k(\boldsymbol{x}, \boldsymbol{x}')$, so that $f \sim \mathcal{GP}(\mu(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}'))$. Commonly, the mean function is selected to be the constant zero vector $\mathbf{0}$. The covariance function operates on the infinite input domain and can be parameterized by a set of hyper-parameters $\boldsymbol{\theta}$. A widely used covariance function is the radial basis function (RBF)

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp(-\frac{1}{2\ell^2}\|\mathbf{x} - \mathbf{x}'\|^2), \tag{3.2}$$

where the signal variance σ_f^2 and the length scale ℓ constitute the set of hyper-parameters.

[1]Note that throughout this chapter the subscript ':' denotes stacking of the variables along the operating dimension.

Thus, we end up with the distribution of the function values

$$p(\boldsymbol{F}|\boldsymbol{X},\boldsymbol{\theta}) = \prod_{j=1}^{D} p(\boldsymbol{f}_{:,j}|\boldsymbol{X},\boldsymbol{\theta}) = \prod_{j=1}^{D} \mathcal{N}(\boldsymbol{0},\boldsymbol{K}), \tag{3.3}$$

where $\boldsymbol{F} = \{\boldsymbol{f}_{:,j}\}_{j=1}^{D}$ is the collection of the finite set of function instantiations, and $\boldsymbol{K} = k(\boldsymbol{X},\boldsymbol{X})$ is the covariance matrix, obtained from evaluating the covariance function on the available finite instances.

Since f in Eq. (3.1) follows a Gaussian distribution, the observed output is also Gaussian with

$$p(\boldsymbol{Y}|\boldsymbol{F}) = \prod_{j=1}^{D} p(\boldsymbol{y}_{:,j}|\boldsymbol{f}_{:,j}) = \prod_{j=1}^{D} \mathcal{N}(\boldsymbol{f}_{:,j}, \sigma_n^2\boldsymbol{I}). \tag{3.4}$$

Marginalization over the infinite set of function values yields the marginal likelihood of the observed outputs given the observed inputs

$$p(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{\theta}) = \prod_{j=1}^{D} p(\boldsymbol{y}_{:,j}|\boldsymbol{X},\boldsymbol{\theta}) = \prod_{j=1}^{D} \mathcal{N}(\boldsymbol{0}, \boldsymbol{K} + \sigma_n^2\boldsymbol{I}). \tag{3.5}$$

Traditionally, in the GP literature the hyper-parameters and the noise variance are learned jointly by maximizing the above marginal likelihood w.r.t. $\{\boldsymbol{\theta}, \sigma_n\}$. By expanding the marginal likelihood as:

$$p(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{\theta}) = \prod_{j=1}^{D} (2\pi)^{-\frac{n}{2}} \left|\boldsymbol{K} + \sigma_n^2\boldsymbol{I}\right|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\boldsymbol{y}_{:,j}^T(\boldsymbol{K} + \sigma_n^2\boldsymbol{I})^{-1}\boldsymbol{y}_{:,j}\right) \tag{3.6}$$

we identify the dual purpose of this objective function: (i) the determinant penalizes complex models, and hence, acts as a natural regularization preventing the model from overfitting, whereas, (ii) the exponential term promotes a good fit to the data.

Once the model's hyper-parameters have been found, the predictive distribution for a new input vector $\boldsymbol{x}_*$ can be obtained by conditioning on all the available training instances

$$p(\boldsymbol{f}_*|\boldsymbol{Y},\boldsymbol{X},\boldsymbol{x}_*) = \mathcal{N}(\boldsymbol{k}_*^T(\boldsymbol{K} + \sigma_n^2\boldsymbol{I})^{-1}\boldsymbol{Y}, \quad k_{**} - \boldsymbol{k}_*^T(\boldsymbol{K} + \sigma_n^2\boldsymbol{I})^{-1}\boldsymbol{k}_*), \tag{3.7}$$

where $\boldsymbol{f}_*$ is the predicted function value, $\boldsymbol{k}_* = k(\boldsymbol{x},\boldsymbol{x}_*)$ and $k_{**} = k(\boldsymbol{x}_*,\boldsymbol{x}_*)$.

3.3 Gaussian Processes with Latent Inputs

In the previous section we demonstrated how we can place probabilistic priors over a family of functions, in order to learn robust and accurate non-linear mappings between input/output data pairs. Herein, we present the Gaussian process latent variable model (GPLVM) [105], an unsupervised flavor of GPs used for non-linear dimensionality reduction.

3.3.1 GPLVM

We assume a similar setting to that of a GP regression from Sec. 3.2, where $\boldsymbol{Y} \in \mathbb{R}^{N \times D}$ and $\boldsymbol{X} \in \mathbb{R}^{N \times q}$. The difference now is that we observe only the high-dimensional outputs $\boldsymbol{Y}$ ($q \ll D$), while the inputs $\boldsymbol{X}$ are considered to be *latent*. The same (noisy) generative process of Eq. (3.1) also applies here. Specifically, each dimension of the observations $\boldsymbol{y}_{:,j}$ is assumed to be generated from the same low-dimensional latent variable $\boldsymbol{X}$ via a GP mapping f. Note that independent GP priors are placed over each dimension of the function values, $\boldsymbol{f}_{:,j}$, while the hyper-parameters $\boldsymbol{\theta}$ of the covariance matrix $\boldsymbol{K}$, are assumed to be shared across the independent processes. Hence, our GP prior has the form of

$$p(\boldsymbol{F}|\boldsymbol{X}, \boldsymbol{\theta}) = \prod_{j=1}^{D} (2\pi)^{-\frac{n}{2}} \left|\boldsymbol{K} + \sigma_n^2 \boldsymbol{I}\right|^{-\frac{1}{2}} \exp\left(-\frac{1}{2} \boldsymbol{f}_{:,j}^T (\boldsymbol{K} + \sigma_n^2 \boldsymbol{I})^{-1} \boldsymbol{f}_{:,j}\right). \tag{3.8}$$

The difference here compared to the standard GP regression is that now the inputs $\boldsymbol{X}$ to the kernel functions are latent random variables. Thus, they can be assigned prior distributions $p(\boldsymbol{X})$. The choice and the constructions of this prior usually depends on the task at hand. For now we keep this structure unspecified, in order to facilitate a more general discussion. By following the same derivation to the standard GP regression, we end up with the marginal likelihood of the observed data given the latent variables

$$p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\theta}) = \frac{1}{\sqrt{(2\pi)^{ND} |\boldsymbol{K} + \sigma_n^2 \boldsymbol{I}|^D}} \exp\left[-\frac{1}{2} \mathrm{tr}\left((\boldsymbol{K} + \sigma_n^2 \boldsymbol{I})^{-1} \boldsymbol{Y} \boldsymbol{Y}^T\right)\right]. \tag{3.9}$$

Since we have access to both the marginal likelihood and the prior of the latent variables, we can follow a maximum a posteriori (MAP) training procedure as in [105], in order to obtain the fixed points estimates of the latent variables $\boldsymbol{X}$ as the mean of the posterior distribution

$$p(\boldsymbol{X}, \boldsymbol{\theta}|\boldsymbol{Y}) \propto p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\theta}) p(\boldsymbol{X}). \tag{3.10}$$

Hence, learning of the GPLVM can be facilitated by minimizing the negative log posterior, given by

$$L = \frac{D}{2} \ln |\boldsymbol{K} + \sigma_n^2 \boldsymbol{I}| + \frac{1}{2} \mathrm{tr}\left((\boldsymbol{K} + \sigma_n^2 \boldsymbol{I})^{-1} \boldsymbol{Y} \boldsymbol{Y}^T\right) - \log p(\boldsymbol{X}) + const., \tag{3.11}$$

w.r.t. the latent coordinates $\boldsymbol{X}$, as well as the hyper-parameters $\boldsymbol{\theta}$.

3.3.2 Different Latent Space Priors and Back-constraints

The original GPLVM is a generative model of the observed data, where a simple spherical Gaussian prior is placed over the manifold, similar to

$$p(\boldsymbol{X}) = \prod_{i=1}^{N} (\boldsymbol{x}_i|\boldsymbol{0}, \boldsymbol{I}) = \prod_{i=1}^{N} \prod_{j=1}^{q} N(x_{i,j}|0, 1). \tag{3.12}$$

Such a prior prevents the GPLVM from placing the latent points infinitely far apart, *i.e.*, latent positions close to the origin are preferred. However, we can also introduce more specific priors, more appropriate to the task at hand (*i.e.*, facial expression analysis), in order to impose discriminative information in the manifold and obtain a latent space with good class separation. This has firstly been explored in [183], where a prior based on linear discriminant analysis (LDA) is proposed. LDA tries to maximize between-class separability and minimize within-class variability by maximizing

$$J(\boldsymbol{X}) = \text{tr}(\boldsymbol{S}_w^{-1}\boldsymbol{S}_b), \tag{3.13}$$

where $\boldsymbol{S}_w$ and $\boldsymbol{S}_b$ are within- and between-class matrices, respectively, defined as

$$\boldsymbol{S}_w = \sum_{c=1}^{C} \frac{N_c}{N} \left[\frac{1}{N_c} \sum_{i=1}^{N_c} (\boldsymbol{x}_i^{(c)} - \boldsymbol{M}_c)(\boldsymbol{x}_i^{(c)} - \boldsymbol{M}_c)^T \right], \tag{3.14}$$

$$\boldsymbol{S}_b = \sum_{i=1}^{C} \frac{N_c}{N} (\boldsymbol{M}_c - \boldsymbol{M}_0)(\boldsymbol{M}_c - \boldsymbol{M}_0)^T. \tag{3.15}$$

Here, N_c training points from class c are stored in $\boldsymbol{X}^{(c)} = [\boldsymbol{x}_1^{(c)}, \ldots, \boldsymbol{x}_{N_c}^{(c)}]^T$, $\boldsymbol{M}_c$ is the mean of examples of class c, and $\boldsymbol{M}_0$ is the mean of examples of all the classes. The energy function in Eq. (3.13) is used to define the discriminative prior over the manifold as

$$p(\boldsymbol{X}) = \frac{1}{Z_q} \exp\left\{ -\frac{1}{\sigma_q^2} J^{-1} \right\}, \tag{3.16}$$

where Z_q is a normalization constant, and σ_q represents a global scaling of the prior. Then, the discriminative GPLVM (D-GPLVM) [183] is obtained by replacing the Gaussian prior in Eq. (3.10) with the prior in Eq. (3.16).

A more general prior based on the notion of the graph Laplacian matrix [29] has been used to derive a discriminative GPLVM model named Gaussian process latent random field (GPLRF) [212]. To define the prior, an undirected graph $\mathcal{G}$ is first constructed and the Laplacian matrix $\boldsymbol{L}$ associated with the graph $\mathcal{G}$ is learned. More details regarding the construction of the graph and the Laplacian matrix are given in Chapter 4. Once the Laplacian matrix $\boldsymbol{L}$ has been defined, the discriminative prior is given by

$$p(\boldsymbol{X}) = \frac{1}{Z_q} \exp\left[-\frac{\beta}{2} \text{tr}(\boldsymbol{X}^T \boldsymbol{L} \boldsymbol{X}) \right], \tag{3.17}$$

where Z_q is a normalization constant and $\beta > 0$ is a scaling parameter. The term $tr(\boldsymbol{X}^T \boldsymbol{L} \boldsymbol{X})$ in the discriminative prior in Eq. (3.17) reflects the sum of the distances between the latent positions of the examples from the same class. Thus, the latent positions from the same class that are closer will be given higher probability. This prior can be seen as a more general version of the LDA prior in Eq. (3.16), without the restriction on the size of the manifold.

Back-constraints With the standard GPLVM we can robustly unravel a low-dimensional manifold, even from small datasets, as long as the selected dimensionality of the latent space is much smaller than that of the observed data. In the case that this condition does not hold, GPLVM can suffer from overfitting, and eventually recover 'weird' latent representations. To address this problem and preserve the topological structure of the data, the authors in [106] proposed to back-constrain the GPLVM, by enforcing the latent positions to be a smooth function of the data space. This ensures that points that are close in the data space are also close on the manifold. More importantly, these constraints allow us to learn the inverse mappings, which are used during the inference step to map the query points from the data space onto the manifold. Specifically, each latent position $\boldsymbol{x}_i$ can be back-constrained so that it satisfies

$$x_{ij} = g_j(\boldsymbol{y}_i; \boldsymbol{A}_j) = \sum_{n=1}^{N} a_{nj} k_{bc}(\boldsymbol{y}_i, \boldsymbol{y}_n), \tag{3.18}$$

where x_{ij} is the j-th dimension of $\boldsymbol{x}_i \in \mathbb{R}^q$, g_j is the kernel ridge regression over $\boldsymbol{Y}$, and $\boldsymbol{A}$ is the matrix that holds the parameters for the regression. To obtain a smooth inverse mapping in the back-constraints, the RBF kernel can be employed again so that

$$k_{bc}(\boldsymbol{y}_i, \boldsymbol{y}_n) = \exp(-\frac{\gamma}{2}\|\boldsymbol{y}_i - \boldsymbol{y}_n\|^2), \tag{3.19}$$

where γ is the inverse width parameter. We can now re-parameterize the GPLVM by substituting the actual latent positions with the mapping from Eq. (3.18), and minimize Eq. (3.11) w.r.t. $\boldsymbol{A}$, as well as the hyper-parameters $\boldsymbol{\theta}$.[2] Hence, in back-constrained GPLVM we indirectly obtain the latent space via an efficient mapping which not only preserves the topology of the observed data, but also acts as a fast inference mechanism for future projections to the manifold.

3.4 Building on top of Gaussian Processes

In the following chapters we use the introductory material presented above as a basis for extending existing GP models, and to propose novel methodologies, applicable to the task of facial expression analysis. Specifically, in Chapter 4 we generalize discriminative flavors of GPLVMs to the multi-view scenario, by means of shared GPs [167]. In Chapter 5 we combine the shared GPs with the logistic regression, to introduce a joint generative and discriminative latent variable model. In Chapter 6 we propose a fully probabilistic auto-encoder based on

[2] Note that the inverse width of the back-constrained kernel γ is commonly obtained via a costly grid search cross-validation procedure.

GPs. Finally in Chapter 7 we explore the conditional property of the Gaussian distribution to introduce a domain adaptation framework with domain-specific GP experts.

Chapter 4

Gaussian Processes for Multi-view and View-invariant Facial Expression Recognition

Contents

Images of facial expressions are often captured from various *views* as a result of either head movements or variable camera position. Existing methods for multi-view and/or view-invariant facial expression recognition typically perform classification of the observed expression by using either classifiers learned *separately* for each view or a single classifier learned for all views. However, these approaches ignore the fact that different views of a facial expression are just different manifestations of the same facial expression. By accounting for this redundancy in information, we can design more effective classifiers for the target task.

4.1 Introduction

To exploit the relations among images of facial expressions captured from various views, in this chapter we introduce the discriminative shared Gaussian process latent variable model

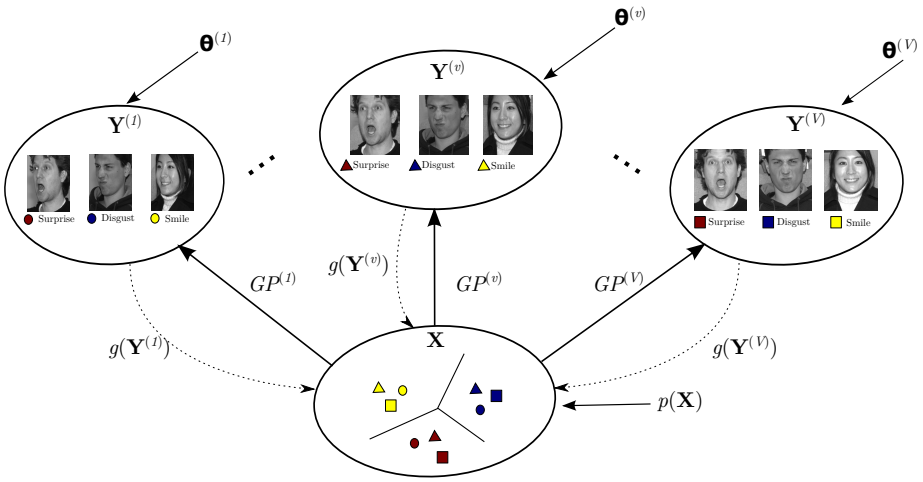


Figure 4.1: The overview of the proposed DS-GPLVM. The discriminative shared manifold $\boldsymbol{X}$ of facial expressions captured at different views ($\boldsymbol{Y}^{(v)}$, $v = 1 \dots V$) is learned using the framework of shared GPs (GP$^{(v)}$). The class separation in the shared manifold is enforced by the discriminative shared prior $p(\boldsymbol{X})$, informed by the data labels. During inference, the facial images from different views are projected onto the shared manifold by using the kernel-based regression, learned for each view separately ($g(\boldsymbol{Y}^{(v)})$) for a view-invariant approach, or simultaneously from multiple views for a multi-view approach. The classification of the query image is then performed using the kNN classifier.

(DS-GPLVM) for multi-view and view-invariant facial expression recognition of basic emotions. We adopt the multi-view learning strategy in order to represent the multi-view facial expression data on a common expression manifold. To facilitate this we assume the existence of instance constraints, *i.e.*, each image should be captured from different views. Toward this approach, we use the notion of shared GPs [167, 50], the generative framework for discovering a non-linear subspace shared across different observation spaces (*e.g.*, the facial views or feature representations). Since our ultimate goal is the expression classification, we place a discriminative prior, informed by the expression labels, over the manifold. The classification of an observed expression is then performed in the learned manifold using the kNN classifier. The proposed model can be regarded as a generalization of discriminative GP latent variable models [183, 212] for non-linear dimensionality reduction and classification of data from a single observation space. The learning of DS-GPLVM is carried out using the expression data from multiple views (corresponding images between the views). Classification of an observed facial expression, however, can be carried out either in a view-invariant manner (in case only a single view of the observed expression is available at runtime) or in a multi-view manner (in case multiple views of the observed expression are available at runtime). Note that during the testing phase, it is assumed that the view from which the image is captured is known. The proposed model can also perform fusion of different facial features (same view angle, but multiple image descriptors), in order to improve view-invariant facial expression classification.

In order to keep the model computationally tractable in the presence of large number of views, we propose a learning algorithm that splits the learning into different sub-problems (for each view), and then employs the alternating direction method of multipliers (ADMM) [18] to optimize each sub-problem separately. The outline of the proposed approach is given in Fig. 4.1. Note that the contents of this chapter are published in [59, 60, 61].

4.2 Discriminative Shared GPLVM

In this section we introduce the discriminative shared GPLVM (DS-GPLVM) for multi-view and view-invariant facial expression classification. We start by placing the DS-GPLVM within the framework of shared GPs [167]. We then define an appropriate discriminative prior for the shared space and introduce back-constraints from multiple observation spaces to the manifold. Finally we describe learning and inference in the proposed DS-GPLVM.

4.2.1 Discriminative Shared GPLVM: Model Definition

The proposed DS-GPLVM uses the notion of shared GPs [167] to learn latent variables $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^{N}$ shared among V observation spaces $\boldsymbol{Y} = \{\boldsymbol{Y}^{(1)}, \ldots, \boldsymbol{Y}^{(V)}\}$, with $\boldsymbol{Y}^{(v)} = \{\boldsymbol{y}_i^{(v)}\}_{i=1}^{N}$ denoting the observed input features from space v, $\boldsymbol{x}_i \in \mathbb{R}^q$ and $\boldsymbol{y}_i^{(v)} \in \mathbb{R}^D$, with $q \ll D$. Within this setting, we assume that each observation space is generated from the shared manifold via a separate GP. Note that a GP for each view is defined by a view-specific covariance matrix computed from the latent variables $\boldsymbol{X}$ that are shared among all the views. Formally, the marginal likelihood of the shared GPLVM is factorized as follows

$$p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{\theta}_s) = p(\boldsymbol{Y}^{(1)}|\boldsymbol{X}, \boldsymbol{\theta}^{(1)}) \ldots p(\boldsymbol{Y}^{(V)}|\boldsymbol{X}, \boldsymbol{\theta}^{(V)}), \tag{4.1}$$

where $\boldsymbol{\theta}_s = \{\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(V)}\}$ are the kernel parameters for each observation space. The kernel function is commonly selected to be the combination of the RBF, bias and noise terms

$$k^{(v)}(\boldsymbol{x}, \boldsymbol{x}') = \theta_1^{(v)} \exp(-\frac{\theta_2^{(v)}}{2}\|\boldsymbol{x} - \boldsymbol{x}'\|^2) + \theta_3^{(v)} + \frac{\delta_{\boldsymbol{x},\boldsymbol{x}'}}{\theta_4^{(v)}}, \tag{4.2}$$

where $\delta_{\boldsymbol{x},\boldsymbol{x}'}$ is the Kronecker delta function, and $\boldsymbol{\theta}^{(v)} = \{\theta_1^{(v)}, \theta_2^{(v)}, \theta_3^{(v)}, \theta_4^{(v)}\}$ are the kernel hyperparameters, associated with the view v.[1]

[1] With such kernels, by enforcing $\boldsymbol{\theta}^{(v)}$ to have small values, we can model (i) small output scales (*i.e.*, $\theta_1^{(v)}, \theta_3^{(v)}$), with (ii) large RBF support (*i.e.*, small $\theta_2^{(v)}$), and (iii) large noise variances (*i.e.*, small $\theta_4^{(v)}$).

The shared latent space $\boldsymbol{X}$ is then found by minimizing the negative log marginal likelihood penalized with the prior placed over the shared manifold, and is given by

$$L_s = \sum_v L^{(v)} - log(p(\boldsymbol{X})) \tag{4.3}$$

where $L^{(v)}$ is the negative log marginal likelihood from view v, and is given by

$$L^{(v)} = \frac{D}{2} \ln |\boldsymbol{K}^{(v)} + \sigma_v^2 \boldsymbol{I}| + \frac{1}{2} \text{tr}[(\boldsymbol{K}^{(v)} + \sigma_v^2 \boldsymbol{I})^{-1} \boldsymbol{Y}^{(v)} \boldsymbol{Y}^{(v)^T}] + \frac{ND}{2} \ln 2\pi. \tag{4.4}$$

In Eq. (4.4), the spherical Gaussian prior is placed over the manifold. To obtain a shared manifold for multi-view classification, in the following we define a discriminative shared-space prior.

4.2.2 Discriminative Shared GPLVM: Shared-space Prior

To define a discriminative shared space prior for multi-view learning, we adopt the modeling approach of discriminative GPLVMs for a single observation space proposed in [183, 212]. Specifically, in the discriminative GPLVM (D-GPLVM) [183], the authors define a prior based on linear discriminant analysis (LDA), which tries to maximize between-class separability and minimize within-class variability in the latent space. Such a prior, however, constrains the dimensionality of the latent space to be at most $C - 1$, where C is the total number of classes. On the other hand, in the GP latent random field (GPLRF) [212], the authors define a more general prior using the notion of the graph Laplacian matrix [29]. We follow the latter approach in our definition of the shared-space prior, as it allows for recovering of more flexible latent representations.

To define a prior based on a graph Laplacian matrix, we first need to construct an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{V_1, V_2, \ldots, V_N\}$ is the node set, with node V_i corresponding to a training example $\boldsymbol{x}_i$, and $\mathcal{E} = \{(V_i, V_j)_{i,j=1\ldots N} | i \neq j, \boldsymbol{x}_i \text{ and } \boldsymbol{x}_j \text{ belong to the same class}\}$ is the edge set. Since we have paired each node with the random variable $\boldsymbol{x}_i$ we have obtained a Gaussian Markov Random Field (GMRF) [152] w.r.t. the graph $\mathcal{G}$. Next, each edge in the graph needs to be associated with a weight. To design a shared-space prior we construct view-specific weight matrices $\boldsymbol{W}^{(v)}$, $v = 1, \ldots, V$. Specifically, the elements of the weight matrices are obtained by applying the RBF kernel to the data from each view as

$$\mathbf{W}_{ij}^{(v)} = \begin{cases} \exp\left(-\frac{\|\mathbf{y}_i^{(v)} - \mathbf{y}_j^{(v)}\|^2}{t^{(v)}}\right) & \text{if } i \neq j \text{ and } c_i = c_j, \\ 0 \quad \text{otherwise.} \end{cases} \tag{4.5}$$

where c_i is the class label, and $t^{(v)}$ is the kernel width which is set to the mean squared distance between the training inputs as in [155]. Then, the graph Laplacian for view v is $\boldsymbol{L}^{(v)} = \boldsymbol{D}^{(v)} - \boldsymbol{W}^{(v)}$, where $\boldsymbol{D}^{(v)}$ is a diagonal matrix with $\boldsymbol{D}^{(v)}_{ii} = \sum_j \boldsymbol{W}^{(v)}_{ij}$. Because the graph Laplacians from different views vary in their scale, we use the normalized graph Laplacian, defined as

$$\boldsymbol{L}_N^{(v)} = (\boldsymbol{D}^{(v)})^{-1/2} \boldsymbol{L}^{(v)} (\boldsymbol{D}^{(v)})^{-1/2}, \tag{4.6}$$

Subsequently, we define the (regularized) joint Laplacian as

$$\tilde{\boldsymbol{L}} = \boldsymbol{L}_N^{(1)} + \boldsymbol{L}_N^{(2)} + \ldots + \boldsymbol{L}_N^{(V)} + \xi \boldsymbol{I} = \sum_v \boldsymbol{L}_N^{(v)} + \xi \boldsymbol{I}, \tag{4.7}$$

with $\boldsymbol{I}$ the identity matrix, and ξ a regularization parameter (typically set to a small value *e.g.*, 10^{-4}), which ensures that $\tilde{\boldsymbol{L}}$ is positive-definite [214]. This, in turn, allows us to define the discriminative shared-space prior as

$$p(\boldsymbol{X}) = \prod_{v=1}^{V} p(\boldsymbol{X}|\boldsymbol{Y}^{(v)})^{\frac{1}{V}} = \frac{1}{V \cdot Z_q} \exp\left[-\frac{\beta}{2} \mathrm{tr}(\boldsymbol{X}^T \tilde{\boldsymbol{L}} \boldsymbol{X})\right]. \tag{4.8}$$

Here, Z_q is a normalization constant and $\beta > 0$ is a scaling parameter. The discriminative shared-space prior in (4.8) aims at maximizing the class separation in the manifold learned from data from all the views, and it can be regarded as a multi-view kernel extension of the priors defined for a single view in [183, 212]. By incorporating this prior in Eq. (4.3) we obtain the final form of the negative log marginal likelihood of the proposed DS-GPLVM

$$L_s = \sum_v L^{(v)} + \frac{\beta}{2} \mathrm{tr}(\boldsymbol{X}^T \tilde{\boldsymbol{L}} \boldsymbol{X}), \tag{4.9}$$

where $L^{(v)}$ is defined by Eq. (4.4).

4.2.3 Discriminative Shared GPLVM: Back-constraints

As we have seen in Section 3.3.2, in order to assure that the topology of the observed space is preserved on the manifold we need to back-constrain the GPLVM. In DS-GPLVM, this is achieved by the discriminative shared-space prior, since the weight matrix used to define the prior is built from the observed data. However, to perform fast inference with DS-GPLVM we still need to learn the inverse mappings that project data from different views onto the shared manifold. For this, we consider two scenarios. In the first, we define v sets of constraints (one for each view), which are enforced by separate inverse mappings from each view to the shared space. In the second, we define one set of constraints (for all the views), which are enforced by a single inverse mapping from all the views to the shared space. We refer to the former

as independent back-projections (IBP), and the latter as single back-projection (SBP). These are given by

- **IBP** from each view $v = 1, \ldots, V$

$$\boldsymbol{X} = g(\boldsymbol{Y}^{(v)}, \boldsymbol{A}^{(v)}) = \boldsymbol{K}_{bc}^{(v)} \boldsymbol{A}^{(v)}. \tag{4.10}$$

- **SBP** from V views

$$\boldsymbol{X} = g(\boldsymbol{Y}, \boldsymbol{A}) = \left(\sum_{v=1}^{V} w_v \boldsymbol{K}_{bc}^{(v)} \right) \boldsymbol{A} = \tilde{\boldsymbol{K}} \boldsymbol{A}, \tag{4.11}$$

where $g(\cdot, \cdot)$ represents the mapping function(s) learned using the kernel ridge regression. w_v is the (scalar) weight for view v, while the elements of $\boldsymbol{K}_{bc}^{(v)}$ are given by Eq. (3.19), which for convenience we re-introduce here as well

$$k_{bc}^{(v)}(\boldsymbol{y}_i^{(v)}, \boldsymbol{y}_n^{(v)}) = \exp(-\frac{\gamma^{(v)}}{2} \|\boldsymbol{y}_i^{(v)} - \boldsymbol{y}_n^{(v)}\|^2). \tag{4.12}$$

Note that for a single view, the model can be re-parametrized to obtain an unconstrained optimization problem (see Sec. 3.3.2). Yet, in the case of multiple views, this is not possible as it would result in different $\boldsymbol{X}$ for each view. Therefore, we need to solve a constrained optimization problem, the complexity of which increases with the number of views. To efficiently solve this, in the following section we propose an iterative learning algorithm for simultaneous learning of the shared space and inverse mappings in the proposed model.

4.2.4 Discriminative Shared GPLVM: Learning and Inference

Learning of the model parameters $\boldsymbol{X}, \theta_s$ and $\mathbf{A}$, consists of minimizing the negative log marginal likelihood given by Eq. (4.9) subject to either the IBP or SBP constraints. Formally, we aim to solve the following minimization problem:

$$\begin{aligned} &\arg\min_{\boldsymbol{X}, \theta_s, \boldsymbol{A}} L_s(\boldsymbol{X}) + R(g) \\ &s.t. \begin{cases} IBP(\boldsymbol{X}, \boldsymbol{A}^{(v)}) \triangleq \boldsymbol{X} - \boldsymbol{K}_{bc}^{(v)} \boldsymbol{A}^{(v)} = \mathbf{0} \ , v = 1, \ldots, V \\ SBP(\boldsymbol{X}, \boldsymbol{A}) \triangleq \boldsymbol{X} - \tilde{\boldsymbol{K}} \boldsymbol{A} = \mathbf{0} \ , \sum_{v=1}^{V} w_v = 1, w_v \geq 0, \end{cases} \end{aligned} \tag{4.13}$$

where $R(g)$ is a regularization term. To obtain the function form for $R(g)$, we first derive the solution of the kernel ridge regression from the mapping function of the infinite-dimensional feature space $g(\boldsymbol{x}_i) = \phi(\boldsymbol{x}_i)^T w$, as in [78]. The solution to this problem is of the form of

$w = \sum_{i=1}^{N} a_i \phi(\boldsymbol{x}_i)$. Hence, by applying the Representer Theorem [160] on this space, and by using the Tikhonov regularization for the parameters w, we arrive at the optimal functional form for $R(g)$ as

$$R(g) = \begin{cases} \sum \frac{\lambda^{(v)}}{2} r(g^{(v)}),\ r(g^{(v)}) = \text{tr}(\boldsymbol{A}^{(v)^T} \boldsymbol{K}_{bc}^{(v)} \boldsymbol{A}^{(v)}), & \text{for } IBP \\ \frac{\lambda}{2} \text{tr}(\boldsymbol{A}^T \tilde{\boldsymbol{K}} \boldsymbol{A}), & \text{for } SBP \end{cases} \tag{4.14}$$

IBP: Parameter Optimization. We first present the learning procedure for the more general case involving the IBP constraints, and then provide the solution for the SBP case. From Eq. (4.13), we see that the back-mapping from each view is represented by an independent set of linear constraints. We exploit this to find the model parameters by iteratively solving a set of sub-problems. We first incorporate the IBP constraints into the regularized negative log marginal likelihood in Eq. (4.13) by using the Lagrange multipliers. As a result, we obtain the following augmented Lagrangian function:

$$\begin{aligned} \mathcal{L}^{IBP}(\boldsymbol{X}, \{\boldsymbol{A}^{(v)}, \boldsymbol{\Lambda}^{(v)}\}_{v=1}^{V}) =& L_s(\boldsymbol{X}) + R(g) + \\ & \sum_{v=1}^{V} \langle \boldsymbol{\Lambda}^{(v)}, IBP(\boldsymbol{X}, \boldsymbol{A}^{(v)}) \rangle + \frac{\mu}{2} \sum_{v=1}^{V} \| IBP(\boldsymbol{X}, \boldsymbol{A}^{(v)}) \|_F^2, \end{aligned} \tag{4.15}$$

where $\boldsymbol{\Lambda}^{(v)}$ are the Lagrange multipliers for view v, $\langle \cdot, \cdot \rangle$ is the inner product, and $\mu > 0$ is the penalty parameter. We can see from Eq. (4.15) that the linear constraint has been incorporated into the cost function as a quadratic penalty term without affecting the solution to the problem. The role of the Lagrange multipliers (inner product term) is to achieve efficiency in obtaining the solution without the requirement of sequentially increasing the penalty parameter to infinity [18]. The standard approach is to minimize the objective in Eq. (4.15) w.r.t. all the model's parameters simultaneously. Yet, this is impractical, as the fact that the objective function is separable, is not exploited to simplify the problem. To remedy this, we employ the alternating direction method of multipliers (ADMM) [18] to decompose the minimization into subproblems, each of which can be solved separately w.r.t. to a subset of the model parameters. More specifically, we split the learning of the parameters of the shared space and the back-mappings from each view, by defining the iterations of ADMM as follows. We first solve for $\boldsymbol{X}$ and $\boldsymbol{\theta_s}$ as

$$\{\boldsymbol{X}, \boldsymbol{\theta}_s\}_{t+1} = \arg\min_{\boldsymbol{X}, \boldsymbol{\theta}_s} \ L_s(\boldsymbol{X}) + \frac{\mu_t}{2} \sum_{v=1}^{V} \| IBP(\boldsymbol{X}, \boldsymbol{A}_t^{(v)}) + \frac{\boldsymbol{\Lambda}_t^{(v)}}{\mu_t} \|_F^2. \tag{4.16}$$

Then, for each view $v = 1, ..., V$, we solve for $\boldsymbol{A}^{(v)}$ as

$$\boldsymbol{A}_{t+1}^{(v)} = \arg\min_{\boldsymbol{A}^{(v)}} \ r(\boldsymbol{A}^{(v)}) \ + \frac{\mu_t}{2} \| IBP(\boldsymbol{X}_{t+1}, \boldsymbol{A}^{(v)}) + \frac{\boldsymbol{\Lambda}_t^{(v)}}{\mu_t} \|_F^2, \tag{4.17}$$

and finally update the Lagrangian and the penalty parameter as

$$\boldsymbol{\Lambda}_{t+1}^{(v)} = \boldsymbol{\Lambda}_{t}^{(v)} + \mu_t IBP(\boldsymbol{X}_{t+1}, \boldsymbol{A}_{t+1}^{(v)}) \tag{4.18}$$

$$\mu_{t+1} = \min(\mu_{max}, \rho\mu_t), \tag{4.19}$$

respectively. Note that in Eq. (4.19), ρ is kept constant (it is typically set to $\rho = 1.1$).

Since there is not a closed-form solution for the problem in Eq. (4.16), we use the conjugate gradient optimization algorithm[2] to minimize the objective w.r.t. the latent positions $\boldsymbol{X}$ and the kernel parameters $\boldsymbol{\theta}_s$[3]. On the other hand, the problem in Eq. (4.17) is similar to that of kernel ridge regression, and it has a closed-form solution, which is given by

$$\boldsymbol{A}^{(v)} = \left(\boldsymbol{K}_{bc}^{(v)} + \frac{\lambda^{(v)}}{\mu_t}\boldsymbol{I}\right)^{-1}\left(\boldsymbol{X} + \frac{\boldsymbol{\Lambda}_t^{(v)}}{\mu_t}\right) \tag{4.20}$$

However, this solution depends on the parameters $\gamma^{(v)}$ (*i.e.*, the inverse width of the back-projection kernel from Eq. (4.12)), and $\lambda^{(v)}$ (*i.e.*, the regularization weight associated with IBP/SBP), which are normally tuned through costly cross-validation procedures. To alleviate this, we reformulate the optimization problem in Eq. (4.17). For this, we use the notion of the leave-one-out (LOO) cross-validation procedure for the kernel ridge regression [174] to define the learning of the parameters $\gamma^{(v)}$ and $\lambda^{(v)}$. Once estimated, these parameters are used to compute $\boldsymbol{A}^{(v)}$. Note that by employing the LOO optimization scheme we reduces the chances of overfitting.

The idea of the LOO learning procedure is based on the fact that given any training set and the corresponding learned regression model, if we add a sample to the training set with the target equal to the output predicted by the model, the latter will not change since the cost function will not increase [174]. Thus, given the training set with the sample $\boldsymbol{y}_i^{(v)}$ left out, the predicted outputs $\hat{\boldsymbol{X}}^{(\setminus i)}$ (the superscript denotes that the i-th sample is left out) will not change if the sample $\boldsymbol{y}_i^{(v)}$ with target $\hat{\boldsymbol{x}}_i^{(\setminus i)}$ is added to the set. Then, the goal of LOO is to minimize the difference between the predictions $\hat{\boldsymbol{x}}_i^{(\setminus i)}$ and the actual outputs $\boldsymbol{x}_i$ for all the samples. To compute this, we first need to define the matrix

$$\boldsymbol{M} \triangleq \begin{bmatrix} m_{ii} & \boldsymbol{m}_i^T \\ \boldsymbol{m}_i & \boldsymbol{M}_i \end{bmatrix} = (\boldsymbol{K}_{bc}^{(v)} + \frac{\lambda^{(v)}}{\mu_t}\boldsymbol{I}), \tag{4.21}$$

where we partitioned the inverse matrix from Eq. (4.20) so that the elements corresponding to the i-th sample appear only in the first row and column of $\boldsymbol{M}$ (the same is done for $\boldsymbol{X}$

[2] We used Rasmussen's *minimize.m* function provided from http://learning.eng.cam.ac.uk/carl/code/minimize/.
[3] The derivatives of the objective w.r.t. the model parameters are given in the appendix.

and $\boldsymbol{\Lambda}_t^{(v)}$ in order to place the i-th row on the top). Furthermore, $\boldsymbol{M}_i$ is the kernel matrix formed from the remaining elements as $\boldsymbol{M}_i = (\boldsymbol{K}_{bc\setminus i}^{(v)} + \frac{\lambda^{(v)}}{\mu_t}\boldsymbol{I}_{N-1})$. Then, using Eq. (4.20), the prediction and the actual target for sample i are given by

$$\hat{\boldsymbol{x}}_i^{(\setminus i)} = \boldsymbol{m}_i^T \boldsymbol{M}_i^{-1} \boldsymbol{m}_i \boldsymbol{A}_i^{(v)} + \boldsymbol{m}_i^T \boldsymbol{A}_{\setminus i}^{(v)} \tag{4.22}$$

$$\boldsymbol{x}_i = m_{ii} \boldsymbol{A}_i^{(v)} + \boldsymbol{m}_i^T \boldsymbol{A}_{\setminus i}^{(v)} - \boldsymbol{\Lambda}_i^{(v)}/\mu_t. \tag{4.23}$$

We can now define the cost for the LOO procedure, which is

$$E_{LOO} = \frac{1}{2}\sum_{i=1}^{N} \|\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i^{(-i)}\|^2 = \frac{1}{2}\sum_{i=1}^{N} \|\frac{\boldsymbol{A}_i^{(v)}}{[\boldsymbol{M}^{-1}]_{ii}} - \frac{\boldsymbol{\Lambda}_i^{(v)}}{\mu_t}\|^2 \tag{4.24}$$

Minimization of E_{LOO} w.r.t. $\gamma^{(v)}$ and $\lambda^{(v)}$ is accomplished using again the conjugate gradient algorithm.[4] By plugging these parameters into Eq. (4.20), we obtain $\boldsymbol{A}^{(v)}$. Note that by adopting the LOO learning approach, we: (i) avoid the burden of the standard cross-validation procedures, which are time consuming, and (ii) reduce the chances of overfitting the model parameters by using the additional cost defined in Eq. (4.24).

At this point, it is important to clarify that under the proposed ADMM-based optimization scheme we are able to automatically learn the majority of the model's parameters (*i.e.*, $\boldsymbol{X}$ $\boldsymbol{\theta}_s$, μ, λ, γ), avoiding the need of their tuning via validation procedures. The only parameter learned by means of cross-validation is the weight of the prior, β, while we also need to explore the effect of the dimensionality, q, of the manifold.

SBP: Parameter Optimization. Analogous to the IBP case, we define the Augmented Lagrangian function for the SBP case using the regularized negative log-likelihood and the SBP constraints from Eq. (4.13). The resulting function has the form as in Eq. (4.15), but after dropping the dependencies on v, and replacing the IBP by SBP constraints. The model parameters are then found by applying the proposed ADMM to the augmented Lagrangian function. For this, the objectives in each iteration of the ADMM for the IBP case described above are adjusted accordingly.

To achieve efficiency, when applying the conjugate gradient algorithm in each iteration of the ADMM, with either IBP or SBP constraints, we stop at the first line search, update the corresponding parameters, and go to the next iteration. The ADMM cycle is repeated until convergence of the augmented Lagrangian function.

[4]The exact derivation of Eq. (4.22)-(4.23) along with the gradients of Eq. (4.24) w.r.t. $\gamma^{(v)}$ and $\lambda^{(v)}$ are given in the appendix.

Algorithm 1 DS-GPLVM: Learning and Inference

Learning
Inputs: $\mathcal{D} = (\boldsymbol{Y}^{(v)}, \boldsymbol{c}), v = 1, \ldots, V$
Initialize $\mu_{max} >> \mu_0 > 0$, $\rho = const.$, $\boldsymbol{X}_0$, $\boldsymbol{A}_0^{(v)}$, $\boldsymbol{\Lambda}_0^{(v)}$.
repeat
Step 1: Update $(\boldsymbol{X}, \boldsymbol{\theta}_s)$ by minimizing Eq. (4.16).
Step 2: Minimize E_{LOO} from Eq. (4.24) w.r.t $(\gamma^{(v)},\lambda^{(v)})_{v=1,\ldots,V}$ for IBP, and (γ,λ) for SBP.
Step 3: Update $(\boldsymbol{\Lambda}^{(v)}, \mu, \boldsymbol{A}^{(v)})$ for IBP, and $(\boldsymbol{\Lambda}, \mu, \boldsymbol{A})$ for SBP, from Eq. (4.18)–(4.20).
until convergence of Eq. (4.15)
Outputs: $\boldsymbol{X}$, $\boldsymbol{A}$

Inference
Inputs: $\boldsymbol{y}_*^{(v)}$ for IBP, and $[\boldsymbol{y}_*^{(1)}, ..., \boldsymbol{y}_*^{(V)}]$ for SBP, k for classification.
Step 1: Find the projection $\boldsymbol{x}_*$ to the latent space using Eq. (4.10) for IBP, and Eq. (4.11) for SBP.
Step 2: Apply kNN classifier to the latent space to obtain the class prediction: $c_* = k\text{NN}(\boldsymbol{x}_*, \boldsymbol{X})$.
Output: c_*

Inference in the DS-GPLVM is straightforward. The test data $\boldsymbol{y}_*$ (which for the view-invariant case come from a single view v, and for the mutli-view case from all available views) are first projected to the shared space using the back-mappings defined by Eq. (4.10) for the IBP, or Eq. (4.11) for the SBP case. In the second step, classification of the target facial expression is accomplished by using a single classifier trained on the discriminative shared manifold. For this, we use the kNN classifier[5]. In Algorithm 1 we summarize the learning and inference of the proposed DS-GPLVM.

4.3 Relation to Prior Work on Multi-view Learning

In what follows, we make a short overview of the most popular multi-view learning methods that can be applied to the multi-view facial expression analysis. A common approach in multi-view classification is to learn the view-specific projection using paired samples from different views, and to project those samples onto a common latent space, followed by their classification. The paired samples usually refer to samples that come from the same subject (*e.g.*, face images of a person in two different views). The goal here is to learn a latent space

[5]In the model as defined, the resulting posterior is the manifold and not the class information, so it cannot be used for the classification. For this reason, we need to apply a classifier to the inputs projected onto this manifold during inference. A reasonable choice would be to opt for the GP classifier, however, in our case this would be impractical for two reasons: (i) in the case of more than two classes, the computation complexity of GP classification increases significantly since we have to learn a different kernel for each class, making it less applicable to the large number of classes/views. (ii) More importantly, since we are not interested in the classification uncertainty, the GP classification is expected to perform similarly to the standard kernel regression, as noted in [146]. Thus, we opt for the deterministic kNN classifier which is the commonly employed classifier in the GPLVM discriminative models (*e.g.*, see GPLRF [212]).

where the paired samples are placed close if they come from the same class/subject, and far apart otherwise.

A widely used unsupervised approach to learn such latent spaces is canonical correlation analysis (CCA) [84] and its non-linear variant kernel CCA (KCCA) [79]. The goal of these methods is to find projection to a common subspace where the correlation between the low-dimensional embeddings is maximized. These methods can handle data only in the pair-wise manner (thus, only two views at a time), which makes them inappropriate for multi-view classification problems with more than two views. A generalization of CCA to the multi-view setting, multiview CCA (MCCA), has been proposed in [153]. The main idea of MCCA is to find a common subspace where the correlation between the low-dimensional embeddings of any two views is maximized. Apart from CCA-based methods, there are a few works that extend the single-view subspace learning to the multi-view case. [103] is a representative of this approach. It is a spectral clustering approach for the multi-view setting. In particular, the spectral embedding from one view is used to constrain the data of the other view. Note that the methods mentioned above are proposed for unsupervised learning. Thus, in the context of the multi-view facial expression analysis, they are not expected to perform well as the view alignment by these methods is not optimized for classification.

Another group of methods performs supervised multi-view analysis. For instance, multi-view Fisher discriminant analysis (MFDA) [45] learns classifiers in different views, by maximizing the agreement between the predicted labels of these classifiers. However, MFDA can only be used for binary problems. In [95], the authors extended the LDA to the multiview case, named multi-view discriminant analysis (MvDA). This model maximizes the between-class and minimizes the within-class variations, across all the views, in the common subspace. Generalized multi-view analysis (GMA) [165] has also been proposed for extending dimensionality reduction techniques for single views to multiple views. An instance of GMA, the generalized multi-view LDA (GMLDA), finds a set of projections in each view that attempt to separate the content of different classes and unite different views of the same class in a common subspace. Another example of GMA is the generalized multi-view locality preserving projections (GMLPP), that extends the LPP [130] model, which can be used to find a discriminative data manifold using the labels. Although effective in some tasks, these models are all based on linear projection functions. This can limit their performance when dealing with high-dimensional input features (*i.e.*, appearance based facial features), as well as their ability to successfully unravel non-linear manifold(s) of multiple views. The above limitations have been addressed in the proposed DS-GPLVM model.

Figure 4.2: Example images from MultiPIE (top), LFPW (middle) and SFEW (bottom) datasets with the facial point annotations for the first two.

4.4 Experiments

Herein, we empirically assess the multi-view learning abilities of the proposed DS-GPLVM on the tasks of facial expression classification of basic emotions and smile detection.

4.4.1 Experimental Protocol

Datasets. We evaluate the performance of the proposed DS-GPLVM on expressive face images from three publicly available datasets: MultiPIE [76], labeled face parts in the wild (LFPW) [17] and static facial expressions in the wild (SFEW) [44]. Fig. 4.2 shows sample images from these datasets. From the MultiPIE dataset we used images of 270 subjects depicting acted 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 1531 images per pose. For all images, we selected the flash from the view of the corresponding camera in order to have the same illumination conditions. The LFPW dataset contains images downloaded from google.com, flickr.com, and yahoo.com, depicting spontaneous facial expressions (mainly smiles), in large variation of poses, illumination and occlusion. We used 200 images of NE and SM expressions from the test set provided by [17]. We manually annotated the images in terms of the poses used in MultiPIE. Lastly, the SFEW dataset consists of 700 images of 95 subjects, extracted from movies containing facial expressions with various head poses, occlusions and illumination conditions. The images have been labeled in terms of six basic emotion expressions, *i.e.*, Anger (AN), Disgust (DI), Fear (FE), Happiness (HA), Sadness (SA), Surprise (SU), as well as Neutral (NE).

Features. The images from both MultiPIE and LFPW were cropped so as to have equal size (140 × 150 pixels), and annotations of the locations of 68 facial landmark points were provided by [154], which were used to align the facial images in each pose using an affine transform. Similarly, the images from SFEW were cropped (112 × 164 pixels) and aligned using 5 facial landmark points (center of the eyes, tip of the nose, and corners of the mouth) provided by [44]. For the experiments on MultiPIE, we used three sets of features: (I) facial points, (II) LBPs [131], and (III) DCT [4]. More specifically, from each aligned facial image we extracted LBPs and DCT features from local patches of size 15 × 15 around the facial landmarks. For LBPs, we used 8 neighbors with radius 2, and in the case of DCT we kept the first 15 coefficients (zig-zag method) of each patch. We then concatenated all the patches to form the feature vectors. Note that LBP and DCT are complementary features, since the former captures local information between a neighborhood of pixels, while the latter preserves the spatial correlation of the pixels inside the neighborhood. Finally, we applied PCA on the three feature sets, keeping 95% of the total energy, to remove unwanted noise and artifacts, and reduce the dimensionality of the original feature vectors (especially the appearance based). The resulting dimensionality of each set varies among the views. The dimensionality of feature set (I) is around 20D, while for feature sets (II)&(III) we obtain 100D feature vectors. In the experiments conducted on LFPW, we used only feature set (I), while for SFEW we extracted the same local texture descriptors as in [44], *i.e.*, local phase quantization (LPQ) [132] and pyramid of HOG (PHOG) [22]. To reduce the dimensionality, we applied again PCA by keeping the same amount of energy, *i.e.*, 95%, resulted in 47D and 220D feature vectors, respectively.

Models Compared. We compare the DS-GPLVM to the state-of-the-art view-invariant and multi-view learning methods. As the baseline method, we use the 1-nearest neighbor (1-NN) classifier trained/tested in the original feature space. Similarly, we apply 1-NN classifier to the subspace obtained by LDA, supervised LPP [211], and their kernel counterparts, the D-GPLVM [183] with the LDA-based prior, and the GPLRF [212]. These are well-known methods for supervised dimensionality reduction, and we show their performance in the view-invariant version of the experiments. In the experiments conducted in the multi-view/feature fusion settings, we compare DS-GPLVM to the baseline methods: CCA [84] and KCCA [79]. Since they are designed to deal with only two modalities (feature sets), we follow the pair-wise (PW) evaluation approach, as in [95], *i.e.*, the methods are trained on all combinations of view pairs, and their results are averaged. We also compare DS-GPLVM to the state-of-the-art methods for multi-view learning, namely, the MvDA [95], and the multi-view extensions of LDA (GMLDA), and LPP (GMLPP), proposed in [165].

Evaluation Procedure. For the experiments in MultiPIE and LFPW we performed 5-fold subject independent cross-validation. We used a separate validation set to tune the parameters of each model. More specifically, for all the GPLVM-based methods (*i.e.*, DS-GPLVM, GPLRF and D-GPLVM) the optimal weight for the prior β was set using a grid search. For the GPLRF and D-GPLVM we performed additionally an extra grid search to tune the kernel's parameter of the mapping from the back-constrain (RBF kerenel was used) as in [183]. For the GMA-based methods (*i.e.*, GMLDA and GMLPP) we tuned the parameter that controls the alignment of the subspaces as suggested in [165]. Finally, in KCCA the width of the employed RBF kernel was cross-validated, while LPP, LDA and MvDA had no parameters to tune. For the experiments on SFEW we adopted the configuration proposed by the creators of the dataset in [44]. The data were already split into two folds, for training and testing. Each time the training fold was further split in 5 folds, to tune the parameters of the models with 5-fold subject independent cross-validation. For this experiment, due to the small size of the dataset, after tuning the parameters with the cross validation, each model was re-trained on the whole train and validation set (the one of the two original folds of the dataset) with the optimal parameters, before reporting the results on the test set. To report the accuracy of facial expression recognition, we use the classification rate, where the classification was performed on the test set using the 1-NN classifier in all the subspace-based models.

The conducted experiments are organized as follows. In Section 4.4.2, we evaluate the performance and the convergence of DS-GPVLM in terms of different parameter choices and settings, using the MultiPIE dataset. In Section 4.4.3, we evaluate the effectiveness of the proposed DS-GPLVM in the task of multi-view FER on MultiPIE. Specifically, we consider two settings: the standard *multi-view* setting, where images from all the views are available during training/inference, and *view-invariant* setting, where images from all the views are available during training but only a single view is available during inference. It is important to explicitly note the inherent limitation of all models on such multi-view setting, which has to do with the existence of instance constraints, *i.e.*, same image captured from different views, during the training phase. Moreover, we also evaluate the model on the feature fusion task, where different types of features extracted within the same view are used. In addition, we challenge the robustness of the model under different illumination, where we evaluate the performance of the model on images with different lighting conditions within the same view. In Section 4.4.5, we test the ability of the DS-GPLVM to generalize to spontaneously displayed facial expressions. For this, we perform the cross-dataset evaluation of the model, where images of SM and NE class from MultiPIE are used for training, and images of the corresponding classes from LFPW for testing. Finally, in Section 4.4.6, we evaluate DS-GPLVM on the

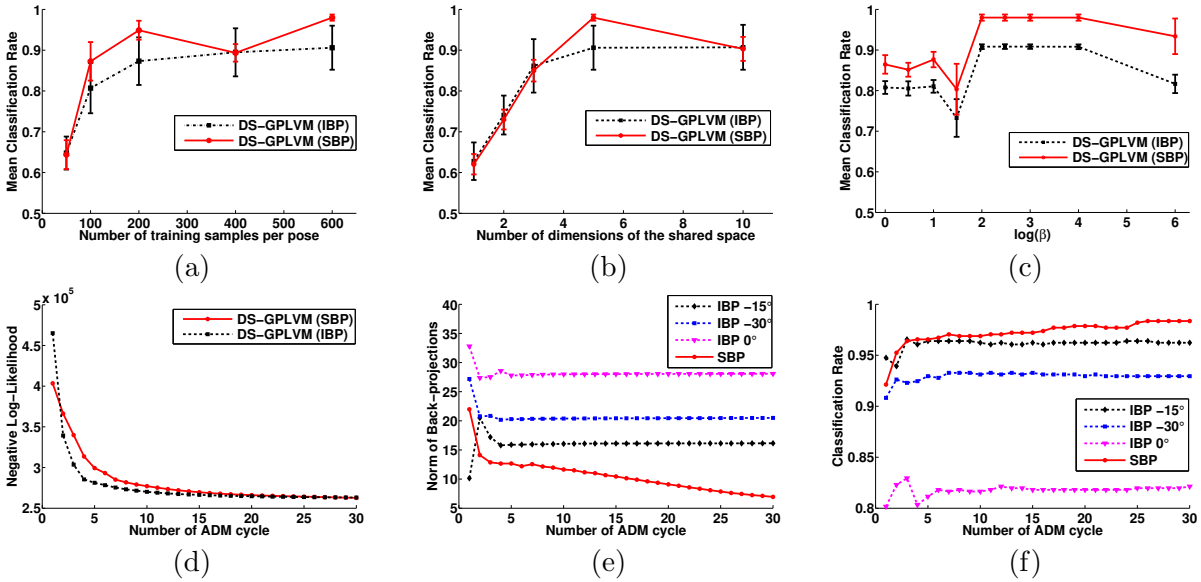


Figure 4.3: DS-GPLVM. Upper row shows mean classification rate across all 5 poses from the MultiPIE dataset using feature-set (I) as a function of: (a) the number of training data per pose, (b) the dimensionality of the latent space, and (c) the prior scale parameter β. Lower row depicts: (d) the negative Log-Likelihood, (e) the norms of the constraints in the DS-GPLVM, and (f) the mean classification rate, as a function of the number of the ADMM cycles.

feature fusion task using real-world images from the SFEW dataset.

4.4.2 DS-GPLVM: Theoretical Evaluation

In this section, we evaluate the performance of the proposed DS-GPLVM w.r.t. the various parameter values. For this, we use the feature set (I), *i.e.*, the facial points, extracted from the MultiPIE dataset. Fig. 4.3(a)–(c) show the average classification rate (across the views) of the DS-GPLVM for different number of training samples per view, the size of the shared-space, and parameter $\beta = \{1, 3, 10, 30, 100, 300, 1000, 10000\}$. Fig. 4.3(a) shows the performance of SBP and IBP versions of DS-GPLVM, the parameters of which are learned using a varying number of training data, while the manifold size is fixed to 5. We see that the SBP version of DS-GPLVM (*multi-view* setting) achieves a high classification rate ($\sim 87\%$) when using a relatively small number of training data (*i.e.*, 100 images per view). On the other hand, the IBP version of DS-GPLVM (*view-invariant* setting) requires more training data (~ 500 images per view) to achieve a similar performance. This is a consequence of not using the images from all available views during the inference step. However, with the increased number of training data, the model effectively learns the correlations among the views, rendering the information from some views redundant during the inference. In Fig. 4.3(b), we see how the size of the

shared space affects the accuracy of the learned model. It is clear that both SBP and IBP variants of the model find the 5-dimensional shared space optimal for classification. Lower dimensional manifolds fail to explain the correlations among the views, while manifolds with more than 5 dimensions do not include any additional discriminative information. Fig. 4.3(c) illustrates the influence of the shared space discriminative prior on the classification task. In the case of both SBP and IBP, $\beta = 300$ results in the best performance of the model, while its further increase leads to a drop in the performance. This is expected, as for high values of β the likelihood term in the DS-GPLVM is fully ignored, and hence, the model resembles the LPP. Evidently, such model is prone to overfitting mainly because of the strong influence of the labels during training. On the other hand, for small values of β the shared space is not sufficiently informed about the class labels, resulting again in a lower performance. In what follows, we set for both the SBP and IBP variants of the model the number of training examples to 500, the size of the shared space to 5, and $\beta = 300$.

Fig. 4.3(d)–(f) illustrate the convergence properties of the DS-GPLVM. We see from Fig. 4.3(d) that the regularized negative log-likelihood of the model reaches a local minimum in less than 25 cycles of the ADMM. Fig. 4.3(e) shows the Frobenius norm [19] of the constraints for the SBP and IBP variants, *i.e.*, the difference between the estimated shared space and the back-mappings. Note that the DS-GPLVM is always initialized in the $-15°$ view (it is found to be the most informative view). Hence, we can see that the norm of this view (black curve) starts from a low value when IBP is used. However, with more cycles of the ADMM, the DS-GPLVM learns the shared manifold by taking into account all views, and thus, the error of back projections from the remaining views to the shared subspace decreases, while the one from the initialized view, *i.e.*, the $-15°$, increases slightly – the consequence of the model trying to align the manifolds of different views. The red curve represents the error between the learned subspace and the back projections in the case of SBP. It is clear that the SBP variant outperforms the IBP variant of the model, since the former achieves a closer back-projection to the shared discriminative manifold, resulting in a better classification performance. This comes with a larger number of the ADMM cycles during learning of the DS-GPLVM with SBP, since it uses all views simultaneously to learn the back-mapping. Finally, from Figs. 4.3(e)–(f), we observe strong correlation between the norms of the model variants and the classification rate. In all cases, the increased classification performance is achieved by decreasing the gap between the shared-space and back-mappings, with both measures converging synchronously.

Table 4.1: Average classification rate across five views from the MultiPIE dataset for three feature sets. IBP version of DS-GPLVM was trained using all available views, and tested per view. The reported standard deviation is across five views.

Methods	Features		
	I	II	III
kNN	76.15 ± 5.42	81.71 ± 2.86	71.80 ± 2.23
LDA	87.72 ± 6.67	86.24 ± 2.31	87.02 ± 2.59
LPP	87.81 ± 6.65	86.16 ± 2.16	86.82 ± 2.60
D-GPLVM	87.17 ± 5.80	85.92 ± 2.95	86.87 ± 3.15
GPLRF	86.93 ± 6.30	85.58 ± 2.66	86.88 ± 2.91
GMLDA	86.72 ± 6.57	85.18 ± 2.94	86.40 ± 3.40
GMLPP	87.74 ± 6.12	86.10 ± 2.13	86.21 ± 2.06
MvDA	87.84 ± 6.51	86.66 ± 2.84	86.79 ± 2.86
DS-GPLVM	**90.60 ± 5.40**	**88.44 ± 2.84**	**89.18 ± 2.83**

4.4.3 Comparisons with other Multi-view Learning Methods

Same Facial Features in Multiple Views

We evaluate the proposed DS-GPLVM model across views in both view-invariant and multi-view setting. The former refers to the scenario where data from all views are used for training, while testing is performed using data from each view separately, and the latent space is back-constrained using the IBP. The latter refers to the scenario where data from all views are used during training and testing, and the latent space is back-constrained using the SBP. The same strategy was used for evaluation of other multi-view techniques *i.e.*, GMLDA and GMLPP. Table 4.1 summarizes the results for the three sets of features, averaged across the five views from MultiPIE. We see that the facial points (feature set (I)) result in a more discriminative descriptor for all methods, although we end up with higher standard deviation compared to the appearance features (feature sets (II) and (III)). Evidently, DS-GPLVM outperforms the other view-invariant and multi-view models on all three feature sets, showing that it can successfully unravel the discriminative shared-space that is better suited for FER. Interestingly, in this experiment LDA- and LPP-based linear methods achieve high accuracy, which is comparable to that of D-GPLVM and GPLRF. Moreover, GMLDA and GMLPP perform similarly to their single view trained counterparts, indicating that they were not able to fully benefit from the presence of additional views. We also observe a similar performance of the MvDA and the standard LDA. Note that the accuracy of DS-GPLVM is higher by 3% than that of GPLRF, which is a special case of DS-GPLVM. We attribute this to the ability of the DS-GPVLM to integrate the discriminative information from multiple views into the shared space.

Table 4.2 shows the performance of the models tested across all views, when feature set (I) (the best for all the models from Table 4.1) is used. It is evident that the proposed DS-GPLVM

Table 4.2: View-invariant classification rate on MultiPIE dataset for the best feature set (*i.e.*, facial points (I)). IBP version of DS-GPLVM is trained using all available views, and tested per view. The reported standard deviation is across 5 folds.

Methods	Poses				
	$-30°$	$-15°$	$0°$	$15°$	$30°$
kNN	80.88 ± 0.007	81.74 ± 0.014	68.36 ± 0.054	75.03 ± 0.024	74.78 ± 0.012
LDA	92.52 ± 0.015	94.37 ± 0.013	77.21 ± 0.014	87.07 ± 0.040	87.47 ± 0.007
LPP	92.42 ± 0.017	94.56 ± 0.011	77.33 ± 0.021	87.06 ± 0.045	87.68 ± 0.011
D-GPLVM	91.65 ± 0.017	93.51 ± 0.009	78.70 ± 0.021	85.96 ± 0.040	86.04 ± 0.010
GPLRF	91.65 ± 0.017	93.77 ± 0.007	77.59 ± 0.021	85.66 ± 0.026	86.01 ± 0.008
GMLDA	90.47 ± 0.012	94.18 ± 0.007	76.60 ± 0.029	86.64 ± 0.032	85.72 ± 0.015
GMLPP	91.86 ± 0.013	94.13 ± 0.002	78.16 ± 0.013	87.22 ± 0.023	87.36 ± 0.008
MvDA	92.49 ± 0.011	94.22 ± 0.014	77.51 ± 0.022	87.10 ± 0.031	87.89 ± 0.010
DS-GPLVM	**93.55 ± 0.019**	**96.96 ± 0.012**	**82.42 ± 0.018**	**89.97 ± 0.023**	**90.11 ± 0.028**

performs consistently better than the compared models across all views. Note that all models achieve the lowest classification rate in the frontal view. However, the DS-GPLVM significantly improves the performance attained by the other models in this view. We attribute this to the fact that DS-GPLVM performs the classification in the shared space, where the classification of the expressions from the frontal view is facilitated due to the discriminative information learned from the other views. Furthermore, it is worth noting that the models' accuracy on the negative pan angles (the left side of the face) is higher than on the corresponding positive pan angles (the right side of the face). Since MultiPIE contains more examples of negative emotion expressions, this confirms recent findings in [138] showing that the left hemisphere of the face is more informative when it comes to expressing negative emotions (*e.g.*, Disgust). The right hemisphere is more informative for positive emotions (*e.g.*, Happiness). In other words, due to the imbalance of the emotion categories in the used dataset, the learned classifiers were biased toward negative emotion expressions, and, hence, to the negative pan angles.

Table 4.3 compares the performance of the SBP variant of DS-GPLVM with other multi-view learning methods on three feature sets. The poor performance of KCCA can be attributed to its inherent propensity to overfitting the training data, as also observed in, *e.g.*, [79]. In addition, both CCA and KCCA do not use any supervisory information during the subspace learning, which further explains their low performance. By comparing GPLRF (with concatenated features from different views) and DS-GPLVM, we see that the former, although not a multi-view method, performs comparably to our DS-GPLVM in the case of feature set (I). We attribute this to the fact that GPLRF can effectively explain the variations in facial points from multiple views using a single GP. Yet, because of the large variation in the appearance of facial expressions from different views, the same is not the case when feature sets (II) and (III) are used. When compared to the state-of-the-art methods for multi-view learning (GMA and

Table 4.3: Classification rate for the multi-view testing scenario using the SBP version of DS-GPLVM. The reported standard deviation is across the 5 folds.

Methods	Features		
	I	II	III
PW-CCA	72.42 ± 0.020	73.56 ± 0.025	56.07 ± 0.028
PW-KCCA	52.92 ± 0.039	69.15 ± 0.017	42.42 ± 0.026
GPLRF (conc.)	97.37 ± 0.014	89.42 ± 0.012	89.94 ± 0.012
GMLDA	96.33 ± 0.015	93.04 ± 0.011	92.15 ± 0.013
GMLPP	96.20 ± 0.014	91.37 ± 0.019	90.83 ± 0.017
MvDA	97.12 ± 0.017	93.56 ± 0.011	92.81 ± 0.015
DS-GPLVM	**97.98 ± 0.008**	**93.96 ± 0.015**	**93.29 ± 0.010**

MvDA), DS-GPLVM performs similarly or better on all three feature sets. Furthermore, the SBP version of DS-GPLVM during inference succeeds to model complementary information from all available views, resulting in a higher accuracy compared to the best performing view, *i.e.*, $-15°$, of the IBP variant of DS-GPLVM (see Table 4.2).

Feature Fusion

We next evaluate DS-GPLVM in the feature fusion task, where the goal is to augment view-invariant facial expression classification by fusing different feature sets. Specifically, we trained the SBP version of DS-GPLVM using the three feature sets extracted from the frontal view only. This choice has been made because the frontal view is not the most informative one ($-15°$ is), and hence, there is a lot space for improvement. From Table 4.4, we see that the accuracy of DS-GPLVM in the frontal view outperforms that achieved by the GPLRF by more than 3%, where the features are simply concatenated and used as input. This is because GPLRF cannot fully account for the variations in all three feature sets using a single GP. By contrast, DS-GPLVM learns separate GPs for each feature set, resulting in improved classification performance in the frontal view. It is also important to mention that by training GPLRF using each feature set separately, we obtained the following classification rates: 77.6%, 81.3% and 82.1%, for feature sets (I), (II), and (III), respectively. Compared to the accuracy of DS-GPLVM in Table 4.4 (87.1%), the proposed feature fusion significantly outperforms each of the feature sets used independently. This is expected since the appearance features (LBPs and DCT), extracted from local patches, do not encode global information about face geometry, which is efficiently encoded by facial points. On the other hand, facial points are not informative regarding transient changes in facial appearance (*e.g.*, wrinkles and bulges) which are successfully captured by the appearance features. Thus, the combination of these features within the proposed framework turns out to be highly effective. The other multi-view methods also achieve significant increase in their performance (apart from GMLDA). However,

Table 4.4: Accuracy of the augmented classification in the frontal pose. Feature fusion is attained with the SBP version of DS-GPLVM.

Methods				
GPLRF (conc.)	GMLDA	GMLPP	MvDA	DS-GPLVM
83.16 ± 0.021	78.94 ± 0.018	85.95 ± 0.019	86.19 ± 0.014	**87.13 ± 0.019**

DS-GPLVM outperforms (although marginally in some cases) all these state-of-the-art models.

Same Facial Features in Different Illumination

Herein, we evaluate the proposed DS-GPLVM under different illumination on MultiPIE, where the goal is to learn an illumination-free manifold for facial expression classification. For the purposes of this experiment, we use only images from the frontal view with two different lighting conditions: (i) no lighting source (dark view), and (ii) lighting from the flash of the corresponding camera (bright view). Each lighting condition has been considered as a separate view to train the IBP variant of DS-GPVLM with feature set III. DCT features are selected, since they are less robust to illumination variations than LBPs, and thus a difference in the performance between the two illumination conditions is expected. In Table 4.5 we see that this difference is present in the results of the single-view method, *i.e.*, the GPLRF. The latter is trained separately for each lighting condition, and hence, the two learned manifolds falsely encode the illumination as important information, resulting in a considerable gap between the performance of the bright and the dark view. Contrary to that, the compared multi-view methods, *i.e.*, GMLDA, GMLPP and MvDA, manage to remove, to some extent, the lighting condition of the views under the common space. This is evidenced by the improvement on the performance of the dark view, although a notable difference between the performance of the two views still exists. On the other hand, the proposed DS-GPLVM, not only achieves better results under both illumination conditions, but it also manages to align them by discarding the illumination under the shared space. Note that the DS-GPLVM reports similar classification rate, regardless the original lighting condition of the view.

4.4.4 Comparisons with other Multi-view Methods

Herein, we compare DS-GPLVM (with the IBP variant using feature set (III)) to the state-of-the-art methods for view-invariant facial expression classification. The results for the LGBP-based method, where the LBP features are extracted from Gabor images, are obtained from [127]. For the method in [175], we extracted the sparse SIFT (SSIFT) features from the same images that we used from MultiPIE. In both of the aforementioned methods, the

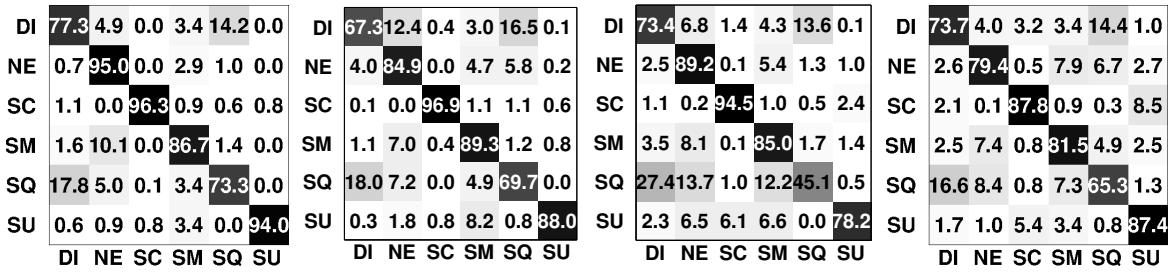
Table 4.5: Classification rate on the frontal view under different illumination for feature set (III). The IBP variant of DS-GPLVM was used. The reported standard deviation is across the 5 folds.

Methods	Illumination	
	Frontal flash 	No flash 
GPLRF	82.09 ± 0.015	77.00 ± 0.025
GMLDA	82.76 ± 0.017	84.01 ± 0.029
GMLPP	82.10 ± 0.029	84.75 ± 0.030
MvDA	83.80 ± 0.015	84.20 ± 0.019
DS-GPLVM	**85.51 ± 0.032**	**85.68 ± 0.021**

target features (LGBP and SSIFT) are extracted per-view, and then fed into the view-specific SVM classifiers. We also compared our model to the coupled GP (CGP) [148], where first view-normalization is performed by projecting a set of facial points (feature set (I)) from non-frontal views to the canonical view. In our experiments with CGP, we set the canonical view to the most discriminative view among the positive pan angles (*i.e.*, 15°). This was followed by classification using the SVM learned in this view. Table 4.6 shows the comparative results. We observe first that all methods (except [175]) achieve the best results for the 15° view, indicating that regardless of the method/features employed, this view is more discriminative (among the positive pan angles) for the target task. We also note that DS-GPLVM outperforms on average the other two methods, which are based on the appearance features. This difference is in part due to the features used and in part due to the fact that the methods in [127] and [175] both fail to model correlations between different views. By contrast, the CGP method accounts for the relations between the views in a pair-wise manner, while DS-GPLVM does so for all the views simultaneously. Hence, the proposed DS-GPLVM shows superior performance to that of CGP. This is because CGP performs view alignment (i) directly in the observation space, and (ii) without using any discriminative criterion during this process. Thus, the effects of high-dimensional noise and the errors of view-normalization adversely affect its performance in the classification task. On the other hand, DS-GPLVM imposes further constraints on the shared manifold, resulting in a better performance on the target task. This is also reflected in the confusion matrices in Fig. 4.4. Note that the main source of confusion is between the facial expressions of *Disgust* and *Squint*. This is because they are characterized by similar facial changes in the region of the eyes. However, the proposed DS-GPLVM improves significantly the accuracy on *Squint*, compared to the other models.

Table 4.6: Comparison of state-of-the-art methods on the MultiPIE database. The IBP version of DS-GPLVM with feature set (III), outperforms the state-of-the-art methods for view-invariant facial expression classification. The reported standard deviation is across 5 folds.

Methods	Poses		
	0°	15°	30°
LGBP [127]	82.1	87.3	75.6
SSIFT [175]	81.14 ± 0.009	79.25 ± 0.016	77.14 ± 0.019
CGP [148]	80.44 ± 0.017	86.41 ± 0.013	83.73 ± 0.019
DS-GPLVM	**84.31 ± 0.025**	**89.21 ± 0.015**	**90.26 ± 0.025**



(a) DS-GPLVM (b) CGP (c) SSIFT (d) LGBP

Figure 4.4: Comparative confusion matrices for facial expression classification over all angles of view for the (a) DS-GPLVM, (b) CGP, (c) SSIFT and (d) LGBP.

4.4.5 Cross Dataset Experiments on MultiPIE and LFPW

In this section, we test the ability of DS-GPLVM (the IBP variant) to generalize to unseen real-world spontaneous data. We evaluate different models on the smile detection task, where the feature set (I) extracted from images from MultiPIE is used for training. Images from LFPW are used for testing. This is a rather challenging task mainly because the test images are captured in an uncontrolled environment, which is characterized by large variation in head poses and illumination, and occlusions of parts of the face. Also, the models are trained using data of *posed* (deliberately displayed as opposed to spontaneous and 'in the wild') expressions, which can differ considerably in subtlety compared to the *spontaneous* expressions used for testing. The difficulty of the task is evidenced by the results in Table 4.7, where we observe a significant drop in accuracy of all methods. Furthermore, we observe that the most informative views for smile detection are the ones with positive degrees (the right side of the face). This, again, is for the same reasons as explained in Sec. 4.4.3. However, all methods attain the highest accuracy in the frontal pose. We attribute this to the fact that the faces with non-frontal poses do not exactly belong in the discrete set of poses, but rather in a continuous range from 0° to ±30°. Thus, the accuracy of the pose registration significantly affects the performance of the models. Nevertheless, the proposed DS-GPLVM outperforms the other models by a large margin in all poses except −30°. To explain this, we checked the number of

Table 4.7: Smile detection on images from the LFPW dataset. The methods were trained on images from the MultiPIE dataset using feature set (I). We used the IBP version of DS-GPLVM for the view-invariant facial expression classification.

Method	Poses				
	−30°	−15°	0°	15°	30°
GMLDA	69.00	43.00	80.94	55.76	76.00
GMLPP	**70.00**	47.50	81.25	57.58	79.66
MvDA	**70.00**	50.00	81.25	51.52	**80.00**
DS-GPLVM	55.33	**58.00**	**90.00**	**74.55**	**80.00**

Table 4.8: Classification rates per expression category obtained by different models trained/tested using the SFEW dataset.

	Anger	Disgust	Fear	Happiness	Neutral	Sadness	Surprise	Average
Baseline	23.00	13.00	13.90	29.00	23.00	17.00	13.50	18.90
GMLDA	23.21	17.65	**29.29**	21.93	25.00	11.11	10.99	19.90
GMLPP	16.07	21.18	27.27	39.47	20.00	19.19	**16.48**	22.80
MvDA	23.21	17.65	27.27	40.35	**27.00**	10.10	13.19	22.70
DS-GPLVM	**25.89**	**28.24**	17.17	**42.98**	14.00	**33.33**	10.99	**24.70**

test examples of smiles in this pose, and found that only few were available (contrary to other poses, which contained far more examples). Therefore, the misclassification of some resulted in a significant drop in the performance of DS-GPLVM.

4.4.6 Expression Recognition on Real World Images from SFEW

Finally, we evaluate the models on the feature fusion task, where the features are extracted from images of spontaneously displayed facial expressions in real-world environment. Specifically, we used LPQ [132] and PHOG [22] features from expressive images from the SFEW dataset. Contrary to the cross-dataset evaluation from the previous section, here both training and testing are performed using real-world spontaneous facial expressions. Note that LPQ is a texture descriptor that captures local information over a neighborhood of pixels, resulting in its being robust to illumination changes. On the other hand, PHOG is a local descriptor which is capable of preserving the spatial layout of the local shapes in an image. Thus, we expect the fusion of these two to achieve improved performance on the target task. The provided images of SFEW were originally divided into two subject independent folds, and we report the average results over the folds.

Table 4.8 shows the results obtained for different methods. We employ the SBP variant of the DS-GPLVM. As a baseline we use the results obtained by the database creators [44]. The authors used a non-linear SVM classifier on the concatenation of the features to report the classification rate for the fusion task. We can see that all the employed multi-view learning

methods outperform the baseline, on average. This is due to their ability to effectively exploit the discriminative information that is embedded in both feature spaces. However, in most cases, the linear multi-view learning methods are outperformed by the proposed DS-GPLVM. We attribute this to the fact that the linear models are unable to fully fuse the employed features on a linear shared space. By contrast, better fusion is attained by the non-linear mappings in the DS-GPLVM, resulting in its average performance being the best among the tested models. Note, however, that in the case of Surprise, Fear and Neutral, DS-GPLVM reports the lowest performance. By inspecting the back-projected test examples of these two expressions on the shared manifold, we observed that Neutral was spread around other emotion categories. This finding suggests that the learned back-projections of DS-GPLVM cannot effectively explain the varying level of expressiveness among the different subjects. Hence, examples of expressive images with low-intensity levels are being recognized as Neutral. Nevertheless, DS-GPLVM outperforms the other models on the remaining expressions, with a considerable improvement on Disgust, Happiness and Sadness.

4.5 Conclusion

In this chapter, we proposed the DS-GPLVM model for learning a discriminative manifold shared among expressive images from multiple views. Due to the introduced prior, which explicitly encodes the class information from the available labels, the recovered latent space is optimial for the expression classification task. The DS-GPLVM can be regarged as a multi-view generalization of latent variable models that learn a discriminative subspace from a single observation space. As such, DS-GPLVM constitutes a complete non-parametric multi-view framework that can instantiate other non-linear single-view models (*i.e.*, D-GPLVM [183] and GPLRF [212]), and can also extend the linear multi-view techniques (*i.e.*, GMA [165] and MvDA [95]) to their non-linear counterparts. The conducted experimental analysis on posed and spontaneously displayed facial expressions, indicates that modeling of the manifold shared across different views and/or features using the proposed framework considerably improves both multi- and per- view/feature classification of facial expressions.

Chapter 5

Latent Variable Models for Joint Action Unit Detection

Contents

5.1 Introduction

As we have already discussed, facial expressions are typically encoded as a combination of facial muscle activations, *i.e.*, action units (AUs). Depending on context, these AUs co-occur in specific patterns, and rarely operate in isolation. Yet, most existing methods for automatic AU detection fail to exploit dependencies among them, and even if they do so, they cannot exploit correlations between different types of facial features. This has an adverse impact on the detection task. Hence, a desired model should be able to account for the variations in both sources, *i.e.*, input features and output labels. To our knowledge, the only methods that attempt both are [194, 215, 206]. However, these methods either suffer from the curse of dimensionality as they perform feature fusion by concatenation of geometric- and appearance-based features using parametric models [194, 215], or cannot model more than a few AUs jointly due to the computational burden of their (non-parametric) inference methods [206].

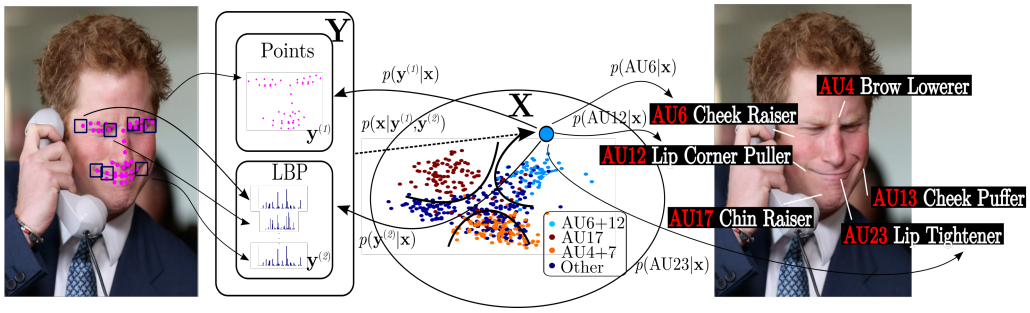


Figure 5.1: The proposed MC-LVM. The geometrical and appearance input features, $\boldsymbol{y}^{(1)}$ and $\boldsymbol{y}^{(2)}$, are first projected onto the shared manifold $\boldsymbol{X}$. The fusion is attained via GP conditionals, $p(\boldsymbol{y}^{(1)}|\boldsymbol{x})$ and $p(\boldsymbol{y}^{(2)}|\boldsymbol{x})$, that generate the inputs. Classification is performed on the manifold via jointly learned logistic functions $p(z^{(c)}|\boldsymbol{x})$ for multiple AU detection. The subspace is regularized using constraints imposed on both latent positions and output classifiers, encoding local and global dependencies among the AUs.

In this chapter, we propose a multi-conditional latent variable model (MC-LVM) that performs simultaneously the fusion of different facial features and joint detection of AUs. Instead of performing the AU detection in the original feature space, as done in existing works [194, 206, 215], the MC-LVM attains the feature fusion via a low-dimensional subspace shared across the feature sets. This subspace is learned by employing the framework of shared GPs [167]. Here, the learning is constrained by two types of newly introduced constraints. *Topological* constraints encode local dependencies (from image pairs) among multiple AUs by means of string kernels [146]. *Relational* constraints, enforce the co-occurrences of the model predictions to match those of the target labels. The learning of the subspace is performed jointly with the AU detectors. The latter are modeled via multiple logistic regressors which operate on the shared subspace of the fused features. Note that, in contrast to existing multi-output subspace learning methods (*e.g.*, [202, 1]), the MC-LVM learns a subspace for multiple AU detection that combines both the generative and discriminative properties of probabilistic models, while simultaneously modeling the AU correlations at both feature level (via the proposed fusion approach) and model level (via the introduced regularizers). Due to its multi-conditional likelihood function, the proposed model is less susceptible to overfitting compared to purely discriminative models. Its generative part acts as an efficient regularizer during the learning stage. The proposed multi-conditional learning is motivated by the fact that discriminative learning usually yields better results when provided with sufficient training data. On the other hand, generative models, if specified well, can generalize better with fewer training data [97]. Thus, leveraging the advantages of the two approaches during the model learning process is expected to lead to better generalization performance. To further improve the robustness and efficiency of the parameter estimation, a Bayesian learning of the

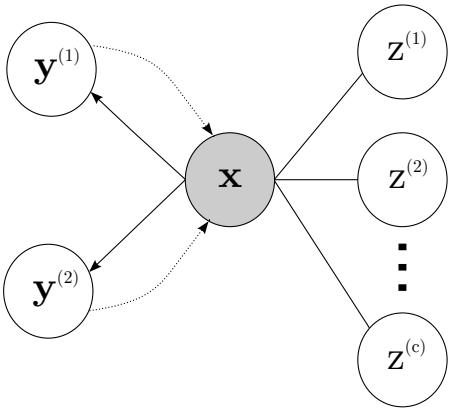


Figure 5.2: Graphical representation of the proposed MC-LVM. The definition of the conditionals is given in Section 5.2.3.

data subspace is facilitated through Monte Carlo sampling, and an expectation-maximization (EM)-like learning approach. During inference, the simultaneous detection of multiple AUs is performed by applying the learned back-mappings from inputs to the shared subspace, where the detection of target AUs is performed consequently. The outline of the proposed approach is illustrated in Fig. 5.1. Note that the contents of this chapter are published in [62, 63].

5.2 Multi-conditional Latent Variable Model

5.2.1 Notation and Preliminaries

Let us denote the training set as $\mathcal{D} = \{\boldsymbol{Y}, \boldsymbol{Z}\}$, which is comprised of V observed and corresponding input channels $\boldsymbol{Y} = \{\boldsymbol{Y}^{(v)}\}_{v=1}^{V}$, and the associated output labels $\boldsymbol{Z}$. Each observed channel is comprised of N i.i.d. multivariate samples $\boldsymbol{Y}^{(v)} = \{\boldsymbol{y}_i^{(v)}\}_{i=1}^{N}$, where $\boldsymbol{y}_i^{(v)} \in \mathbb{R}^{D_v}$ denote corresponding facial features across the multiple channels. Furthermore, $\boldsymbol{Z} = \{\boldsymbol{z}_i\}_{i=1}^{N}$ denote multiple binary labels, with $\boldsymbol{z}_i \in \{-1, +1\}^C$ encoding C (co-occurring) outputs. Let us further assume the existence of a latent space $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^{N}$, where $\boldsymbol{x}_i \in \mathcal{R}^q$, $q \ll D_v$, which is a low-dimensional representation of the original observations $\boldsymbol{Y}$. This implies that there exists a set of latent functions $f^{(v)}$, that can generate $\boldsymbol{y}_i^{(v)}$ from $\boldsymbol{x}_i$, *i.e.*, $\boldsymbol{y}_i^{(v)} = f^{(v)}(\boldsymbol{x}_i) + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma_v^2 \boldsymbol{I})$ is additive Gaussian noise. In the proposed approach we model these functions using the framework of GPs [146]. For notation simplicity, we set the number of input spaces to $V = 2$, as generalization to more than two input spaces is straightforward. The model outline is depicted in Fig. 5.2.

5.2.2 MC-LVM: Model Definition

Our goal is to learn a model that simultaneously combines different inputs and detects activations of multiple outputs. We are interested in finding the latent representations $\boldsymbol{x}$, that jointly generate $\boldsymbol{y}$ and $\boldsymbol{z}$. In a Bayesian approach, this requires the computation of the joint marginal likelihood:

$$p(\boldsymbol{y}, \boldsymbol{z}) = \int p(\boldsymbol{y}^{(1)}|\boldsymbol{x})p(\boldsymbol{y}^{(2)}|\boldsymbol{x})p(\boldsymbol{z}|\boldsymbol{x})p(\boldsymbol{x})d\boldsymbol{x}, \tag{5.1}$$

where we exploited the property of conditional independence, *i.e.*, $\{\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}, \boldsymbol{z}\}$ are independent given the latent variable $\boldsymbol{x}$. Note that in order to compute the above integral, we need to marginalize out $\boldsymbol{x}$. However, for the non-linear conditional models, which we detail in Section 5.2.3, the integral in Eq. (5.1) is intractable. To overcome this, we numerically approximate the marginal likelihood using Monte Carlo sampling [19]

$$p(\boldsymbol{y}, \boldsymbol{z}) \approx \frac{1}{S}\sum_{s=1}^{S} p(\boldsymbol{y}^{(1)}|\boldsymbol{x}_s)p(\boldsymbol{y}^{(2)}|\boldsymbol{x}_s)p(\boldsymbol{z}|\boldsymbol{x}_s). \tag{5.2}$$

The samples $\boldsymbol{x}_s$, $s = 1, \ldots, S$ are drawn from $p(\boldsymbol{x})$, which is defined in Section 5.2.3. Using the Bayes' rule, we can derive the posterior over the latent variable

$$p(\boldsymbol{x}|\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}, \boldsymbol{z}) = \frac{p(\boldsymbol{z}|\boldsymbol{x})p(\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}|\boldsymbol{x})p(\boldsymbol{x})}{\frac{1}{S}\sum_{s=1}^{S} p(\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}|\boldsymbol{x}_s)p(\boldsymbol{z}|\boldsymbol{x}_s)}. \tag{5.3}$$

We then calculate the above probability for all pairs of training data i and Monte Carlo latent samples s, to obtain the membership probabilities $p(s, i) = p(\boldsymbol{x}_s|\boldsymbol{y}_i^{(1)}, \boldsymbol{y}_i^{(2)}, \boldsymbol{z}_i)$. Hence, $p(s, i)$ denotes the posterior probability of acquiring the sample $\boldsymbol{x}_s$, having observed the inputs $\boldsymbol{y}_i^{(1)}, \boldsymbol{y}_i^{(2)}$ and outputs $\boldsymbol{z}_i$. This gives rise to the expectation of the latent points under the sampling distribution:

$$\boldsymbol{x}_i = E\{\boldsymbol{x}|\boldsymbol{y}_i^{(1)}, \boldsymbol{y}_i^{(2)}, \boldsymbol{z}_i\} = \sum_{s=1}^{S} p(s, i)\boldsymbol{x}_s, \tag{5.4}$$

which allows us to obtain the point estimates of the shared latent positions without explicitly optimizing them for each training pair. In this way, not only we end up with a probabilistic estimate of the latent space, but we also considerably reduce the number of model parameters, and hence, avoid overfitting the latent coordinates.

5.2.3 MC-LVM: Conditional Models

From Eq. (5.1), we see that the marginal likelihood of the desired model is composed of the conditional probabilities $p(\boldsymbol{y}^{(v)}|\boldsymbol{x})$ and $p(\boldsymbol{z}|\boldsymbol{x})$, while it also depends on the sampling distribution $p(\boldsymbol{x})$. Hence, the correct choice of these distributions critically affects the representation

abilities of the shared subspace, and thus, the model's performance. Effectively, this requires the learning of the conditional models that facilitate: (i) *generative* mappings from the latent space to the inputs ($\boldsymbol{x} \rightarrow \boldsymbol{y}^{(v)}, v = 1, 2, \cdots, V$); (ii) *projection* mappings from the inputs to latent space ($\boldsymbol{y}^{(v)} \rightarrow \boldsymbol{x}$); (iii) *discriminative* mappings from latent space to multiple binary outputs ($\boldsymbol{x} \rightarrow \boldsymbol{z}$), as depicted in Fig. 5.2.

Generative mappings. Different probabilistic models such as Gaussian models [21] or naive Bayes models [123] can be employed to recover the generative mappings. Yet, parametric models are limited in their ability to recover non-linear mappings from the latent space to high-dimensional input features. Herein, we place GP priors on the functions that generate the observed features. This gives rise to the likelihood:

$$p(\boldsymbol{Y}^{(v)}|\boldsymbol{X}, \boldsymbol{\theta}^{(v)}) = \frac{1}{\sqrt{(2\pi)^{ND_v}|\boldsymbol{K}_{\boldsymbol{Y}}^{(v)} + \sigma_v^2 \boldsymbol{I}|^{D_v}}} \exp\left[-\frac{1}{2}\mathrm{tr}\left((\boldsymbol{K}_{\boldsymbol{Y}}^{(v)} + \sigma_v^2 \boldsymbol{I})^{-1}\boldsymbol{Y}^{(v)}\boldsymbol{Y}^{(v)^T}\right)\right], \tag{5.5}$$

where $\boldsymbol{K}_{\boldsymbol{Y}}^{(v)}$ is an $N \times N$ kernel matrix, obtained by applying the covariance function $k^{(v)}(\boldsymbol{x}, \boldsymbol{x}')$ to the elements of $\boldsymbol{X}$, and it is shared across the dimensions of $\boldsymbol{Y}^{(v)}$. As a covariance function we choose again the sum of the RBF, bias and noise terms

$$k^{(v)}(\boldsymbol{x}, \boldsymbol{x}') = \theta_1^{(v)} \exp(-\frac{\theta_2^{(v)}}{2}\|\boldsymbol{x} - \boldsymbol{x}'\|^2) + \theta_3^{(v)} + \frac{\delta_{\boldsymbol{x},\boldsymbol{x}'}}{\theta_4^{(v)}}, \tag{5.6}$$

where $\delta_{\boldsymbol{x},\boldsymbol{x}'}$ is the Kronecker delta function, and $\boldsymbol{\theta}^{(v)} = \{\theta_1^{(v)}, \theta_2^{(v)}, \theta_3^{(v)}, \theta_4^{(v)}\}$ are the kernel hyper-parameters. The predictive probability of the specified GP for a new $\boldsymbol{x}_*$ is given by

$$p(\boldsymbol{y}_*^{(v)}|\boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{Y}^{(v)}) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{y}_*^{(v)}}, \sigma^2_{\boldsymbol{y}_*^{(v)}}), \tag{5.7}$$

with $\boldsymbol{\mu}_{\boldsymbol{y}_*^{(v)}}$ and $\sigma^2_{\boldsymbol{y}_*^{(v)}}$ as:

$$\boldsymbol{\mu}_{\boldsymbol{y}_*^{(v)}} = \boldsymbol{k}_*^{(v)^T}(\boldsymbol{K}_{\boldsymbol{Y}}^{(v)} + \sigma_v^2 \boldsymbol{I})^{-1}\boldsymbol{Y}^{(v)} \tag{5.8}$$

$$\sigma^2_{\boldsymbol{y}_*^{(v)}} = k_{**}^{(v)} - \boldsymbol{k}_*^{(v)^T}(\boldsymbol{K}_{\boldsymbol{Y}}^{(v)} + \sigma_v^2 \boldsymbol{I})^{-1}\boldsymbol{k}_*^{(v)} + \sigma_v^2. \tag{5.9}$$

The kernel values $\boldsymbol{k}_*^{(v)}$ and $k_{**}^{(v)}$ are computed by applying Eq. (5.6) to the pairs $(\boldsymbol{X}, \boldsymbol{x}_*)$ and $(\boldsymbol{x}_*, \boldsymbol{x}_*)$, respectively, and σ_v^2 is the noise of the process. Hence, the conditional model $p(\boldsymbol{y}^{(v)}|\boldsymbol{x}), v = 1, 2$, in Eq. (5.3) is now fully defined by the Gaussian distribution in Eq. (5.7), where the latent sample $\boldsymbol{x}_s$ acts as the new latent position $\boldsymbol{x}_*$.

Projection mappings and sampling. To model the sampling distribution $p(\boldsymbol{x})$, the simplest choice is to assume a spherical Gaussian prior over the latent points $\boldsymbol{x}$. However,

such an uninformative prior would give rise to latent representations that cannot effectively exploit the structure of input data. Thus, we define a sampling distribution that constraints the samples $\boldsymbol{x}_s$ by conditioning them on the inputs, *i.e.*, $\tilde{p}(\boldsymbol{x}) = p(\boldsymbol{x}|\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)})$. This is motivated by the notion of back-constraints in GPLVM [106], where this type of conditional distribution is used to learn the mappings from the input to the latent space. We learn the conditional model for $\tilde{p}(\boldsymbol{x})$ using GPs, as done for the generative mappings. The use of GPs in the projection mappings, apart from modeling the sampling distribution, also allows us to easily combine multiple features within its kernel matrix as $\boldsymbol{K}_X = \boldsymbol{K}_X^{(1)} + \boldsymbol{K}_X^{(2)}$, corresponding to the sum of the kernel functions defined on $\boldsymbol{y}^{(1)}$ and $\boldsymbol{y}^{(2)}$, respectively. Hence, the resulting kernel is responsible for effectively performing the non-linear fusion of the input features into a single latent point. It can be regarded as an automatic MKL approach with non-parametric GP regression functions. Finally, the resulting conditional model $p(\boldsymbol{x}_*|\boldsymbol{y}_*^{(1)}, \boldsymbol{y}_*^{(2)})$ has the form of Eq. (5.7) (with the relations between $\boldsymbol{y}^{(v)}$ and $\boldsymbol{x}$ being reverted), and since it is a low-dimensional Gaussian distribution, sampling from it can be performed efficiently.

Discriminative mappings. Since we are interested in binary detection of activations of multiple AUs, we use the conditional models based on the logistic regression [146] to model $p(\boldsymbol{z}|\boldsymbol{x})$. By assuming conditional independence given the latent positions $\boldsymbol{x}$, we can factorize this conditional as:

$$p(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{W}) = p(z^{(1)}|\boldsymbol{x}, \boldsymbol{w}_1) \dots p(z^{(C)}|\boldsymbol{x}, \boldsymbol{w}_C), \tag{5.10}$$

$$p(z^{(c)}|\boldsymbol{x}, \boldsymbol{w}_c) = (1 + e^{-\boldsymbol{x}^T \boldsymbol{w}_c z^{(c)}})^{-1}, \ c = 1, \dots, C, \tag{5.11}$$

where $\boldsymbol{W} = [\boldsymbol{w}_1, \dots, \boldsymbol{w}_C] \in \mathcal{R}^{q \times C}$ contains the weight vectors of the individual functions. During inference, if $p(z_*^{(c)}|\boldsymbol{x}_*) > 0.5$, the c-th output is active, *i.e.*, $z_*^{(c)} = 1$.

5.2.4 MC-LVM: Output Constraints

Due to the potentially large number of outputs, the topology of the latent space needs to be constrained to avoid the model focusing on unimportant variation in the data (*e.g.*, modeling relations between rarely co-occurring outputs). Furthermore, we need to encourage the model to produce similar predictions for outputs that are more likely to co-occur (*e.g.*, AU6+12), and competing predictions for those that rarely co-occur (*e.g.*, AU12 and AU17). We describe below how we construct appropriate constraints based on the output relations, and how these are incorporated into the MC-LVM framework.[1]

[1]For the mathematical analysis of this subsection, the negative class in the output labels $\boldsymbol{z}$ will be denoted with 0 instead of the used -1.

Topological constraints. Herein, we define the constraints that encode co-occurrences of the output labels using the notion of graph regularization [29]. This process resembles the one we described in Section 4.2.2 for constraining the DS-GPLVM with the discriminative information. However, the challenge here is to design a similarity matrix that would encode the discriminative information from the multiple available labels. Hence, we construct the matrix by measuring the similarity between the output label vectors using the notion of string kernels [146] as:

$$\boldsymbol{S}(\boldsymbol{x}, \boldsymbol{x}') = \sum_{l \in \mathcal{A}} \boldsymbol{z}_{l,x}^T \boldsymbol{z}_{l,x'}, \tag{5.12}$$

where $\mathcal{A}$ is the set of all possible 2^C combination of the output labels and l is the set of possible sub-labels of tuples, triples, etc. $\boldsymbol{z}_{l,x}$ denotes the specific sub-label of $\boldsymbol{x}$ and holds the currently active 'sub-string' l of the actual labels. Hence, $\boldsymbol{S}_{ij}$ contains the number of co-activated outputs in all sub-labels between two instances i and j. Note that contrary to [209], we measure the similarity of the outputs based on all possible groups of co-occurring AUs, and not only on pairs of AUs. The graph Laplacian matrix is then defined as $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{S}$, where $\boldsymbol{D}$ is a diagonal matrix with $\boldsymbol{D}_{ii} = \sum_j \boldsymbol{S}_{ij}$. Finally, using Eq. (5.4), we arrive at the Laplacian regularization term

$$C = \text{tr}(\boldsymbol{X}^T \boldsymbol{L} \boldsymbol{X}) = \sum_{i,j}^{N} \sum_{s=1}^{S} \sum_{t=1}^{S} L_{ij} p(s,i) p(t,j) \boldsymbol{x}_s^T \boldsymbol{x}_t. \tag{5.13}$$

Eq. (5.13) incurs higher penalty if latent projections of co-occurring AUs are distant in the manifold. Thus, projections with strongly related AUs are placed close to each other.

Global relational constraints. In order for the MC-LVM to fully benefit from the above topological constraint, it is important to ensure that the model produces similar predictions for frequently co-occurring AUs. Therefore, we introduce the global *relational* constraints as:

$$R = \|\boldsymbol{P}_{\boldsymbol{z}}^T \boldsymbol{P}_{\boldsymbol{z}} - \boldsymbol{Z}^T \boldsymbol{Z}\|_F^2, \tag{5.14}$$

where $\boldsymbol{P}_{\boldsymbol{z}} = [p(\boldsymbol{z}_1|\boldsymbol{x}_1), \ldots, p(\boldsymbol{z}_N|\boldsymbol{x}_N)]^T$ are the predictions from Eq. (5.11) for each $\boldsymbol{x}_i$, and $\boldsymbol{Z}$ is the true label set. Thus, Eq. (5.14), incurs a high penalty if correlated outputs have dissimilar predictions. In this way, the co-occurrence matrix of the predictions is forced to be similar to that of the true labels, and hence, the discriminative power of the output detectors is increased.

5.2.5 MC-LVM: Learning and Inference

The objective function of our model is the sum of the complete data log-likelihood of the (weighted) joint distribution in Eq. (5.2) penalized by the constraints in Eq. (5.13)–(5.14)

$$\mathcal{L}(\boldsymbol{\Theta}) = \sum_{i=1}^{N} \log \sum_{s=1}^{S} \underbrace{p(\boldsymbol{y}_i^{(1)}, \boldsymbol{y}_i^{(2)} | \boldsymbol{x}_s)}_{p_{gen}}^{1-\alpha} \underbrace{p(\boldsymbol{z}_i | \boldsymbol{x}_s)}_{p_{disc}}^{\alpha} - \lambda_C C - \lambda_R R, \tag{5.15}$$

where $\boldsymbol{\Theta} = \{\boldsymbol{\theta}^{(v)}, \boldsymbol{W}\}$. Note that in contrast to the standard maximum likelihood (ML) optimization, we set the parameter $\alpha \in [0, 1]$ to find an optimal balance between the generative (p_{gen}) and discriminative (p_{disc}) components of our MC-LVM. The generative component has the key role in unraveling the latent space of the fused features, while the discriminative component regularizes the manifold by using the labels' structure information. Large α values give rise to models that depend more on the labels to define the decision boundaries for the detection, while for small α the model expends more effort on capturing the variations in the features (*e.g.*, due to various sources of noise in data such as head-pose variation in spontaneous data). By finding optimal α via a cross-validation procedure based on a grid search, as explained in Section 5.4.2, we allow the model to find a trade-off between the discriminative and generative part.

Another key difference to the ML approach, is that the Bayesian optimization requires the computation of the posterior of the latent space. The latter depends on the parameters $\boldsymbol{\Theta}$, and thus, direct optimizing of the objective in Eq. (5.15) w.r.t. $\boldsymbol{\Theta}$ is not possible. Hence, we propose an EM-based approach for parameter learning. In the E-step, we find the expectation of the complete data log-likelihood in Eq. (5.15) under the posterior in Eq. (5.3), which is given by

$$Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(old)}) = \sum_{i=1}^{N} \sum_{s=1}^{S} p(s, i) \log \left(p(\boldsymbol{y}_i^{(1)}, \boldsymbol{y}_i^{(2)} | \boldsymbol{x}_s)^{1-\alpha} p(\boldsymbol{z}_i | \boldsymbol{x}_s)^{\alpha} \right), \tag{5.16}$$

where the membership probabilities, $p(s, i)$, are computed with $\boldsymbol{\Theta}^{(old)}$. In the M-step, we find $\boldsymbol{\Theta}^{(new)}$ by optimizing

$$\boldsymbol{\Theta}^{(new)} = \arg\max_{\boldsymbol{\Theta}} \quad Q(\boldsymbol{\Theta}, \boldsymbol{\Theta}^{(\text{old})}) - \lambda_C C - \lambda_R R, \tag{5.17}$$

w.r.t. $\boldsymbol{\Theta}$ using the conjugate gradient method.[2]

The full training of the model is split into two stages, where in each stage we compute $p(\boldsymbol{x}|\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)})$ and $p(\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}, \boldsymbol{z}|\boldsymbol{x})$ in an alternating fashion. Specifically, we first initialize

[2] We used Rasmussen's *minimize.m* function provided from http://learning.eng.cam.ac.uk/carl/code/minimize/.

the latent coordinates $\boldsymbol{X}$, using a dimensionality reduction method, *e.g.*, PCA, on the concatenation of the two feature sets. Then, we learn the sampling distribution $p(\boldsymbol{x}|\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)})$ by training a GP on the projection mappings, as explained in Section 5.2.3, and collect S samples from corresponding GP posterior. During the second stage, we employ the EM algorithm described above to learn the parameters $\boldsymbol{\Theta}$. Note that both the topological and relational constraints implicitly depend on the posterior, which is a function of the current estimate of $\boldsymbol{\Theta}$, hence, we need to compute their derivatives w.r.t to $\boldsymbol{\Theta}$. The penalized log-likelihood can be optimized jointly [21] or separately [80] without violating the EM-optimization scheme, since the updates from the penalty terms do not affect the computation of the expectation. After the M-step we refine our original estimate of the latent space $\boldsymbol{X}$, using Eq. (5.4). We iterate between stage 1 and 2 until convergence of the objective function in Eq. (5.17).

Algorithm 2 MC-LVM: Learning and Inference

Learning
Inputs: $\mathcal{D} = (\boldsymbol{Y}^{(v)}, \boldsymbol{Z}), v = 1, \ldots, V$
Initialize $\boldsymbol{X}$ using PCA on the concatenated $\boldsymbol{Y}^{(v)}$.
repeat
  Stage 1
    Learn $\tilde{p}(\boldsymbol{x}) = p(\boldsymbol{x}|\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)})$ by training the specified GP.
    Draw S samples $\boldsymbol{x}_s$ from the Gaussian distribution $\tilde{p}(\boldsymbol{x})$.
  Stage 2
    E-step: Use the current estimate of the parameters $\boldsymbol{\Theta}^{(old)}$ to compute the membership probabilities in Eq. (5.3).
    M-step: Update $\boldsymbol{\Theta}$ by maximizing Eq. (5.17).
  Stage 3
    Update the latent space using Eq. (5.4).
until convergence of Eq. (5.17).
Outputs: $\boldsymbol{X}$, $\boldsymbol{\Theta}$

Inference
Inputs: $\boldsymbol{y}_*^{(1)}, \boldsymbol{y}_*^{(2)}$
Step 1: Find the projection $\boldsymbol{x}_*$ to the latent space using Eq. (5.8).
Step 2: Apply the logistic classifiers from Eq. (5.11) to the obtained embedding to compute the outputs $\boldsymbol{z}_*$.
Output: $\boldsymbol{z}_*$

Inference: Inference in the proposed MC-LVM is straightforward. The test data $\boldsymbol{y}_*^{(1)}, \boldsymbol{y}_*^{(2)}$, are first projected onto the manifold using Eq. (5.7). In the second step, the activation of each output is detected by applying the classifiers from Eq. (5.11) to the obtained latent position. The learning and inference procedure described above is summarized in Alg. 2.

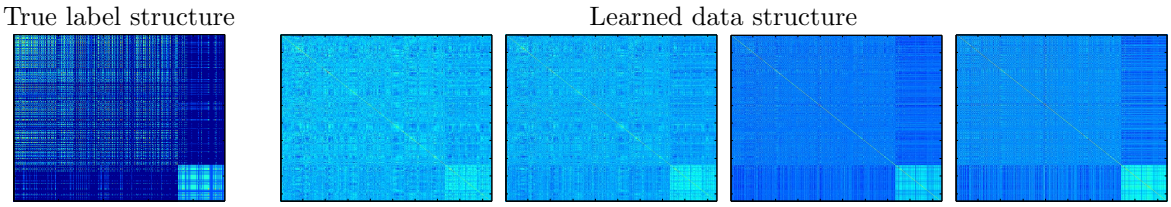


Figure 5.3: Evolution of the learned data structure in $\boldsymbol{K}_{\boldsymbol{Y}}^{(1)}$, through the EM-iterations during the optimization on CK+ dataset. The kernels are sorted in order to depict the structure of AU12 (bottom right square) compared to other AU activations (upper right square).

Theoretical Analysis: The optimization scheme described earlier in this section does not have theoretical guarantees that it increases the penalized complete log-likelihood after each EM cycle. The reasons behind this are twofold: (i) Eq. (5.17) cannot be solved analytically, and thus, we need to resort to an iterative procedure based on the conjugate gradient method. Therefore, in each M-step we can only guarantee that a local optimum of the posterior will be recovered. (ii) The expectation of the complete log-likelihood in Eq. (5.16) is numerically approximated via Monte Carlo sampling, and thus, as in every stochastic optimization problem there is no guarantee that the objective function will strictly increase after each iteration. Hence, it is required to take cautious steps in order not to derive diverge solutions. By carefully initializing the latent coordinates (*e.g.*, via PCA) and the kernel hyper-parameters (*e.g.*via following common heuristics regarding the length scales etc.), and appropriate selection of the number of samples, S, we can effectively learn a latent space with correctly recovered data structure. This is illustrated in Fig. 5.3, where we can see how the topological constraint imposes the structure of AU12 on the manifold, through the evolution of the iterative EM algorithm. In the initialization step the latent space can roughly model the structure of the positive class (AU12). As the EM iterations progress we see that MC-LVM not only uncovers the structure of AU12 (iteration #5), but it also differentiates it from the structure of the remaining AUs (iteration #15). Additional experimental evaluations regarding the convergence of MC-LVM and the effect of the various parameters to the solution are given in Sec. 5.4.2.

Complexity: Since MC-LVM is based on the framework of GPs, each iteration during training (within an EM cycle) requires $\mathcal{O}(N^3)$ computations. On the other hand, inference for a new test sample is far more efficient and can be achieved in real-time, since the evaluation of the predictive mean requires $\mathcal{O}(N)$ (predictive variance is not required for classifying a new test point).

5.3 Relation to Prior Art

5.3.1 Multi-label Classification

The proposed MC-LVM is related to existing works on multi-label classification that attempt to learn robust classifiers by exploiting efficiently the label dependencies. For an extensive overview, the reader is referred to [182, 172]. For instance, [204] extended the k-nearest neighbor (kNN) to the multi-label scenario by using the number of neighboring instances belonging to each possible class, as prior information to determine the label set for an unseen instance. In [203] the authors derived the back-propagation algorithm of the neural networks for the multi-label classification. [66] proposed an approximate learning approach in order to extended the work of structured SVM [181] to multi-label classification. The latter is also highly related to multi-task learning techniques. These techniques rely on the introduction of an inductive bias on the joint space of all tasks (*e.g.*, AUs) that reflects our prior beliefs regarding the related structure. A popular approach is to jointly learn the tasks under a regularization framework [64]. The regularization operates on the parameter space and penalizes distances between the different tasks, which results in uncovering a common set of parameters across the tasks. Hence, it allows to capture the similarities among the outputs through parameter sharing. Based on this idea, [1] introduced a manifold regularization approach to the multi-task learning. The key assumption is that the task parameters lie on a low dimensional manifold, and thus, they cannot vary arbitrarily. Instead of explicitly learning the manifold, the authors model the projection functions in a parametric formulation, and alternate between solving for the task parameters and minimizing their distances in the projected manifold. Similarly, [202] defines a latent variable model, which generates the task specific parameters in a probabilistic fashion. Due to its probabilistic formulation, several priors can be imposed on the latent variables to induce a desired structure to the task specific manifold.

The above methods rely on implicit assumptions that all tasks are related to each other. Contrary to this belief, [102] aims to uncover a structured pattern among the tasks, and combine them into different groups. Each task parameters are assumed to be a sparse, linear combination of underlying latent basic tasks. The overlap in the sparsity patterns of any two tasks controls the amount of sharing between them. In a similar fashion, [128] introduced the use of multi-output GP, for modeling task dependent regressors (latent functions) via GP priors. The output of each task is a weighted combination of a number of shared latent functions, which enables the collaboration among the tasks, plus an individual task-specific latent function. In order to deal efficiently with the problem of large number of output tasks and input data points, the authors derived a formulation based on variational inference.

Following a different approach, [7] used the notion of spectral graph regularization to jointly learn clusters of closely related tasks. Relationships between the tasks are defined in terms of the graph Laplacian, which favors similar tasks to be close in the parameter space. The authors proposed an alternating optimization algorithm based on proximity operators, in order to jointly learn the tasks and the graph. While applicable to the task of multiple AU detection, these methods do not perform simultaneous feature fusion and multi-label classification. By contrast, the proposed MC-LVM can be seen as a multi-task learning approach, where the relations of different tasks (*i.e.*, AUs) are learned directly in the shared subspace, by implicitly relating them through their feature and label dependencies. The latter are encoded by the local and global priors proposed in our model.

More recent works in the GP and multi-label classification context [185, 36] try to combine multi-task learning and feature fusion via subspace learning. [185] jointly optimizes latent variables in order to reconstruct the input data, and account for multiple tasks in the output. A downside of this method is that the learning of the latent space is achieved via MAP estimation, *i.e.*, the latent space is directly optimized during learning. In the case of large amount of data, this can easily lead to overfitting [192]. To ameliorate this, [36] proposed a fully Bayesian framework, based on variational inference, to integrate out the latent space.

In contrast to these methods, MC-LVM employs multi-conditional learning strategies to re-weight the generative and discriminative conditionals, in order to unravel a suitable subspace for joint feature fusion and multi-label classification. In our Bayesian approach, the latent space is approximated via an efficient Monte Carlo sampling, where the conditional models determine the importance of each sample. Moreover, the inference step is efficiently performed via the learned projection mappings to the manifold. This overcomes the requirement of [36] to learn another approximation to the posterior of the test inputs. Finally, note that none of these approaches have been evaluated in the task of multiple AU detection.

5.3.2 Multi-conditional Models and GPLVM

In the proposed MC-LVM, we employ the GP framework to derive a latent variable model with a joint distribution given by Eq. (5.1). We then introduce a set of conditional distributions (observed variables given latent positions $p(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{x})$, and latent positions given the observed data $p(\boldsymbol{x}|\boldsymbol{y})$) to form the multi-conditional objective function. The idea of multi-conditional learning has originally been explored in [123, 21]. However, these approaches are based on simple parametric conditional models and can deal with single-input single-output scenarios only. The proposed MC-LVM is a generalization of these approaches to multi-input multi-

output settings and non-parametric conditionals, modeled via GPs.

Modeling of the aforementioned conditionals in MC-LVM resembles the process we followed in Chapter 4 for the DS-GPLVM, and in general for various models that are based on the GPLVM [105]. Most of these models, as purely generative methods, try to model the joint likelihood

$$p(\boldsymbol{Y}, \boldsymbol{X}) = p(\boldsymbol{Y}|\boldsymbol{X})p(\boldsymbol{X}). \tag{5.18}$$

As we have already seen, the learning in these methods consists of maximizing the (marginal) log-likelihood of the joint given above, by following either a MAP optimization or variational approximations. By contrast, in MC-LVM we model the distribution of both observed inputs and latent variables by employing the predictive posterior of the GP. This results in learning a more robust mapping $\boldsymbol{x} \rightarrow \boldsymbol{y}$, and also allows us to efficiently estimate an instance of the latent space using the Monte Carlo sampling.

Finally, our proposed sampling distribution is closely related to the notion of 'back-constraints' in the GP literature. Recall from Chapter 3 that back-constraints were introduced in [106] as a deterministic, parametric mapping that pairs the latent variables of the GPLVM with the observations. This mapping facilitates a fast inference mechanism and enforces structure preservation in the manifold. The same mechanism has been used in Chapter 4 for making inference in the proposed DS-GPLVM. On the contrary, MC-LVM learns probabilistic mappings via the non-parametric GPs, which can result in latent projections, that are less prone to overfitting.

5.4 Experiments

In this section we evaluate the proposed MC-LVM on the joint task of feature fusion and multiple AU detection on data from both posed and spontaneous expressions.

5.4.1 Experimental Protocol

Datasets. We evaluate the proposed model on three publicly available datasets: Extended Cohn-Kanade (CK+) [114], UNBC-McMaster Shoulder Pain Expression Archive (Shoulder-pain) [116], and Denver Intensity of Spontaneous Facial Actions (DISFA) [122]. These are benchmark datasets of posed (CK+), and spontaneous (Shoulder-pain, DISFA) data, containing a large number of FACS coded AUs.

Figure 5.4: Example images with activated AUs from CK+ (top), DISFA (middle) and Shoulder-pain (bottom) datasets.

- The CK+ dataset [114] contains 593 video recordings of 123 subjects displaying posed facial expressions in near frontal views. The image sequences begin from neutral and proceed to the target expression. The last frame (peak frame) is annotated in terms of AU activations (presence/absence). For our experiments, we used the peak frames of all available subjects.

- The Shoulder-pain dataset [116] contains video recordings of 25 patients suffering from chronic shoulder pain while performing a range of arm motion tests. Each frame is coded in terms of AU intensity on a six-point ordinal scale.

- DISFA dataset [122] contains video recordings of 27 subjects while watching YouTube videos. Again, each frame is coded in terms of the AU intensity on a six-point ordinal scale.

For both DISFA and Shoulder-pain datasets, we treated each AU with intensity larger than zero as active. Sample images from the three datasets, along with examples of AUs present, are shown in Fig. 5.4. Fig. 5.5 depicts the AU relations, and the distribution of the AU activations for the data used from each dataset. Note that the co-occurrence patterns and the relations among the AUs differ significantly across all three datasets.

Table 5.1: Definitions of the used AUs from CK+, DISFA, and Shoulder-pain datasets.

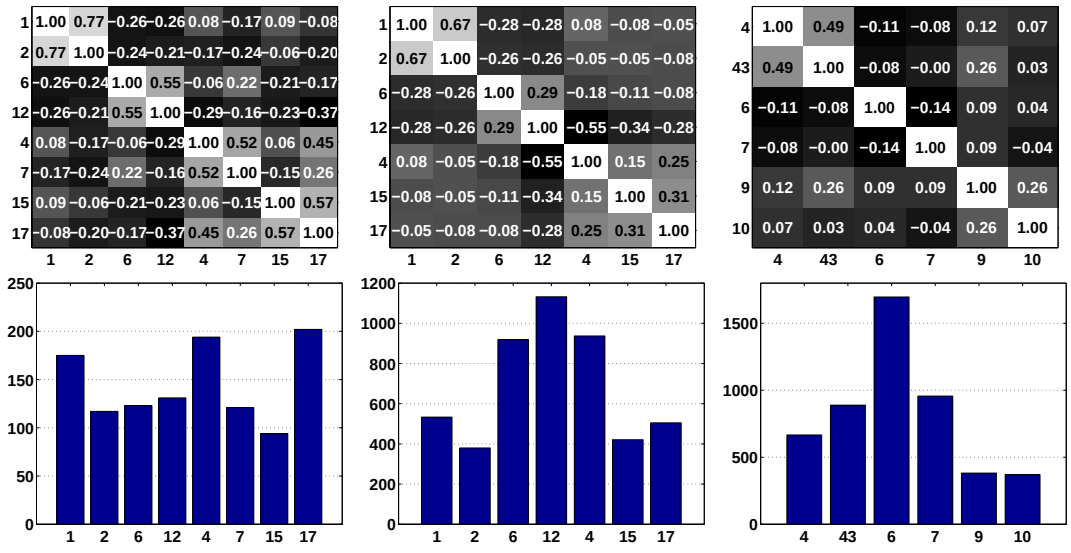
AU	Definition	AU	Definition	AU	Definition
1	Inner brow raiser	7	Lid tightener	15	Lip corner depress.
2	Outer brow raiser	9	Nose wrinkler	17	Chin raiser
4	Brow lowerer	10	Upper lip raiser	43	Eyes closed
6	Cheek raiser	12	Lip corner puller		

Features. In each frame of an input sequence 49 fiducial facial points were extracted using the 2D Active Appearance Model [120]. Based on these points, we registered the images to a reference face (average face for each dataset) using an affine transformation. As input to our model, we used both geometric features, *i.e.*, the registered facial points (feature set I), and appearance features, *i.e.*, local binary patterns (LBP) histograms [131] (feature set II) extracted around each facial point from a region of 32×32 pixels. We chose these features as they showed good performance in variety of AU recognition tasks [162]. To reduce the dimensionality of the extracted features we applied PCA, retaining 95% of the energy. This resulted in approximately 20D (geometric) and 40D (appearance) feature vectors, for each dataset.

Evaluation procedure. Some AUs occur rarely (*e.g.*, AU9,11,26 in CK+). Others do not exhibit strong co-occurrence patterns (*e.g.*, AU5 in DISFA). Hence, we selected the following subsets of highly correlated AUs: AUs (1, 2, 4, 6, 7, 12, 15, 17) for CK+, AUs (1, 2, 4, 6, 12, 15, 17) for DISFA and AUs (4, 6, 7, 9, 10, 43) for Shoulder-pain. The selected AUs occur jointly in the context of recorded expressions (*e.g.*, pain expression, see [116]).[3] In order to prove the model's ability to deal with large number of outputs, we also show the performance when all AUs (from CK+) are used. A detailed description of the AUs used for the model evaluation is shown in Table 5.1. We report the F1 score and the area under the ROC curve (AUC) as the performance measures. Both metrics are widely used in the literature as they quantify different characteristics of the classifier's performance. Specifically, F1, defined as $F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$, is the harmonic mean between the precision and recall. It puts emphasis on the classification task, while being largely robust to imbalanced data (such as examples of different AUs). AUC quantifies the relation between true and false positives, showing the robustness of a classifier to the choice of its decision threshold. In all our experiments, we performed a 5 fold subject independent cross-validation.

Models compared. We compare the proposed MC-LVM to GP methods with different learning strategy. Specifically, we compare to the manifold relevance determination

[3] We do not include the frequently occurring AU25 in our subsets because the associated action, *i.e.*, 'lips part', is also present during speech, and when this is the case it is not coded as active. Hence, a static model cannot determine whether the action happened during the speech or not.



(a) CK+ (b) DISFA (c) Shoulder-pain

Figure 5.5: The global AU relations (in terms of correlation coefficients) (upper row), and the distribution of AU activations within the datasets (lower row).

(MRD) [36], which uses the variational approximation, to the DS-GPLVM from Chapter 4 and multi-task latent GP (MT-LGP) [185], which perform MAP estimation. We also compare to the multi-label backpropagation and kNN (k=1), *i.e.*, the BPMLL [203] and ML-KNN [204], respectively. Lastly, we compare to the state-of-the-art methods for multiple AU detection: the parametric methods Bayesian group-sparse compressed sensing (BGCS) [171], hierarchical RBM (HRBM) [194], joint patch multi-label learning (JPML) [209], and the kernel method l_p-regularized multi-task MKL (l_p-MTMKL) [206]. All the compared methods are evaluated using the same previously described input features. Note that implementation of JPML [209] was not available, and thus, in our comparison we report the results from the corresponding paper ([209] employed the SIFT appearance descriptor). For the single input methods (*i.e.*, BGCS, HRBM, BPMLL and ML-KNN), we concatenated the two feature sets. For the kernel-based methods, we used the RBF kernel. For l_p-MTMKL we also used the polynomial kernel, as suggested in [206]. Due to the high learning complexity of l_p-MTMKL ($\mathcal{O}(N^2T^2)$), where T is the number of target AUs), we followed the training scheme in [206] where multiple AUs were split into groups: $\{\{AU1, AU2, AU4\}, \{AU6, AU7, AU12\}, \{AU15, AU17\}\}$ for CK+, the same groups (without AU7) for DISFA, and $\{\{AU4, AU43, AU7\}, \{AU6, AU9, AU10\}\}$ for Shoulder-pain. The parameters of each method were tuned as described in the corresponding papers. For the MC-LVM, optimal values for the weighting parameters α, the regularization parameters λ_C, λ_R, as well as the size of the latent space were found via a validation procedure on the training set.

5.4.2 MC-LVM: Theoretical Evaluation

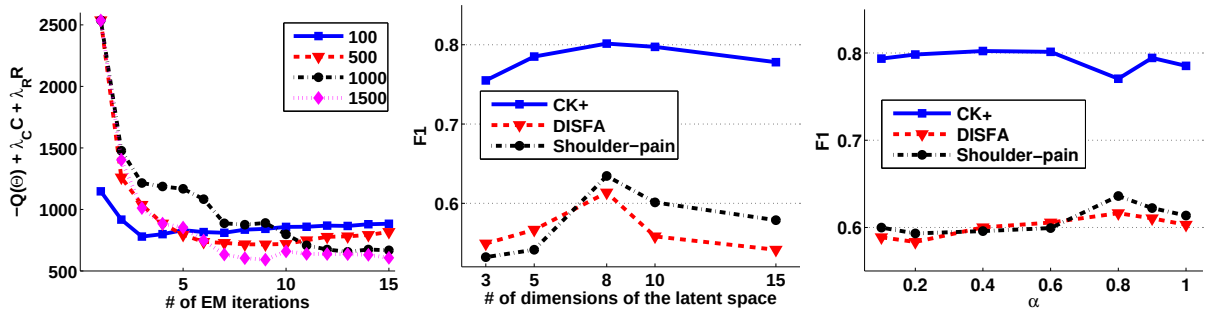


Figure 5.6: The penalized negative log-likelihood of the MC-LVM for different number of samples used to estimate the posterior of the latent space (left), and average F1 score for multiple AU detection as a function of the dimensionality of the latent space (middle), and the regularization parameter α (right).

This section analyzes MC-LVM performance in terms of different parameter choices and settings. Fig. 5.6 (left) shows the convergence of the learning criterion in MC-LVM as a function of the used Monte Carlo samples during training on the CK+ dataset. We see that for small number of samples, the model does not converge to a (local) minimum. This is expected, since with $100 - 500$ samples the posterior in Eq. (5.3) cannot be approximated well. The model converges when 1000 samples are used, and its convergence does not change considerably after that. Thus, we fixed the number of samples to 1000. From Fig. 5.6 (middle), we see how the size of the latent space affects the performance of the learned model. It is clear that for both posed and spontaneous data, an 8-dimensional latent space is sufficient for the task of joint feature fusion and multiple AU detection, and results in the best average F1-score. Lower dimensional manifolds fail to explain the correlations between the input features and to capture the dependencies among multiple AUs, while manifolds with more than 8D do not include any additional discriminative information. Hence, in what follows, we fixed the size of the latent space to 8D. Fig. 5.6 (right) shows the effect of changing α on the discriminative power of the model. We observe that the model prefers a weighted conditional distribution over a fully generative or discriminative component. The optimal value of α is around 0.4 for posed, and 0.8 for spontaneous data. This difference is due to the fact that in case of spontaneous data (DISFA, Shoulder-pain), the model puts less focus on explaining unnecessary variations for the AU detection task, *e.g.*, due to the subject-specific features and errors due to the pose registration. Therefore, the influence of the generative component is lower (higher α) than in the case of posed facial expressions from CK+. Moreover, the CK+ dataset contains significantly less data (around 600 annotated frames) than DISFA and Shoulder-pain. Hence, MC-LVM prioritizes the generative component, to avoid overfitting the training data. On the other hand, when we have sufficient training examples (DISFA,

Shoulder-pain), MC-LVM prefers to give less emphasis to the conditional distribution of the features (generative component). Such behavior of multi-conditional models has been also observed in other domains (e.g., in [97] for pixel classification).

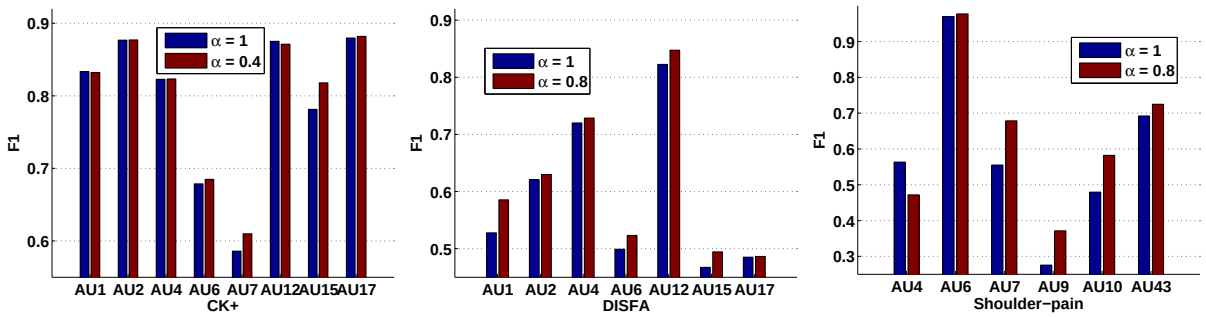


Figure 5.7: Joint AU detection with MC-LVM on CK+ (left), DISFA (middle) and Shoulder-pain (right) for different value of α. The comparisons are between the discriminative-only conditional ($\alpha = 1$) and the optimal weighted conditionals ($\alpha = 0.4$ for CK+ and $\alpha = 0.8$ for DISFA and Shoulder-pain) obtained after cross-validation of α.

To provide a better insight regarding the advantages of selecting a weighted conditional distribution, in Fig. 5.7 we compare the performance of the MC-LVM when the likelihood term consists of only the discriminative conditional ($\alpha = 1$), and the optimal weighted conditional ($\alpha = 0.4$ for CK+ and $\alpha = 0.8$ for DISFA and Shoulder-pain). We can see that the weighted conditional improves the performance on most of AUs, with significant enhancement in the performance on certain AUs (3% on AU7,15 on CK+, 6% on AU1 and 3% on AU6,15 on DISFA, and 10% on AU7,9,10 on Shoulder-pain).

In Fig. 5.8 (left) we see the effect of the introduced relational constraints on the model's performance. At first we observe that when no regularization is used ($\lambda_C, \lambda_R = 0$), MC-LVM achieves the lowest performance for both posed and spontaneous data. By including only the topological constraint ($\lambda_C \neq 0, \lambda_R = 0$), MC-LVM attains a better representation of the

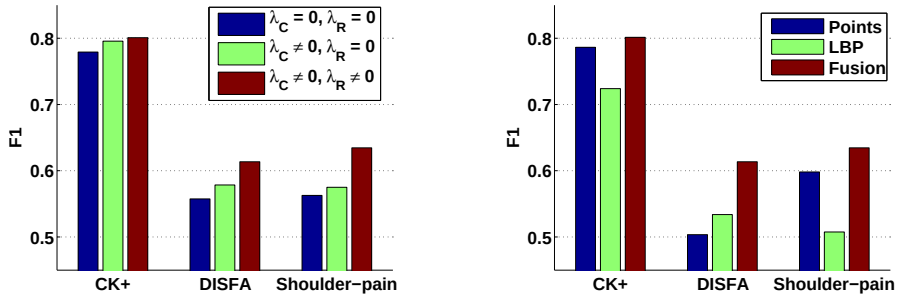


Figure 5.8: Average F1 score on all three datasets. The effect of the relational constraints (left), and the feature fusion (right) on the joint AU detection task.

data in the manifold, which results in higher F1 scores. Finally, with the addition of the global relational constraint ($\lambda_C, \lambda_R \neq 0$) MC-LVM achieves the highest scores. Note that the difference is more pronounced in data from DISFA and Shoulder-pain. This evidences the importance of modeling the global relations for the detection of spontaneous (more subtle) AUs. This is because the features of these AUs are corrupted by higher noise levels and thus, their joint prediction can help to reduce uncertainty of the classifiers, as has been reported in [121]. Fig. 5.8 (right) shows the average performance of the model for different feature combinations. In the single input case, we observe that, on average, geometric features (I) outperform the appearance features (II) (apart from DISFA where features (I) suffer from residual errors from the pose registration due to large variations in the head pose). This is because, by concatenating the LBP histograms obtained from each patch, the local information of the data is lost, and thus, the model obtains lower scores. However, when both inputs are used, MC-LVM can unravel a very informative shared latent space. This results in the highest F1 score, with significant improvement on the spontaneous data of DISFA and Shoulder-pain. In general, from Fig. 5.8 we see that the effect of the introduced regularization and the feature fusion is far more pronounced in the case of spontaneous facial expressions, where a limited and imbalanced number of examples of each AU is available (*e.g.*, AU1,2,15,17 for DISFA, and AU4,9,10 for Shoulder-pain).

5.4.3 Model Comparisons on Posed Data

We next compare the proposed MC-LVM to several state-of-the-art methods on the posed data from CK+. We first inspect the performance of MC-LVM and the GP-related methods. From Table 5.2, we can see that the MAP-based methods, *i.e.*, the MT-LGP [185] and DS-GPLVM, achieve similar performance on average since they are based on the same learning scheme. On the other hand, MRD [36], uses a variational distribution to approximate a manifold shared across multiple inputs and outputs, without any additional constraints over the latent variables. This results in a poor accuracy. Also, MRD learns an approximation to the posterior, in order to predict the variational latent positions that best generate the inputs, while MT-LGP and DS-GPLVM learn accurate back mappings from the input spaces to the manifold. By contrast, the combination of the approximate learning with the relational constraints used in the proposed MC-LVM results in a significant increase in performance over the GP-based methods. We partly attribute this to the explicit modeling of AU co-occurrences through the introduced constraints, as well as to the multi-conditional learning based on the proposed sampling scheme. The importance of the latter is further evidenced in the performance of the single output instance of MC-LVM, which for the case of the posed data

Table 5.2: F1 score (a) and AUC (b) for joint AU detection on CK+ dataset. Comparisons to state-of-the-art.

(a)

Methods (I+II)	F1 score								
	AU1	AU2	AU4	AU6	AU7	AU12	AU15	AU17	Avg.
MC-LVM	84.39	86.55	81.60	68.42	61.67	**88.48**	**82.54**	**87.40**	**80.14**
MC-LVM (SO)	86.06	**88.37**	**82.93**	70.80	57.27	87.16	73.26	85.57	78.93
MRD [36]	80.72	79.18	69.93	69.81	53.24	77.83	65.70	85.20	72.70
MT-LGP [185]	**89.12**	83.70	79.79	67.16	60.89	80.53	64.63	85.97	76.47
DS-GPLVM	87.41	81.78	79.70	68.48	63.29	81.04	60.33	84.29	76.17
BGCS [171]	84.57	86.19	81.17	69.82	59.48	87.77	74.77	84.84	78.58
HRBM [194]	87.62	84.00	74.10	62.90	50.74	82.38	66.06	84.56	74.04
l_p-MTMKL [206]	87.50	85.50	51.43	**72.65**	58.82	85.95	74.21	75.44	73.93
BPMLL [203]	75.41	84.31	64.85	69.14	**64.34**	83.98	69.50	76.25	73.47
ML-KNN [204]	76.83	84.34	63.28	67.23	53.19	82.88	65.88	78.71	71.54
JPML* [209]	91.2	96.5	-	75.6	50.9	80.4	76.8	80.1	78.8

(b)

Methods (I+II)	AUC								
	AU1	AU2	AU4	AU6	AU7	AU12	AU15	AU17	Avg.
MC-LVM	95.66	96.80	**93.97**	92.07	87.84	**97.78**	**94.60**	**96.10**	**94.35**
MC-LVM (SO)	**98.22**	97.25	93.95	92.20	85.71	97.41	94.05	95.80	94.33
MRD [36]	95.58	92.53	91.85	**92.73**	82.69	94.50	91.32	94.78	92.00
MT-LGP [185]	96.70	**97.33**	90.90	91.45	86.37	96.92	94.25	94.80	93.59
DS-GPLVM	96.10	96.69	89.56	89.83	85.91	95.69	92.56	94.03	92.55
BGCS [171]	97.76	96.63	93.21	91.59	85.06	97.69	94.04	95.43	93.85
HRBM [194]	95.99	95.13	88.00	88.37	78.09	93.73	93.49	95.60	91.05
l_p-MTMKL [206]	93.19	94.99	90.95	90.01	84.41	95.67	91.06	92.97	91.65
BPMLL [203]	89.06	95.21	76.88	90.53	85.51	95.48	90.20	88.19	88.88
ML-KNN [204]	89.07	95.54	76.46	90.58	**90.71**	94.31	92.65	89.13	89.81
JPML* [209]	-	-	-	-	-	-	-	-	-

achieves comparable scores to the multi-output. We see that joint learning does not improve detection of all AUs. It even shows reduced performance for certain AUs. For example, from Fig. 5.5, we see that AU1,2 are strongly correlated, yet single output achieves higher F1 on both AUs compared to the multi-output setting. This shows that for given data, these two AUs can be predicted well without relying on each other. On the other hand, the performance of AU15, which is strongly correlated with AU17, and has significantly less examples than other AUs, is considerably improved (F1 9% higher). The similar performance between the two settings is also explained from the nature of the posed data of CK+. Joint AU learning is expected to be advantageous, in cases where the input data suffer from high-dimensional noise [121]. Hence the superior performance of the multi-output setting will be evidenced in the evaluations on the spontaneous data from DISFA and Shoulder-pain in Sec. 5.4.4.

Table 5.3: F1 score for joint AU detection (all 17) on CK+ dataset. Comparison to state-of-the-art.

(a)

Methods (I+II)	AU1	AU2	AU4	AU5	AU6	AU7	AU9	AU11	AU12
MC-LVM	82.49	**86.96**	**79.16**	**73.47**	**72.80**	**57.52**	87.94	**31.11**	**87.60**
BGCS [171]	83.04	85.10	77.45	72.21	69.26	55.94	**89.03**	29.41	86.79
HRBM [194]	**86.86**	85.47	72.58	72.04	61.74	54.47	85.91	26.51	72.65

(b)

Methods (I+II)	AU15	AU17	AU20	AU23	AU24	AU25	AU26	AU27	Avg.
MC-LVM	**76.40**	**86.76**	70.27	67.27	**51.02**	91.81	21.05	**91.14**	**71.45**
BGCS [171]	74.92	83.33	**71.10**	**68.01**	48.14	76.60	34.21	88.55	70.12
HRBM [194]	72.53	81.66	47.46	56.64	35.29	**92.57**	**37.61**	87.65	66.45

Table 5.2, also summarizes the performance of the state-of-the-art models for joint AU detection: BGCS, HRBM and l_p-MTMKL. These models, manage to improve the detection of AU1 and AU6, by successfully modeling their co-occurrences between the related AUs (AU2 and AU12 respectively) in the expressions of Surprise and Happiness. However, their performance on more subtle AUs, *e.g.*, AU7,15,17 is significantly lower than that of the proposed MC-LVM. This is due to the fact that the parametric models BGCS and HRBM cannot handle simultaneously the fusion of the *concatenated* features and the modeling of the AU dependencies using compressed/binary latent variables. On the other hand, l_p-MTMKL can perform the fusion through the MKL framework. However, due to its modeling complexity, it is trained on subsets of AUs, which affects its ability to capture all AU relations. More importantly, in contrast to MC-LVM, these models lack the generative component, which, evidently, acts as a powerful regularizer. The results of JPML were obtained from [209], thus, they are not directly comparable to the other models. Yet, we report this performance as a reference to the state-of-the-art. Finally, the baseline multi-label methods, BPMLL and ML-KNN attempt to model the AU dependencies directly in the classifier level, as in l_p-MTMKL, but they cannot perform the fusion of the input features. Hence, they achieve the lowest average scores.

To demonstrate the model's scalability when dealing with large number of outputs, we compare the proposed approach to the state-of-the-art HRBM and BGCS for joint AU detection on *all* 17 AUs from CK+ (l_p-MTMKL cannot be evaluated on this experiment due to its learning complexity). As we can see from Table 5.3, modeling the remaining (less frequently occurring) AUs affects the overall performance of all three models, *i.e.*, MC-LVM, BGCS and HRBM, which suffer a drop of 8.6%, 8.4% and 7.6%, respectively. However, MC-LVM outperforms HRBM on 14 out of 17 AUs and BGCS on 12 out of 17 AUs, which demonstrates the ability

Table 5.4: F1 score and AUC for joint AU detection on the DISFA dataset. Comparisons to the state-of-the-art.

Methods (I+II)	F1 score								AUC							
	AU1	AU2	AU4	AU6	AU12	AU15	AU17	Avg.	AU1	AU2	AU4	AU6	AU12	AU15	AU17	Avg.
MC-LVM	**58.55**	**62.99**	**72.85**	52.32	**84.74**	**49.44**	**48.63**	**61.36**	**79.58**	84.01	**84.87**	62.75	**92.43**	78.97	**73.87**	**79.50**
MC-LVM (SO)	35.50	52.68	70.99	54.67	82.58	37.11	47.76	54.47	64.71	85.21	82.52	**68.15**	92.20	**79.22**	72.39	77.77
MT-LGP [185]	41.44	36.84	61.19	45.98	49.78	40.12	43.01	45.48	69.28	79.31	74.23	62.08	70.22	58.61	67.69	68.27
BGCS [171]	50.13	36.49	72.05	59.64	78.47	39.93	40.29	53.86	69.54	49.72	78.93	66.76	86.55	73.67	63.36	69.79
HRBM [194]	39.67	55.92	61.56	54.01	79.16	38.72	38.82	52.55	61.55	**85.88**	67.10	58.08	81.74	64.93	64.41	69.10
l_p-MTMKL [206]	42.21	45.81	47.18	**62.79**	76.33	34.47	41.40	50.03	71.77	73.42	62.49	66.27	78.83	59.16	63.98	67.98

of the former to better model the relations among AUs, even in case of many AU classes.

5.4.4 Model Comparisons on Spontaneous Data

Table 5.5: F1 score and AUC for joint AU detection on the Shoulder-pain dataset. Comparisons to the state-of-the-art.

Methods (I+II)	F1 score							AUC						
	AU4	AU6	AU7	AU9	AU10	AU43	Avg.	AU4	AU6	AU7	AU9	AU10	AU43	Avg.
MC-LVM	47.20	**97.75**	67.88	37.13	58.23	**72.51**	**63.45**	53.58	**82.27**	57.80	54.65	87.80	66.13	**67.04**
MC-LVM (SO)	57.76	95.57	63.59	34.54	49.93	64.49	60.98	**66.36**	50.47	60.04	53.23	64.20	65.81	60.02
MT-LGP [185]	50.42	50.48	63.52	33.38	**61.62**	61.00	53.40	61.35	44.40	60.96	52.47	**90.39**	60.90	61.75
BGCS [171]	**61.42**	71.52	60.40	**37.86**	54.50	63.49	58.20	63.28	59.29	59.93	**59.23**	69.96	67.10	63.13
HRBM [194]	47.20	93.93	63.67	29.80	52.39	69.54	59.42	57.33	77.41	62.56	53.21	71.36	**73.19**	65.85
l_p-MTMKL [206]	37.69	**97.75**	**70.08**	33.28	41.79	44.03	54.10	54.95	71.86	**64.15**	53.84	68.62	64.69	63.01

We further investigate the models' performance on spontaneous data from DISFA and Shoulder-pain datasets. We focus here on the best performing methods from Table 5.2. From Tables 5.4–5.5, we can observe a significant drop in the performance of all methods on both datasets. This evidences the difficulty of the task of AU detection in realistic environments, where spontaneous expressions are present. Also, typical for naturalistic data, the distribution of the activated AUs is more imbalanced than in the case of the posed dataset. This poses an additional modeling challenge since training data for certain AUs (*e.g.*, AU2,15 for DISFA, and AU9,10 for Shoulder-pain) are limited. Consequently, the models need to put more emphasis on the AU co-occurrences for detection of these AUs. As evidenced by the results in Tables 5.4–5.5, this adversely affects the single output MC-LVM. Contrary to the high achieved performance on the posed data, the single output instance reports here significantly lower scores for the aforementioned AUs in both datasets. Furthermore, the small amount of training data for some AUs, imposes an additional difficulty when modeling the global AU relations. Consequently, the parametric discriminative models, BGCS and HRBM, overfit the data and report low performance. This exemplifies the importance of modeling the relations among the features via the generative component, in the proposed approach. Note that for some AUs with sufficient training data, *e.g.*, AU4,6 in DISFA, BGCS and HRBM achieve

Table 5.6: Cross-dataset evaluations of the state-of-the-art models on 7 AUs present in both CK+ and DISFA datasets. The models are trained on data from CK+ dataset and tested on data from DISFA dataset (C→D), and the other way around (D→C).

(a)

Train→Test	Methods (I+II)	F1 score							
		AU1	AU2	AU4	AU6	AU12	AU15	AU17	Avg.
C→D	MC-LVM	53.92	**54.69**	**68.37**	51.99	70.77	**37.14**	42.81	54.24
	BGCS [171]	**59.01**	49.37	68.34	**57.75**	**80.26**	36.59	43.54	**56.41**
	HRBM [194]	43.20	36.83	52.10	36.15	40.70	35.61	**51.13**	42.25
	l_p-MTMKL [206]	39.13	41.24	44.77	49.42	69.67	31.55	39.12	44.98
D→C	MC-LVM	**72.22**	**85.85**	**75.05**	59.94	63.45	54.81	**73.35**	**69.24**
	BGCS [171]	61.11	71.90	67.84	**65.05**	**80.46**	54.23	69.98	67.22
	HRBM [194]	66.81	64.52	60.12	54.11	65.60	**60.47**	66.67	62.61
	l_p-MTMKL [206]	68.10	61.94	56.06	57.86	66.26	43.30	63.66	59.60

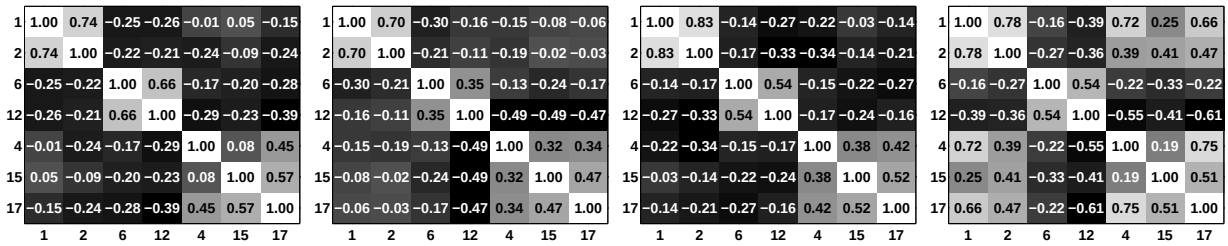
(b)

Train→Test	Methods (I+II)	AUC							
		AU1	AU2	AU4	AU6	AU12	AU15	AU17	Avg.
C→D	MC-LVM	76.78	86.80	**79.74**	**73.21**	86.73	62.28	67.83	76.20
	BGCS [171]	**86.75**	91.75	78.97	69.97	**87.83**	64.83	69.67	**78.54**
	HRBM [194]	67.41	71.84	65.62	59.32	62.62	60.77	**74.05**	65.95
	l_p-MTMKL [206]	71.77	73.42	72.70	68.38	67.46	69.31	65.85	65.56
D→C	MC-LVM	**92.51**	**96.60**	**90.51**	84.24	**95.02**	87.21	90.82	**90.99**
	BGCS [171]	84.44	91.21	88.21	84.91	94.54	84.12	84.97	87.49
	HRBM [194]	88.88	92.26	81.47	**88.23**	94.19	**87.91**	**91.61**	89.22
	l_p-MTMKL [206]	80.21	82.41	69.45	79.59	86.28	74.64	78.88	78.78

similar or better scores than the MC-LVM. This is in part due to modeling the multiple AU detectors under a joint cost function – each method selects to put more emphasis on modeling different AUs than the others. However, the MC-LVM outperforms these models on average. l_p-MTMKL obtains very low scores (especially in the Shoulder-pain), which is a result of not modeling global relations, due to its training scheme. MT-LGP also fails to model explicitly the relations between AUs, achieving low scores as well. The proposed MC-LVM is more robust to the data imbalance, and can better discover the AU relations, which in turn gives not only the best average F1 scores, but also achieves more robust performance as evidenced by the higher AUC.

5.4.5 Cross Dataset Experiments on CK+ and DISFA

In this section, we evaluate the robustness of the models in a cross dataset experiment, in order to assess the generalizability of each model when dealing with new instances obtained under different settings. Specifically, we perform two different cross-dataset experiments,



(a) CK+→CK+ (b) DISFA→DISFA (c) DISFA→CK+ (d) CK+→DISFA

Figure 5.9: The learned global AU relations (in terms of correlation coefficients) for within datasets (a),(b) and cross-datasets (c),(d) experiments.

CK+→DISFA and DISFA→CK+.[4] We evaluate the models' performance on 7 AUs (*i.e.*, $1, 2, 4, 6, 12, 15, 17$) that are present in both datasets. This is a rather challenging task due to the different characteristics of the data. First of all, as shown in Fig. 5.4, the facial images differ in terms of illumination, pose and size, which imposes a further difficulty on the alignment of the input facial features. Another key challenge is the difference in the context of the two datasets. The data from CK+ contain posed expressions, which vary considerably in subtlety compared to the spontaneous data of DISFA. The latter also affects the co-occurrence patterns among the AUs, as can be seen from Fig. 5.5.

From Table 5.6, we see that the performance of the models is lower for most of AUs compared to that attained on the original dataset (see Tables 5.2-5.5). This is expected for the reasons mentioned above. Interestingly, BGCS achieves higher performance on the cross dataset experiment CK+→DISFA, than when both training and testing is performed on DISFA dataset. This confirms our claims in Section 5.4.4 that this method cannot fully unravel the dependencies among the AUs when dealing with imbalanced data in the training phase. The parametric model, *i.e.*, BGCS, can better model the AU relations with small (but well distributed) amount of training data, as in CK+. Hence, it achieves higher performance compared to MC-LVM. However, on the DISFA→CK+ experiment, we see that the proposed MC-LVM, benefits from the use of the non-parametric feature fusion, and manages to successfully unravel the structure and the co-occurrence patterns in the data, regardless of the imbalances in the amount of training examples and the subtlety of the spontaneous facial expressions. Thus, it attains superior performance compared to the BGCS, especially for AU1,2,4, where the two models achieve similar predictions for training and testing on CK+ (see Table 5.2). Finally, the proposed MC-LVM consistently outperforms HRBM and l_p-MTMKL on both cross-dataset experiments, as evidenced from both F1 and AUC results.

[4] 'A→B' denotes the training on dataset A and testing on dataset B.

Finally, in Fig. 5.9 we see the recovered AU dependencies from the MC-LVM, on the test data in both within and cross-dataset experiments. As we observe from Fig. 5.9(a) and Fig. 5.9(c), the recovered AU dependencies for CK+ are similar to the original co-occurrence patterns from Fig. 5.5. Hence, the proposed MC-LVM attains competitive results for CK+ and the DISFA→CK+ experiments. On the other hand, by comparing Fig. 5.9(d) to Fig. 5.9(b) and Fig. 5.5, we observe that MC-LVM has falsely recovered strong correlations between AU1,2 and AU15,17, which results in the low performance in the CK+→DISFA experiment. We attribute this to the fact that AU1,4,17, as we can see in Fig. 5.5, are the dominant AUs in CK+, which is not the case for DISFA. Thus the model trained on CK+ seems to have a bias on predicting AU1,4,17. Due to their strong relations with AU2,15 MC-LVM recovers the false dependencies on DISFA dataset.

5.5 Conclusions

To conclude, in this chapter we proposed the multi-conditional latent variable model that brings together GPs and multi-conditional learning to achieve a feature fusion for multi-label classification of facial AUs. The majority of existing approaches perform feature fusion via simple vector concatenation. However, this leads to the false assumption that the multiple feature sets are identically distributed. By assuming conditional independence given the subspace of AUs, MC-LVM learns different distributions for each feature set via separate GPs, resulting in more accurate fusion in the manifold, and hence, more discriminative features for the detection task. More importantly, the newly introduced multi-conditional objective allows the generative and discriminative costs to act in concert during the model learning – the generative component has the key role in unraveling the latent space for the feature fusion, while the discriminative component endows the space with the relational/class information of the outputs. Consequently, the proposed model learns a discriminative manifold structure that is regularized by the amount of shared information between the input features. The retrieved manifold, which is a trade-off between the generative/discriminative components, leads to superior performance compared to other solely discriminative or generative approaches. We further proved that the novel *topological* and *relational* constraints can increase the discriminative power of the model, by successfully encoding the AU dependencies into the learned manifold. We demonstrated the effectiveness of these properties on three publicly available datasets, and showed that the proposed model outperforms the existing works for multiple AU detection, and several methods for feature fusion and multi-label learning. We also showed that the proposed model is able to generalize across different contexts (datasets), however,

with reduced performance.

Chapter 6

Gaussian Process Auto-encoders for Joint Action Unit Intensity Estimation

Contents

6.1 Introduction

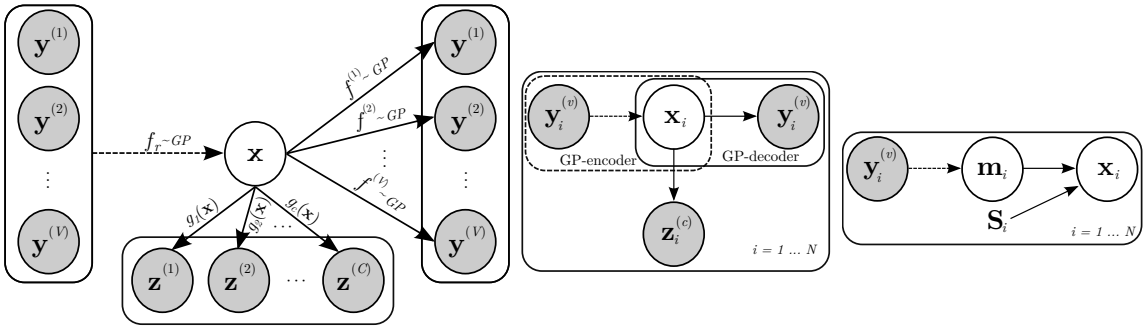
To date, most existing work on automated analysis of facial expressions, including the MC-LVM from the previous chapter, focuses on the detection of AU activations, *i.e.*, presence/absence of an AU. The problem of AU intensity estimation is relatively new in the field. Most of the research in this area focuses on independent modeling of AU intensities, and cast the problem as a classification [151, 118, 122, 125, 186] or regression [158, 92, 89, 93] task, which is a sub-optimal modeling practice, given the ordinal nature of the output labels. Similarly, the models that do attempt multiple AU intenisty estimation (*e.g.*, [109, 156, 94, 129, 126]) adopt the same sub-optimal approach to deal with the nature of the output as the independent methods. Furthermore, they do not exploit potential correlations among different type of input features. Hence, they cannot fully benefit from the joint modeling of AU co-occurrences. Apart from a few exceptions that treat each AU independently [158, 92, 125], none of the aforementioned approaches successfully addresses the task of joint output modeling (*i.e.*, mul-

tiple AUs) while accounting for different modalities in the input (*i.e.*, fusion of geometric and appearance features). These limitations can naturally be addressed by following recent advances in manifold learning [36, 185, 23] and, in particular, using the framework GPs [146]. As we have presented in Chapters 4&5, within this framework, we can transform the problem of feature fusion to that of learning from multiple views, while continuous-valued predictions can be handled efficiently, for more than one output. However, as with the regression-based models described above, these models treat the ordinal labels as continuous values. This also limits their potential to unravel an 'ordinal' manifold, needed to facilitate estimation of target ordinal intensities.

In this chapter, we propose a novel manifold-based GP approach based on the Bayesian GP latent variable model (B-GPLVM) [178] that performs simultaneously the feature fusion and joint estimation of the AU ordinal intensity. Specifically, we propose the variational GP auto-encoder (VGP-AE), which is composed of a probabilistic *recognition* model, used to project the observed features onto the manifold, and a generative model, used for their reconstruction. Our probabilistic recognition model, contrary to our previous defined DS-GPLVM from Chapter 4 that learns deterministic parametric back-mappings, allows us to explicitly model the uncertainty in the projections onto the learned manifold and propagate it to the final predictions. Compared to the MC-LVM from Chapter 5, which also employs the GPs for the back-mappings, in this chapter we propose an optimization scheme in order to learn both the latent space and the recognition model in a single pass, without the requirement of alternating between learning the two. Furthermore, we endow the proposed VGP-AE with ordinal outputs [2]. The fusion of the information from the input features and learning of the joint ordinal output is performed simultaneously in a joint Bayesian framework. In this way, we seamlessly integrate the ordinal structure into the recovered manifold while attaining robust fusion of the target features. To the best of our knowledge, this is the first approach that achieves simultaneous feature fusion and joint AU intensity estimation in the context of facial behavior analysis. Note that the contents of this chapter are published in [58].

6.2 Variational Gaussian Process Auto-Encoder

Similarly to Chapter 5, we assume that we have access to a training data set $\mathcal{D} = \{\boldsymbol{Y}, \boldsymbol{Z}\}$, which is comprised of V observed input channels $\boldsymbol{Y} = \{\boldsymbol{Y}^{(v)}\}_{v=1}^{V}$, and the associated output labels $\boldsymbol{Z}$. Each input channel consists of N i.i.d. samples $\boldsymbol{Y}^{(v)} = \{\boldsymbol{y}_i^{(v)}\}_{i=1}^{N}$, where $\boldsymbol{y}_i^{(v)} \in \mathbb{R}^{D_v}$ denotes corresponding facial features. $\boldsymbol{Z} = \{\boldsymbol{z}_i\}_{i=1}^{N}$ is the common label representation, where $z_{ic} \in \{1, \dots, S\}$ denotes the discrete, ordinal state of the c-th output (*i.e.*, AU intensity



(a) Graphical model of VGP-AE (b) Plate diagram (c) Recognition model

Figure 6.1: The proposed VGP-AE. (a) $f^{(v)}$ and f_r are the GP-decoder and GP-encoder, respectively. The projection of the latent variable $\boldsymbol{x}$ to the labels' ordinal plane is facilitated through the ordinal regression $g(\boldsymbol{x})$. (b) Compact representation of the model. (c) The proposed recognition model (GP-encoder) with the intermediate variable $\boldsymbol{m}$.

level), $c = 1, \ldots, C$. We are interested in simultaneously addressing the tasks of feature fusion and ordinal prediction of the multiple outputs. For this purpose, we propose an approach that resembles recent work of generative models [98, 147]. In these models, auto-encoders are employed to learn compact representations of the input data. In a standard auto-encoding setting, the encoding/decoding functions are modeled via neural networks. Here we replace these functions with probabilistic non-parametric mappings, significantly reducing the number of optimized parameters, and naturally modeling the uncertainty in the mappings. The proposed approach can be regarded as a B-GPLVM (generative model) with a fast inference mechanism based on the non-parametric, probabilistic mapping (recognition model). To achieve this, we impose GP priors on both models, and hence, obtain a well-defined GP-*encoder*, in accordance to the GP-*decoder*.

6.2.1 The Model

Within the above setting, we assume that the observed features $\boldsymbol{Y}^{(v)}$ are generated by a random process, involving a latent (unobserved) set of variables $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^{N}, \boldsymbol{x}_i \in \mathbb{R}^q$, with $q \ll D_v$. The data pairs $\mathcal{D} = \{\boldsymbol{Y}, \boldsymbol{Z}\}$ are assumed to be conditionally independent given the latent variables, *i.e.*, $\boldsymbol{Y} \perp\!\!\!\perp \boldsymbol{Z}|\boldsymbol{X}$. The random process of recovering the latent variables has two distinctive stages: (a) a latent variable $\boldsymbol{x}_i$ is generated from some general prior distribution $p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, and further projected to the labels' ordinal plane via $p(\boldsymbol{z}|\boldsymbol{x})$; (b) an observed input $\boldsymbol{y}_i^{(v)}$ is generated from the conditional distribution $p(\boldsymbol{y}^{(v)}|\boldsymbol{x})$. This process is described in Fig. 6.1(a),(b). Using this approach, we can now perform classification in the lower-dimensional space of $\boldsymbol{X}$. However, this requires access to the intractable true posterior $p(\boldsymbol{x}|\boldsymbol{y}^{(v)})$.

To constrain the distribution of the latent variables we follow [98, 147] and introduce the *recognition* model $p_r(\boldsymbol{x}|\boldsymbol{y}^{(v)})$. Hence, we end up with a supervised auto-encoder setting

$$\boldsymbol{y}_i^{(v)}|\boldsymbol{x}_i = f^{(v)}(\boldsymbol{x}_i;\boldsymbol{\theta}^{(v)}) + \epsilon^{(v)}, \quad \boldsymbol{x}_i|\boldsymbol{y}_i^{(v)} = f_r(\boldsymbol{y}_i^{(v)};\boldsymbol{\theta}_r) + \epsilon_r, \quad \boldsymbol{z}_i|\boldsymbol{x}_i = g(\boldsymbol{x}_i;\boldsymbol{W}), \tag{6.1}$$

where the latent space is further encouraged to reflect the structure of the output labels. Here, $\epsilon^{(v)} \sim \mathcal{N}(\mathbf{0}, \sigma_v^2\boldsymbol{I})$, $\epsilon_r \sim \mathcal{N}(\mathbf{0}, \sigma_r^2\boldsymbol{I})$. We place GP priors on $f^{(v)}, f_r$ with corresponding hyper-parameters $\boldsymbol{\theta}^{(v)}, \boldsymbol{\theta}_r$.[1] Here, g denotes the ordinal regression that transforms the latent variables to the labels' ordinal plane, via $\boldsymbol{W} = \{\boldsymbol{w}_c\}_{c=1}^{C}, \boldsymbol{w}_c \in \mathbb{R}^q$.

In the following, we detail how to learn the GP auto-encoder in Eq. (6.1) by deriving a variational approximation to the log-marginal likelihood

$$\log p(\boldsymbol{Y}, \boldsymbol{Z}) = \log \int p(\boldsymbol{Z}|\boldsymbol{X}) \prod\nolimits_v p(\boldsymbol{Y}^{(v)}|\boldsymbol{X}) p(\boldsymbol{X}) d\boldsymbol{X}. \tag{6.2}$$

6.2.2 Deriving the Lower Bound

We exploit the conditional independence property of $\boldsymbol{Y} \perp\!\!\!\perp \boldsymbol{Z}|\boldsymbol{X}$ and focus our analysis on the GP auto-encoder. The ordinal information from the labels is incorporated in the presented variational framework in Sec. 6.2.3. We follow the analysis from Chapter 5 for the MC-LVM and place GP priors on $f^{(v)}, f_r$. After integrating out the mapping functions, we obtain the conditionals

$$p(\boldsymbol{Y}^{(v)}|\boldsymbol{X}) = \mathcal{N}(\mathbf{0}, \boldsymbol{K}^{(v)} + \sigma_v^2\boldsymbol{I}), \qquad p_r(\boldsymbol{X}|\boldsymbol{Y}) = \mathcal{N}(\mathbf{0}, \boldsymbol{K}_r + \sigma_r^2\boldsymbol{I}), \tag{6.3}$$

where $\boldsymbol{K}^{(v)} = k^{(v)}(\boldsymbol{X}, \boldsymbol{X})$ and $\boldsymbol{K}_r = \sum_v k_r^{(v)}(\boldsymbol{Y}^{(v)}, \boldsymbol{Y}^{(v)})$ are the kernels associated with each process. Note that in the recognition model the relevant kernel allows us to easily combine multiple features via the sum of the individual kernel functions. Training of the recognition model consists of maximizing the conditional $p_r(\boldsymbol{X}|\boldsymbol{Y})$ w.r.t. the kernel hyper-parameters $\boldsymbol{\theta}_r$. For the generative model we maximize the marginal likelihood (labels $\boldsymbol{Z}$ are omitted here)

$$p(\boldsymbol{Y}) = \int \prod\nolimits_{v=1}^{V} p(\boldsymbol{Y}^{(v)}|\boldsymbol{X}) p(\boldsymbol{X}) d\boldsymbol{X}. \tag{6.4}$$

Since the above integral is intractable, we resort to approximations. Our main interest is to recover a Bayesian non-parametric solution for both the GP encoder and decoder. We first need to break the circular dependence between $\boldsymbol{Y}^{(v)}$ and $\boldsymbol{X}$ in order to train the two GPs simultaneously.

[1] The subscript r indicates that the process facilitates the recognition model.

GP-encoder. We decouple $\boldsymbol{X}$ and $\boldsymbol{Y}$ by introducing an intermediate variable $\boldsymbol{M} = \{\boldsymbol{m}_i\}_{i=1}^N$, so that the recognition model becomes $\boldsymbol{y}^{(v)} \rightarrow \boldsymbol{m} \rightarrow \boldsymbol{x}$. The GP operates on $\boldsymbol{y}^{(v)}, \boldsymbol{m}$, while $\boldsymbol{x}$ is the noisy observations of $\boldsymbol{m}$. This process is described in Fig. 6.1(c). We follow a mean field approximation and introduce the variational distribution $q(\boldsymbol{X}|\boldsymbol{M}) = \prod_i q_i(\boldsymbol{x}_i|\boldsymbol{m}_i) = \prod_i \mathcal{N}(\boldsymbol{m}_i, \boldsymbol{S}_i)$. Here, $\boldsymbol{m}_i, \boldsymbol{S}_i \in \mathbb{R}^q$ are variational parameters[2] of q_i. We define $\boldsymbol{M}$ by employing the cavity distribution of the leave-one-out solution of GP [146]

$$p(\boldsymbol{M}|\boldsymbol{Y}) = \prod\nolimits_i p(\boldsymbol{m}_i|\boldsymbol{Y}, \boldsymbol{M}_{\setminus i}) = \prod\nolimits_i \mathcal{N}(\hat{\boldsymbol{m}}_i, \hat{\sigma}_i^2 \boldsymbol{I}), \tag{6.5}$$

where the subscript $\setminus i$ means 'all datapoints except i', and the mean and variance of the Gaussian are given by [146]

$$\hat{\boldsymbol{m}}_i = \boldsymbol{m}_i - \left[\boldsymbol{K}_r^{-1}\boldsymbol{M}\right]_i / \left[\boldsymbol{K}_r^{-1}\right]_{ii}, \qquad \hat{\sigma}_i^2 = 1/\left[\boldsymbol{K}_r^{-1}\right]_{ii}. \tag{6.6}$$

We now integrate out the intermediate layer and propagate the uncertainty of the GP mapping to the latent variable $\boldsymbol{X}$, which yields the variational distribution

$$q(\boldsymbol{X}|\boldsymbol{Y}) = \prod\nolimits_i \mathcal{N}(\hat{\boldsymbol{m}}_i, \boldsymbol{S}_i + \hat{\sigma}_i^2 \boldsymbol{I}). \tag{6.7}$$

GP-decoder. The proposed recognition model, *i.e.*, the variational distribution of Eq. (6.7), can be employed to approximate the intractable marginal likelihood of Eq. (6.4). By introducing the variational distribution as an approximation to the true posterior, and after applying the Jensen's inequality, we obtain the lower bound to the log-marginal likelihood (again, labels $\boldsymbol{Z}$ are omitted)

$$\log p(\boldsymbol{Y}) \geq \mathcal{F}_1 = \sum\nolimits_v \mathbb{E}_{q(\boldsymbol{X}|\boldsymbol{Y})}\left[\log p(\boldsymbol{Y}^{(v)}|\boldsymbol{X})\right] - KL(q(\boldsymbol{X}|\boldsymbol{Y})||p(\boldsymbol{X})). \tag{6.8}$$

Training our model consists of maximizing the lower bound of Eq. (6.8) w.r.t. the variational parameters $\boldsymbol{M}, \boldsymbol{S}$ and the hyper-parameters of the kernels $\boldsymbol{K}^{(v)}, \boldsymbol{K}_r$. Further details are given in Sec. 6.2.4.

6.2.3 Incorporating Ordinal Variables

In the previous section, we presented the recognition model that we employ to learn a nonlinear manifold from the observed inputs. In the following, we further constrain this manifold by imposing an ordinal structure. This is attained by introducing ordinal variables that account for C ordinal levels of AUs. We use the notion of ordinal regression [2] and, in particular, the ordinal threshold model that imposes the monotonically increasing structure of the discrete

[2]For simplicity we assume an isotropic (diagonal) covariance across the dimensions.

output labels to the continuous manifold. Formally, the non-linear mapping between the manifold $\boldsymbol{X}$ and the ordinal outputs $\boldsymbol{Z}$ is modeled as

$$p(\boldsymbol{Z}|g(\boldsymbol{X})) = \prod_{i,c} p(z_{ic}|g_c(\boldsymbol{x}_i)), \quad p(z_{ic} = s|g_c(\boldsymbol{x}_i)) = \begin{cases} 1 & \text{if } g_c(\boldsymbol{x}_i) \in (\gamma_{c,s-1}, \gamma_{c,s}] \\ 0 & \text{otherwise,} \end{cases} \tag{6.9}$$

where $i = 1, \ldots, N$ indexes the training data. $\gamma_{c,0} = -\infty \leq \cdots \leq \gamma_{c,S} = +\infty$ are the thresholds or cut-off points that partition the real line into $s = 1, \ldots, S$ contiguous intervals. These intervals map the real function value $g_c(\boldsymbol{x})$ into the discrete variable s, corresponding to each of S intensity levels of an AU, while enforcing the ordinal constraints. The threshold model $p(z_{ic} = s|g_c(\boldsymbol{x}_i))$ is used for ideally noise-free cases. Here, we assume that the latent functions $g_c(\cdot)$[3] are corrupted by Gaussian noise, leading to the following formulation

$$g_c(\boldsymbol{x}_i) = \boldsymbol{w}_c^T \boldsymbol{x}_i + \epsilon_g, \quad \epsilon_g \sim \mathcal{N}(0, \sigma_g^2). \tag{6.10}$$

By integrating out the noisy projections from Eq. (6.9) (see [27] for details), we arrive at the ordinal log-likelihood

$$\log p(\boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{W}) = \sum\nolimits_{i,c} \mathbb{I}(z_{ic} = s) \log \left(\Phi\left(\frac{\gamma_{c,s} - \boldsymbol{w}_c^T \boldsymbol{x}_i}{\sigma_g} \right) - \Phi\left(\frac{\gamma_{c,s-1} - \boldsymbol{w}_c^T \boldsymbol{x}_i}{\sigma_g} \right) \right), \tag{6.11}$$

where $\Phi(\cdot)$ is the Gaussian cumulative density function, and $\mathbb{I}(\cdot)$ is the indicator function. Finally, by using the ordinal likelihood defined in Eq. (6.11), we obtain the final lower bound of our log-marginal likelihood

$$\begin{aligned} \log p(\boldsymbol{Y}, \boldsymbol{Z}|\boldsymbol{W}) \geq \mathcal{F}_2 = \sum\nolimits_v \mathbb{E}_{q(\boldsymbol{X}|\boldsymbol{Y})} \left[\log p(\boldsymbol{Y}^{(v)}|\boldsymbol{X}) \right] - KL(q(\boldsymbol{X}|\boldsymbol{Y})||p(\boldsymbol{X})) \\ + \sum_{i,c} \mathbb{I}(z_{ic} = s) \mathbb{E}_{q(\boldsymbol{X}|\boldsymbol{Y})} \left[\log \left(\Phi\left(\frac{\gamma_{c,s} - \boldsymbol{w}_c^T \boldsymbol{x}_i}{\sigma_g} \right) - \Phi\left(\frac{\gamma_{c,s-1} - \boldsymbol{w}_c^T \boldsymbol{x}_i}{\sigma_g} \right) \right) \right]. \end{aligned} \tag{6.12}$$

6.2.4 Learning and Inference

Training our model consists of maximizing the lower bound of Eq. (6.12) w.r.t. the variational parameters $\{\boldsymbol{S}, \boldsymbol{M}\}$, the hyper-parameters $\{\boldsymbol{\theta}^{(v)}, \sigma_v, \boldsymbol{\theta}_r^{(v)}, \sigma_r\}$ of the GP mappings, and the parameters $\{\boldsymbol{W}, \gamma, \sigma_g\}$ of the ordinal classifier. For the kernel of the GP-decoder we use the radial basis function (RBF) with automatic relevance determination (ARD), which can effectively estimate the dimensionality of the latent space [36]. For the kernel of the GP-encoder we use the isotropic RBF for each observed input. To utilize a joint optimization scheme, we use stochastic backpropagation [98, 147], where the re-parameterization trick is

[3] Note that we adopt here a linear model for $g_c(\cdot)$ as it operates on a low-dimensional non-linear manifold $\boldsymbol{X}$, already obtained by the GP auto-encoder.

applied in Eq. (6.12). Thus, we can obtain the Monte Carlo estimate of the expectation of the GP auto-encoder from

$$\mathbb{E}_{q(\boldsymbol{X}|\boldsymbol{Y})}\left[\log p(\boldsymbol{Y}^{(v)}|\boldsymbol{X})\right] = \sum\nolimits_i \mathbb{E}_{\mathcal{N}(\boldsymbol{\xi}|\boldsymbol{0},\boldsymbol{I})}\left[\log p(\boldsymbol{y}_i^{(v)}|\hat{\boldsymbol{m}}_i + (\boldsymbol{S}_i^{1/2} + \hat{\sigma}_i\boldsymbol{I})\boldsymbol{\xi})\right]. \tag{6.13}$$

The expectation of the ordinal classifier is computed in a similar manner. The advantage of Eq. (6.13) is twofold: (i) It allows for an efficient computation of the lower bound even when using arbitrary kernel functions (in contrast to [36]); (ii) It provides an efficient, low-variance estimator of the gradient [98]. The extra approximation (via the expectation) in the gradient step requires stochastic gradient descent. We use AdaDelta [198] for this purpose.

Inference in the proposed method is straightforward: The test data $\boldsymbol{y}_*^{(v)}$, are first projected onto the manifold using the trained GP-encoder. In the second step, we apply the ordinal classifier to the obtained latent position.

6.3 Relation to Prior Work on Gaussian Processes

Our auto-encoder approach is inspired by neural-network counterparts proposed in [98, 147], where probabilistic distributions are defined for the input and output mapping functions. In the GP literature, auto-encoders are closely related to the notion of 'back-constraints'. Back-constraints were introduced in [106] as a deterministic, parametric mapping (commonly a multi-layer perceptron (MLP)) that pairs the latent variables of the GPLVM [105] with the observations. This mapping facilitates a fast inference mechanism and enforces structure preservation in the manifold. The same mechanism has been used to constrain the shared GPLVM [167], from one view in [50] and multiple views in the DS-GPLVM from Chapter 4.

Back-constraints have been recently introduced to the B-GPLVM [178]. In [37] the authors proposed to approximate the true posterior of the latent space by introducing a variational distribution conditioned on some unobserved inputs. However, those inputs are not related to the observation space considered in this chapter (*i.e.*, the outputs $\boldsymbol{Y}$ of the GPLVM). In [34] the variational posterior of the latent space is constrained by using the trick of the parametric deterministic mapping from [106]. Finally, in the MC-LVM form the previous chapter, we replaced the variational approximation with a Monte Carlo expectation-maximization algorithm. Samples were obtained from the GP mapping from the observed inputs to the manifold.

Our proposed VGP-AE advances the current literature in many aspects: (1) We introduce a GP mapping for our recognition model. Hence, we can model different uncertainty levels per input and propagate them to the latent representations. (2) The use of the non-parametric

GPs also allows us to model complex structures at a lesser expense than the MLP (fewer parameters). Thus, it is less prone to overfitting and scales better to high-dimensional data. (3) Compared to [37] our probabilistic recognition model facilitates a low-dimensional projection of our observed features, while the variational constraint in [37] does not constitute a probabilistic mapping. (4) We learn the GP encoders/decoders in a joint optimization, while in the MC-LVM we trained the two models in an alternating scheme.

6.4 Experiments

In this section we empirically assess the structure learning abilities of the proposed VGP-AE as well as its efficacy when dealing with data of ordinal nature.

6.4.1 Experimental Protocol

Datasets. We first show the qualitative evaluation of the proposed VGP-AE on the MNIST [107] benchmark dataset of images of handwritten digits. We use it to assess the properties of the auto-endoced manifold. We then show the performance of VGP-AE on two benchmark datasets of facial affect: DISFA [122], and BP4D [208] (using the publicly available data subset from the FERA2015 [186] challenge). Specifically, DISFA contains video recordings of 27 subjects while watching YouTube videos. Each frame is coded in terms of the intensity of 12 AUs, on a six-point ordinal scale. The FERA2015 database includes video of 41 participants. There are 21 subjects in the training and 20 subjects in the development partition. The dataset contains intensity annotations for 5 AUs.

Features. In the experiment on MNIST dataset, we use the normalized raw pixel intensities as input, resulting in a 784D feature vector. For DISFA and FERA2015, we use both geometric and appearance features. Specifically, DISFA and FERA2015 datasets come with frame-by-frame annotations of 66 and 49 facial landmarks, respectively. After removing the contour landmarks from DISFA annotations, we end up with the same set of 49 facial points. We register the images to a reference face using an affine transform based on these points. We then extract LBP histograms [131] with 59 bins from patches centered around each registered point. Hence, we obtain 98D (geometric) and 2891D (appearance) feature vectors, commonly used in modeling of facial affect.

Evaluation. As evaluation measures, we use the negative log-predictive density (NLPD) to assess the generative ability (reconstruction part) of our model. For the task of ordinal

classification, we report the mean squared error (MSE) and the intra-class correlation (ICC(3,1)) [168]. These are the standard measures for ordinal data. The MSE measures the classifier's consistency regarding the relative order of the classes. ICC is a measure of agreement between annotators (in our case, the ground truth of the AU intensity and the model's predictions). Finally, we adopt the subject-independent setting: for FERA2015 we report the results on the subjects of the development set, while for DISFA we perform a 9-fold (3 subjects per fold) cross-validation procedure.

Models. We compare the proposed VGP-AE to the state-of-the-art GP manifold learning methods that perform multi-input multi-output inference. These include: (i) manifold relevance determination (MRD) [36], a regression model based on variational inference, (ii) variational auto-encoded deep GP (VAE-DGP) [34], which uses a recognition model based on an MLP to constrain the learning of MRD, and (iii) multi-task latent GP (MT-LGP) [185], which uses the same MLP-based recognition model and a maximum likelihood learning approach. We also compare to the variational GP for ordinal regression (vGPOR) [166]. As a baseline, we use the standard GP [146] with a shared covariance function among the multi-outputs. We also compare to the single-output ordinal threshold model (SOR) [2]. Finally, we compare to state-of-the-art methods for joint estimation of AU intensity based on MRFs [156] and latent trees (LT) [94], respectively. For the single input (no fusion) methods (GP, vGPOR, SOR, LT, MRF), we concatenate the two feature sets. The parameters of each method were tuned as described in the corresponding papers. For the GP subspace methods, we used the RBF kernel with ARD, and initialized with the 20D manifold. For the GP regression methods, we used the standard RBF. For the sparse variational GP methods (vGPOR, MRD, VAE-DGP) we used 200 inducing points, and 20 hidden units for the MLP in the recognition models of VAE-DGP and MT-LGP.

6.4.2 Assessing the Recognition Model

In the following, we qualitatively assess the benefits of the proposed recognition model in the task of manifold recovery from the MNIST dataset. We select an image depicting the digit '1' and rotate it around 360°. This results in a set of images of '1's rotated at a step of 1°. Our goal is to infer the true structure of the data, for which we know *a priori* that it should correspond to a diagonal-like kernel and a circular manifold. However, the challenge arises from the symmetry of digit '1', which is almost identical at opposite degrees (*e.g.*, 0° and 180°). The results are depicted in Fig. 6.2. Note that since we do not deal with the classification task we exclude the ordinal component in VGP-AE. We compare the learned manifold structure to

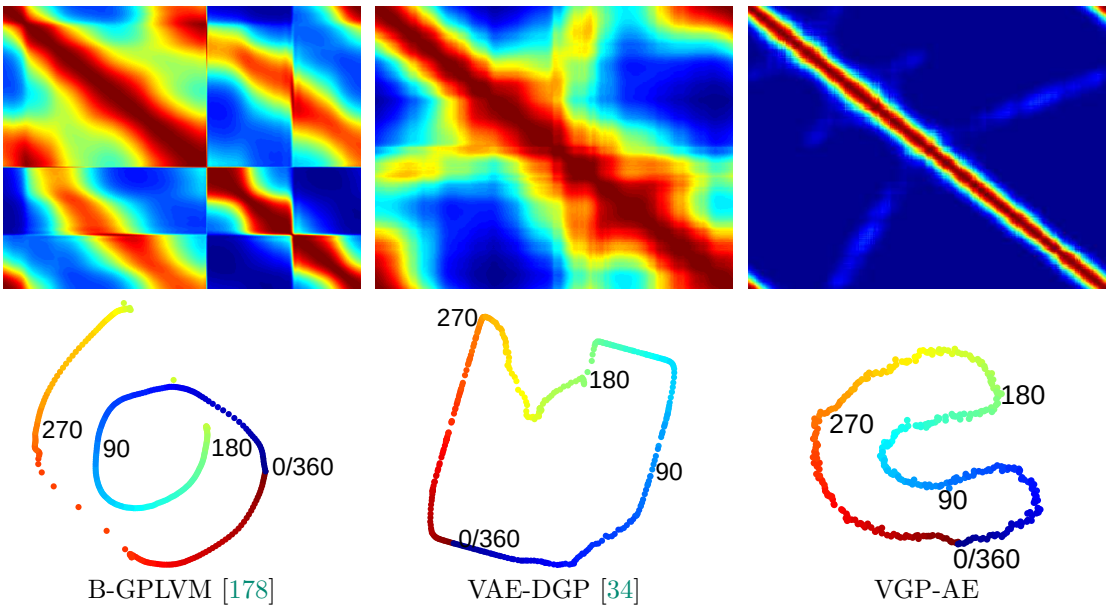
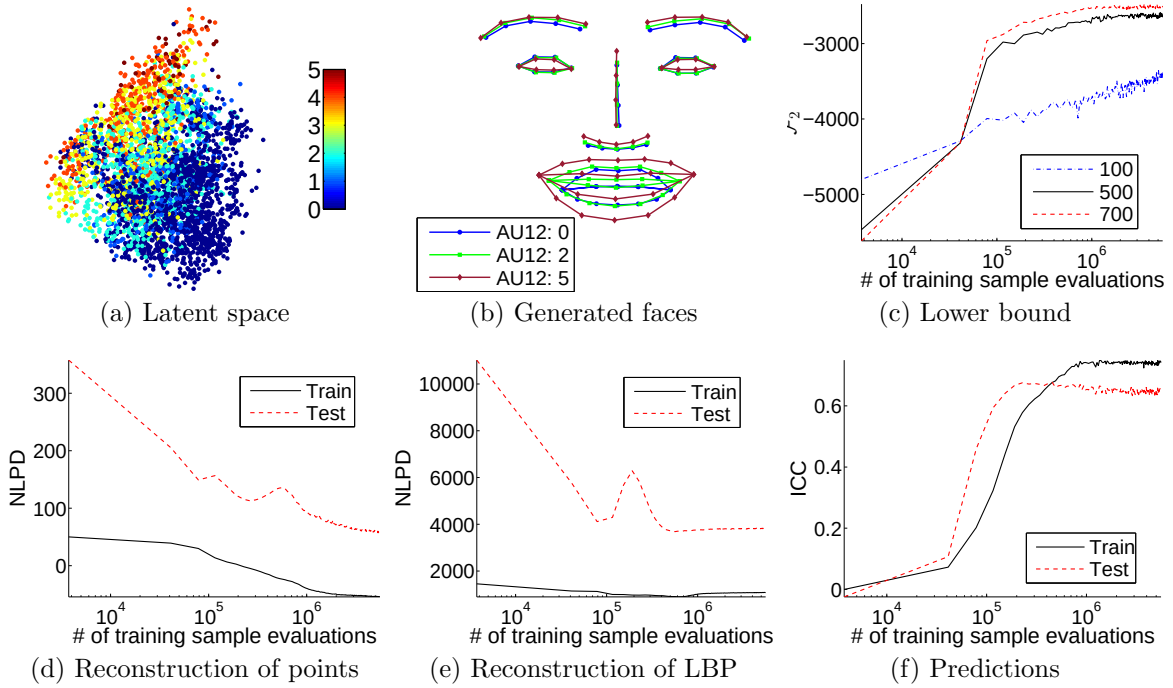


Figure 6.2: Recovering the structure of a rotated '1' from MNIST. The learned kernel matrices (upper row) and 2D manifolds (lower row) obtained from B-GPLVM (left), VAE-DGP (middle) and the proposed VGP-AE (right), initialized from the same random instance.

the B-GPLVM [178], which does not model the back-projection to the latent space, and a single layer VAE-DGP, where the back-projections are modeled using MLP. In Fig. 6.2 (upper row), we see from the learned kernels that the B-GPLVM is unable to fully unravel the dissimilarity between the 'inverted' images, resulting also in a non-smooth kernel with a discontinuity at 180° and 270°. By contrast, the VAE-DGP benefits from the recognition model and manages to resolve this to some extent. Yet, due to the deterministic nature of the recognition model, the recovered kernel still suffers from a discontinuity around 180°, while we also observe a flickering effect as we move away from the main diagonal. On the other hand, the proposed VGP-AE, by using the more general recognition model based on GPs (infinitely wide MLP), succeeds to accurately discover the true underlying manifold, also resulting in a more smooth, almost ideal kernel. These observations are further supported by the instances of the learned 2D manifolds in Fig. 6.2 (lower row). B-GPLVM learns a disconnected manifold with 'jumps' at 180° and 270°. However, both the VAE-DGP and proposed VGP-AE recover a circular manifold, with the manifold recovered by VGP-AE being more symmetric, although more 'wobbly' due to the sampling-based learning scheme.



(a) Latent space (b) Generated faces (c) Lower bound

(d) Reconstruction of points (e) Reconstruction of LBP (f) Predictions

Figure 6.3: Convergence analysis of the proposed method on FERA2015. (a) The recovered latent space with ordinal information from AU12, and (b) reconstructed face shapes sampled from different regions of the manifold. (c) The estimated average variational lower bound, $\mathcal{F}_2$, per datapoint, for different batch sizes. The model's reconstruction capacity for the points (d) and LBP (e) features, measured by the NLPD. (f) The average ICC for the joint AU intensity estimation. The horizontal axis corresponds to the amount of training points evaluated after 1500 epochs of the stochastic optimization.

6.4.3 Convergence Analysis

We next demonstrate the convergence of VGP-AE in the task of AU intensity estimation on FERA2015. Fig. 6.3(a) shows the effect of learning the ordinal classifier and the auto-encoded manifold within the joint optimization framework. It can be clearly seen from the recovered space that the information from the labels has been correctly encoded in the manifold, which now has an ordinal structure (the depicted coloring accounts for the 'ordinality' of AU12). As depicted in Fig. 6.3(b), we can accurately reconstruct face shapes with different AU intensities, by sampling from different regions of the space. Fig. 6.3(c) shows the convergence of the proposed method when optimizing the lower bound $\mathcal{F}_2$ of Eq. (6.12) for different batch sizes of the stochastic optimization. With a small batch size (100 datapoints) the model cannot estimate the structure of the inputs well. Hence, it approximates the log-marginal likelihood less accurately. By increasing the batch size to 500, the model converges to a better solution and optimization becomes more stable since the curve becomes smoother over the iterations. Further increase of the batch size does not have a considerable effect.

Table 6.1: Joint AU intensity estimation on DISFA and FERA2015.

	Dataset	DISFA													FERA2015					
	AU	1	2	4	5	6	9	12	15	17	20	25	26	Avg.	6	10	12	14	17	Avg.
ICC	VGP-AE	.48	.47	.62	.19	**.50**	**.42**	**.80**	.19	**.36**	.15	**.84**	.53	**.46**	**.75**	**.66**	**.88**	.47	**.49**	**.65**
	VAE-DGP [34]	.39	.34	.46	.13	.40	.31	.75	.14	.23	.14	.75	.45	.38	.72	.61	.82	.40	.38	.59
	MRD [36]	.46	.39	.43	.09	.28	.34	.71	.09	.30	.09	.73	.36	.36	.68	.59	.80	.38	.38	.57
	MT-LGP [185]	.41	.33	.28	.10	.23	.22	.56	.13	.26	.18	.65	.23	.30	.67	.61	.80	.37	.41	.57
	vGPOR [166]	**.53**	**.49**	.54	.21	.35	.40	.75	.18	.30	.16	.79	.39	.42	.74	.62	.84	**.48**	.35	.61
	GP [146]	.28	.13	.42	.03	.13	.23	.62	.08	.26	.19	.67	.23	.27	.69	.58	.81	.35	.38	.56
	SOR [2]	.25	.18	**.65**	.08	.46	.15	.77	.14	.24	.04	.82	**.57**	.36	.61	.50	.77	.28	.45	.52
	LT [94]	.28	.26	.44	.24	.50	.13	.69	.06	.21	.06	.62	.37	.32	.70	.59	.76	.30	.31	.53
	MRF [156]	.46	.38	.50	**.37**	.41	.34	.67	**.32**	.29	**.20**	.69	.46	.42	.64	.53	.79	.34	.46	.55
MSE	VGP-AE	.51	**.32**	1.13	.08	.56	.31	.47	.20	**.28**	.16	**.49**	**.44**	.41	**.82**	**1.28**	**.70**	**1.43**	**.77**	**1.00**
	VAE-DGP [34]	.40	.36	.95	.08	.48	.29	**.43**	.19	.32	.16	.76	**.44**	.41	.91	1.33	.81	1.46	.86	1.07
	MRD [36]	.42	.38	1.31	.08	.56	**.27**	.47	.20	.36	.18	.82	.53	.46	1.00	1.39	.83	1.64	.88	1.15
	MT-LGP [185]	.40	.35	1.25	.08	.60	.30	.73	.18	.36	.16	1.19	.67	.52	.97	1.31	.81	1.58	.84	1.10
	vGPOR [166]	.38	.34	.95	**.06**	.57	**.27**	**.43**	.18	.33	.18	.65	.53	.41	1.00	1.54	.76	1.78	1.11	1.24
	GP [146]	.52	.51	1.13	.13	.65	.36	.61	.23	.38	.20	.94	.66	.53	.94	1.40	.76	1.62	.88	1.12
	SOR [2]	.47	.40	1.13	.07	.63	.37	.55	.21	.35	.21	.71	.61	.48	1.44	1.82	1.08	2.58	1.01	1.59
	LT [94]	.44	.38	**.93**	**.06**	**.36**	.32	.46	.16	.29	**.15**	.97	**.44**	.41	.89	1.33	.91	1.48	.85	1.09
	MRF [156]	**.37**	.35	.94	**.06**	.45	.29	.46	**.13**	.32	.16	.77	**.44**	**.40**	1.20	1.66	.86	2.19	.92	1.37

In Fig. 6.3(d)–(e) we evaluate the generative part of the auto-encoder by measuring the model's ability to reconstruct both input features (points and LBPs) in terms of NLPD. First of all, it is clear that our Bayesian training prevents the model from overfitting, since the NLPD of the test data follows the trend of the training data. Furthermore, we can see that the model can reconstruct the geometric features better than the appearance, which is evidenced by the lower NLPD (around -50 for points and 1500 for LBPs). We partly attribute this to the fact that the LBPs are of higher dimension and therefore more difficult to reconstruct. Another reason for this difference is that the model learns to reconstruct the part of the features that enclose the more relevant information regarding the task of classification. The latter is further supported by Fig. 6.3(e), where we see the progress of the average ICC during the optimization. In the beginning, the model has no information since the latent space is initialized randomly. As we progress the model fuses the information of the input features in the latent space and unravels the structure of the data. Thus, ICC starts rising and reaches its highest value, .65 on the test data. After that point the model does no longer benefit from the appearance features: it has reached the plateau.

6.4.4 Model Comparisons on Spontaneous Data of Facial Expressions

We compare the proposed approach to several methods on the spontaneous data from the DISFA and FERA2015 datasets. Table 6.1 summarizes the results. First, we observe that all methods perform significantly better (in terms of ICC) on the data from FERA2015 than

on DISFA. This is mainly due to the fact that FERA2015 contains a much more balanced set of AUs (in terms of activations), and hence, all models (single- and multi-output) can learn the classifiers for the target task better. Furthermore, our proposed approach performs significantly better than the compared GP manifold learning methods, which treat the output labels as continuous variables. MRD lacks the modeling of back-projections. This results in learning a less smooth manifold of facial expressions, which affects its representation abilities, and hence, its predictions. On the other hand, the VAE-DGP learns explicitly the mapping from the observed features to the latent space in a deterministic and parametric fashion. Although this strategy is proven to be superior to unconstrained learning, it can be severely affected in cases where we have access to noisy and high-dimensional features. MT-LGP also models the back-mappings. However, it reports worse results, especially on DISFA. This drop in the performance is accounted to the non-Bayesian learning of the manifold, which makes the model more prone to overfitting.

Regarding the sparse ordinal regression instance of GPs, *i.e.*, vGPOR, we see that it manages to learn relatively accurate mappings between features and labels, and thus, performs close to our proposed method. However, it reports worse results since it cannot achieve the desirable fusion of the features without learning an intermediate latent space. The baseline methods, *i.e.*, GP and SOR, report lower results. The GP attains low scores due to handling the ordinal outputs in a continuous manner while the ordinal modeling helps SOR to report consistently better.

Finally, the proposed approach significantly outperforms the state-of-the-art methods in the literature of AU intensity estimation, *i.e.*, LT and MRF. LT learns the label information in a generative manner, and treats them as extra feature dimensions. Although this approach can be beneficial in the presence of noisy features [94], it suffers from learning complicated and large tree structures when falsely detecting connections between features and AUs. Hence, it performs worse. The MRF performs on par to the proposed method on DISFA and achieves the best average MSE, but it is consistently worse on FERA2015. This inconsistency is due to its two-step learning strategy, which results in unraveling a graph that cannot explain simultaneously all different features and AUs.

In Fig. 6.4 we evaluate the attained fusion between the best performing methods on FERA2015, *i.e.*, the proposed VGP-AE, VAE-DGP [34] and vGPOR [166]. As we can see, the proposed approach (solid line, first tuple) manages to accurately fuse the information from the two input features in the learned manifold. Thus, it achieves higher ICC on all AUs compared to when the two modalities are used individually as input features. On the other

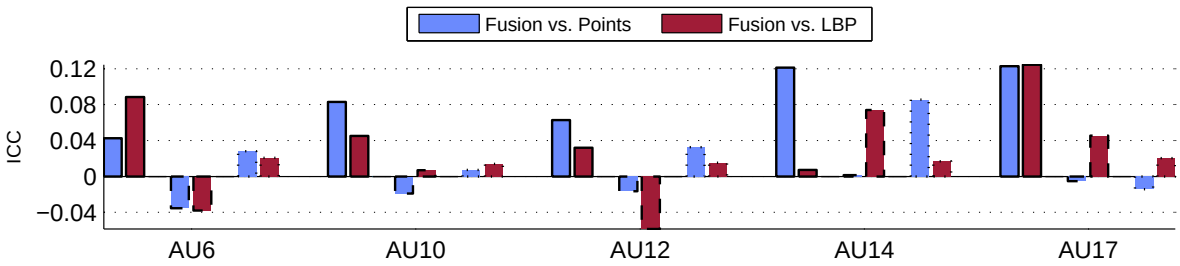


Figure 6.4: Demonstration of the gain/loss from feature fusion for joint AU intensity estimation on FERA2015. Within each AU the first tuple (solid line) corresponds to the proposed VGP-AE, the second tuple (dashed line) to the VAE-DGP [34], and the third tuple (dotted line) to the vGPOR [166].

hand, although vGPOR (third tuple, dotted line) reports also high ICC scores, it does not benefit from the presence of the two features: In most cases it cannot achieve a significant increase compared to the individual inputs. Finally, VAE-DGP (middle tuple, dashed line) consistently attains better performance on all AUs with a single feature as input. This can be attributed to modeling the recognition model via the parametric MLP. The latter affects the learning of the manifold, especially when dealing with the high-dimensional noisy appearance features.

The above mentioned difference between our approach and the VAE-DGP is further evidenced in Fig. 6.5. The proposed fusion along with the novel non-parametric, probabilistic recognition model in our auto-encoder leads to less confusion between the ordinal states across all AUs. We further attribute this to the ordinal modeling of outputs in our VGP-AE, contrary to VAE-DGP that treats the output as continuous variables. This is especially pronounced in the case of the subtle AUs 14&17, where examples of high intensity levels are scarce.

6.5 Conclusion

In this chapter we have presented a fully probabilistic GP auto-encoder, where GP mappings govern both the generative (GP-decoder) and the recognition (GP-encoder) models. The proposed variational GP auto-encoder is learned in a supervised manner, where the ordinal nature of the labels is imposed to the manifold. This allows the proposed approach to accurately learn the structure of the input data, while also remain competitive in the task of AU intensity estimation – an inherent ordinal problem. We have experimentally proved that our proposed probabilistic recognition model, apart from facilitating the back-mapping during inference, is also beneficial on unraveling more representative manifolds compared to when deterministic mappings are used. Furthermore, we have empirically evaluated our model on the task of facial feature fusion for joint intensity estimation of facial AUs. The proposed model outperforms

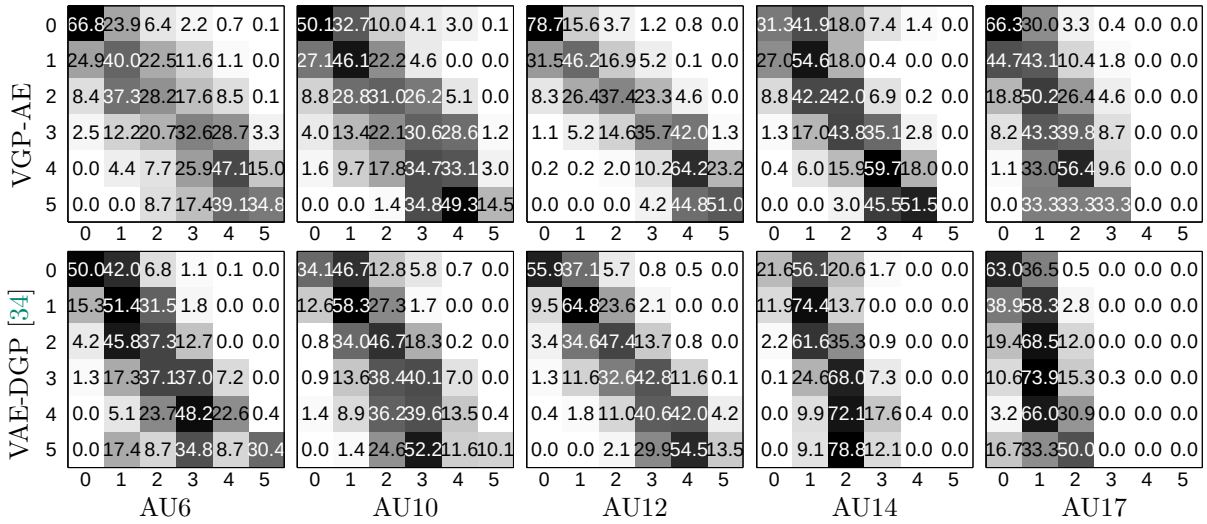


Figure 6.5: Confusion matrices for predicting the $0-5$ intensity of all AUs on FERA2015, when performing fusion with VGP-AE (upper row) and VAE-DGP [34] (lower row).

related GP methods and the state-of-the-art approaches for the target task.

Chapter 7

Gaussian Processes for Context Adaptation in Expression Analysis

Contents

7.1 Introduction

The models that we have presented in all previous chapters, although have been designed based on powerful generative models, may suffer a drop in their performance in the case where the input test data vary significantly from the training set. This can be addressed, up to some extent, by training the models on large amount of data that account for the unwanted variations. Our aim in this chapter is to find a data efficient approach to adapt the already trained generic models for facial behavior analysis. To achieve this we explore the notion of *domain adaptation* to address the tasks of (i) view and (ii) subject adaptation, for facial expression analysis of basic emotions and AUs. In particular, we address the problem of domain adaptation where the distribution of the (facial) features varies across domains (*i.e.*, contexts such as the view or subject), while the output labels (in our case, the emotion or AU activations) remain the same. The two domains are called *source* and *target* domain, respectively.

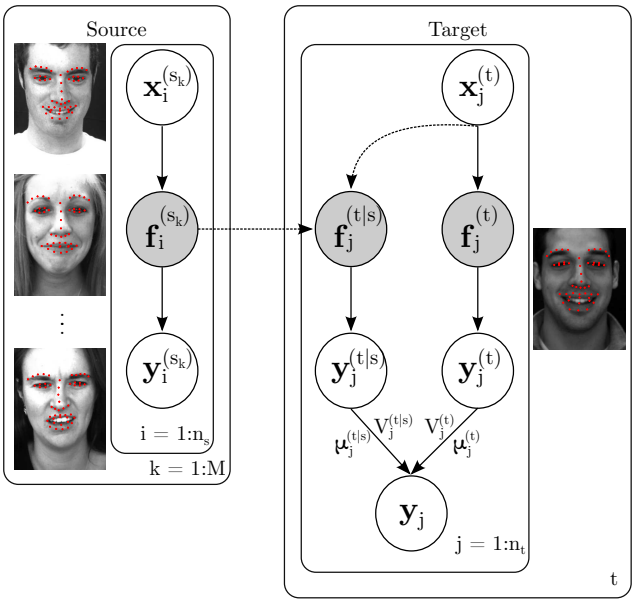


Figure 7.1: The proposed GPDE model. The learning consists of training the multiple source ($s_k, k = 1, \cdots, M$) and the target (t) GP experts (in this case, each subject is treated as an expert), using the available labeled training data pairs ($\boldsymbol{x}, \boldsymbol{y}$) – the input features (*e.g.*, facial landmarks) and output labels (*e.g.*, AU activations), respectively. Adaptation (dashed lines) for the target data is performed via conditioning the latent functions, $\boldsymbol{f}$, of the target GP on the source experts ($t|s$). During inference, we fuse the predictions from the experts ($\boldsymbol{\mu}^{\{t,(t|s)\}}$) by means of their predictive variance ($V^{\{t,(t|s)\}}$), with the role of a confidence measure.

Our domain adaptation model generalizes the product of GP expert models [43, 25] to the domain adaptation scenario. More specifically, instead of adjusting the classifier parameters between the domains, as in [28, 199, 26, 124, 157], we propose domain specific GP experts that model the domain specific data. The modeling power of GPs allows us to model the desired attributes in the target domain, in a data efficient manner. This is crucial for the training of the target expert since the available annotated data are usually scarce. Moreover, instead of minimizing the error between the distributions of the original source and target domain data, as in [28, 124], we use Bayesian domain adaptation [112] and explain the target data by conditioning on the learned source experts. The final prediction for the adapted classifier is obtained as a weighted combination of the predictions from the individual experts. The weighting is facilitated by measuring the confidence of each classifier. Contrary to [200] that represents the confidence heuristically as the agreement between a positive and a negative classifiers, in our probabilistic formulation during the adaptation we exploit the variance in the GP predictions when combining the source and target domains [161]. This results in a *confident* classifier that minimizes the risk of potential negative transfer (*i.e.*, the adapted model performing worse than the model trained using the adaptation data only). Finally,

in contrast to transductive adaptation approaches (*e.g.*, [28]) that need to be retrained completely, adaptation of our model is efficient and requires no retraining of the source model. An outline of the proposed model is depicted in Fig. 7.1. Note that the contents of this chapter are published in [57].

7.2 Gaussian Process Domain Experts (GPDE)

In the following, we introduce the notion of domain adaptation to the framework of GPs and present a novel methodology for obtaining a universal classifier with good generalization abilities and capable of modeling domain specific attributes.

7.2.1 Problem Formulation

We consider a supervised setting for domain adaptation, where we have access to a large collection of labeled *source* domain data, $\mathcal{S}$, and a smaller set of labeled *target* domain data, $\mathcal{T}$. Let $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{Y}}$ be the input (features) and output (labels) spaces, respectively. Hence, $\boldsymbol{X}^{(s)} = \{\boldsymbol{x}_{n_s}^{(s)}\}_{n_s=1}^{N_s}$ and $\boldsymbol{X}^{(t)} = \{\boldsymbol{x}_{n_t}^{(t)}\}_{n_t=1}^{N_t}$, with $\boldsymbol{x}_{n_s}^{(s)}, \boldsymbol{x}_{n_t}^{(t)} \in \mathbb{R}^D$, and $N_t \ll N_s$. In our case, the different domains can be different views or subjects. On the other hand, $\boldsymbol{Y}^{(s)} = \{\boldsymbol{y}_{n_s}^{(s)}\}_{n_s=1}^{N_s}$ and $\boldsymbol{Y}^{(t)} = \{\boldsymbol{y}_{n_t}^{(t)}\}_{n_t=1}^{N_t}$ correspond to same labels for both source and target domains. Each vector $\boldsymbol{y}_n^{\{s,t\}}$ contains the binary class labels of C classes. In order to avoid the burden of learning approximate solutions with GP classification, we formulate the predictions as a regression problem where:

$$\boldsymbol{y}_{n_v}^{(v)} = f^{(v)}(\boldsymbol{x}_{n_v}^{(v)}) + \boldsymbol{\epsilon}^{(v)}, \tag{7.1}$$

where $\boldsymbol{\epsilon}^{(v)} \sim \mathcal{N}(\boldsymbol{0}, \sigma_v^2 \boldsymbol{I})$ is i.i.d. additive Gaussian noise, and the index $v \in \{s,t\}$ denotes the dependence on each domain. The objective is to infer the latent functions $f^{(v)}$, given the training dataset $\mathcal{D}^{(v)} = \{\boldsymbol{X}^{(v)}, \boldsymbol{Y}^{(v)}\}$. To achieve this, we place a GP prior on the functions $f^{(v)}$, so that the function values $\boldsymbol{f}_{n_v}^{(v)} = f^{(v)}(\boldsymbol{x}_{n_v}^{(v)})$ follow a Gaussian distribution $p(\boldsymbol{F}^{(v)}|\boldsymbol{X}^{(v)}) = \mathcal{N}(\boldsymbol{F}^{(v)}|\boldsymbol{0}, \boldsymbol{K}^{(v)})$. Here, $\boldsymbol{F}^{(v)} = \{\boldsymbol{f}_{n_v}^{(v)}\}_{n_v=1}^{N_v}$, and $\boldsymbol{K}^{(v)} = k^{(v)}(\boldsymbol{X}^{(v)}, \boldsymbol{X}^{(v)})$ is the kernel covariance function, which is assumed to be shared among the label dimensions. In this chapter, we employ the RBF kernel

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f^2 \exp\Big(-\frac{1}{2\ell^2} \|\boldsymbol{x} - \boldsymbol{x}'\|^2 \Big), \tag{7.2}$$

where $\{\ell, \sigma_f\}$ are the kernel hyper-parameters. The regression mapping can be fully defined by the set of hyper-parameters $\boldsymbol{\theta} = \{\ell, \sigma_f, \sigma_v\}$. Training of the GP consists of finding the hyper-parameters that maximize the log-marginal likelihood

$$\log p(\boldsymbol{Y}^{(v)}|\boldsymbol{X}^{(v)}, \boldsymbol{\theta}^{(v)}) = -\frac{C}{2} \log |\boldsymbol{K}^{(v)} + \sigma_v^2 \boldsymbol{I}| - \frac{1}{2} \mathrm{tr}\left[(\boldsymbol{K}^{(v)} + \sigma_v^2 \boldsymbol{I})^{-1} \boldsymbol{Y}^{(v)} \boldsymbol{Y}^{(v)^T} \right] + \text{const.} \tag{7.3}$$

Given a test input $\boldsymbol{x}_*^{(v)}$ we obtain the GP predictive distribution by conditioning on the training data $\mathcal{D}^{(v)}$ as $p(\boldsymbol{f}_*^{(v)}|\boldsymbol{x}_*^{(v)}, \mathcal{D}^{(v)}) = \mathcal{N}(\mu^{(v)}(\boldsymbol{x}_*^{(v)}), V^{(v)}(\boldsymbol{x}_*^{(v)}))$ with

$$\mu^{(v)}(\boldsymbol{x}_*^{(v)}) = \boldsymbol{k}_*^{(v)^T}(\boldsymbol{K}^{(v)} + \sigma_v^2\boldsymbol{I})^{-1}\boldsymbol{Y}^{(v)} \tag{7.4}$$

$$V^{(v)}(\boldsymbol{x}_*^{(v)}) = k_{**}^{(v)} - \boldsymbol{k}_*^{(v)^T}(\boldsymbol{K}^{(v)} + \sigma_v^2\boldsymbol{I})^{-1}\boldsymbol{k}_*^{(v)}, \tag{7.5}$$

where $\boldsymbol{k}_*^{(v)} = k^{(v)}(\boldsymbol{X}^{(v)}, \boldsymbol{x}_*^{(v)})$ and $k_{**}^{(v)} = k^{(v)}(\boldsymbol{x}_*^{(v)}, \boldsymbol{x}_*^{(v)})$. For convenience we denote $\boldsymbol{\mu}_*^{(v)} = \mu^{(v)}(\boldsymbol{x}_*^{(v)})$ and $V_{**}^{(v)} = V^{(v)}(\boldsymbol{x}_*^{(v)})$. Under this general formulation, we have the choice to learn either (i) independent functions $f^{(v)}$ or (ii) a universal function f that couples the data from the two domains. However, neither option allows us to explore the idea of domain adaptation: In the former we learn domain-specific models, while in the latter we simplify the problem by concatenating the data from the two domains. An alternative would be to merge the two approaches in order to achieve a better generalization, while also being able to model domain specific attributes. Such a combined approach would allow us to obtain more robust predictions.

7.2.2 GP Adaptation

A straightforward approach to obtain a model capable of performing inference on data from both domains is to assume the existence of a universal latent function with a single set of hyper-parameters $\boldsymbol{\theta}$. Thus, the authors in [112] proposed a simple, yet effective, three-step approach for GP adaptation (GPA):

1. Train a GP on the source data with marginal likelihood $p(\boldsymbol{Y}^{(s)}|\boldsymbol{X}^{(s)}, \boldsymbol{\theta})$ to learn the hyper-parameters $\boldsymbol{\theta}$. The posterior distribution is the given by Eqs. (7.4)–(7.5).

2. Use the obtained posterior distribution of the source data, as a prior for the GP of the target data $p(\boldsymbol{Y}^{(t)}|\boldsymbol{X}^{(t)}, \mathcal{D}^{(s)}, \boldsymbol{\theta})$.

3. Correct the posterior distribution to account for the target data $\mathcal{D}^{(t)}$ as well.

Now the conditional prior of the target data (given the source data) in the second step is given by applying Eqs. (7.4)–(7.5) on $\boldsymbol{X}^{(t)}$

$$\boldsymbol{\mu}^{(t|s)} = \boldsymbol{K}_{st}^{(s)^T}(\boldsymbol{K}^{(s)} + \sigma_s^2\boldsymbol{I})^{-1}\boldsymbol{Y}^{(s)} \tag{7.6}$$

$$\boldsymbol{V}^{(t|s)} = \boldsymbol{K}_{tt}^{(s)} - \boldsymbol{K}_{st}^{(s)^T}(\boldsymbol{K}^{(s)} + \sigma_s^2\boldsymbol{I})^{-1}\boldsymbol{K}_{st}^{(s)}, \tag{7.7}$$

where $\boldsymbol{K}_{tt}^{(s)} = k^{(s)}(\boldsymbol{X}^{(t)}, \boldsymbol{X}^{(t)}), \boldsymbol{K}_{st}^{(s)} = k^{(s)}(\boldsymbol{X}^{(s)}, \boldsymbol{X}^{(t)})$, and the superscript $t|s$ denotes the conditioning order. Given the above prior and a test input $\boldsymbol{x}_*^{(t)}$, the correct form of the adapted posterior after observing the target domain data is given by:

$$\mu_{ad}^{(s)}(\boldsymbol{x}_*^{(t)}) = \boldsymbol{\mu}_*^{(s)} + \boldsymbol{V}_*^{(t|s)^T}(\boldsymbol{V}^{(t|s)} + \sigma_s^2\boldsymbol{I})^{-1}(\boldsymbol{Y}^{(t)} - \boldsymbol{\mu}^{(t|s)}) \tag{7.8}$$

$$V_{ad}^{(s)}(\boldsymbol{x}_*^{(t)}) = V_{**}^{(s)} - \boldsymbol{V}_*^{(t|s)^T}(\boldsymbol{V}^{(t|s)} + \sigma_s^2\boldsymbol{I})^{-1}\boldsymbol{V}_*^{(t|s)}, \tag{7.9}$$

with $\boldsymbol{V}_*^{(t|s)} = k^{(s)}(\boldsymbol{X}^{(t)}, \boldsymbol{x}_*^{(t)}) - k^{(s)}(\boldsymbol{X}^{(s)}, \boldsymbol{X}^{(t)})^T(\boldsymbol{K}^{(s)} + \sigma_s^2\boldsymbol{I})^{-1}k^{(s)}(\boldsymbol{X}^{(s)}, \boldsymbol{x}_*^{(t)})$.

Eqs. (7.8)–(7.9) show that final prediction in the GPA is the combination of the original prediction based on the source data only, plus a correction term. The latter shifts the mean toward the distribution of the target data and improves the model's confidence by reducing the predictive variance. Note that we originally constrained the model to learn a single latent function f for both conditional distributions $p(\boldsymbol{Y}^{(v)}|\boldsymbol{X}^{(v)})$ to derive the posterior for the GPA. However, this constraint implies that the marginal distributions of the data $p(\boldsymbol{X}^{(v)})$ are similar. This assumption violates the general idea of domain adaptation, where by definition, the marginals may have significantly different attributes (*e.g.*, input features from different observation views). In such cases, GPA could perform worse than an independent GP trained solely on the target data $\mathcal{D}^{(t)}$. One possible way to address this issue is to retrain the $\log p(\boldsymbol{Y}^{(t)}|\boldsymbol{X}^{(t)}, \mathcal{D}^{(s)}, \boldsymbol{\theta})$ of the GPA w.r.t. $\boldsymbol{\theta}$ [112]. This option will compensate for the differences in the distributions by readjusting the hyper-parameters. However, it comes with the price of retraining of the model. Furthermore, it does not allow for modeling domain-specific attributes since the predictions are still determined mainly from the source distribution.

7.2.3 Domain Experts

In the proposed GP domain experts (GPDE), we assume that each expert is a GP that operates only on a subset of data, *i.e.*, $\mathcal{D}^{(s)}, \mathcal{D}^{(t)}$. Hence, we can follow the methodology presented in Sec. 7.2.1 in order to train domain-specific GPs and learn different latent functions, *i.e.*, hyper-parameters $\boldsymbol{\theta}^{(v)}$. Within the current formulation we treat the source domain as a combination of multiple source datasets (*e.g.*, subject-specific datasets) $\mathcal{D}^{(s)} = \{\mathcal{D}^{(s_1)}, \ldots, \mathcal{D}^{(s_M)}\}$, where M is the total number of source domains (datasets).

Training. Given the above mentioned data split and assuming conditional independence of the labels from each domain given the corresponding input features, the marginal likelihood

can be approximated by

$$p(\boldsymbol{Y}^{\{s,t\}},|\boldsymbol{X}^{\{s,t\}},\boldsymbol{\theta}^{\{s,t\}}) = p(\boldsymbol{Y}^{(t)}|\boldsymbol{X}^{(t)},\boldsymbol{\theta}^{(t)})\prod_{k=1}^{M} p_k(\boldsymbol{Y}^{(s_k)}|\boldsymbol{X}^{(s_k)},\boldsymbol{\theta}^{(s)}). \tag{7.10}$$

Note that we share the set of hyper-parameters $\boldsymbol{\theta}^{(s)}$ across all the source domains. The intuition behind this is that in each source domain we may observe different label distribution $p(\boldsymbol{Y}^{(s_k)})$, yet after exploiting all the available datasets we can model the overall distribution $p(\boldsymbol{Y}^{(s)})$ with a single set of hyper-parameters $\boldsymbol{\theta}^{(s)}$. However, this does not guarantee that we are also able to explain the target label distribution $p(\boldsymbol{Y}^{(t)})$ with the same hyper-parameters. Thus, we also search for $\boldsymbol{\theta}^{(t)}$ for modeling the domain-specific attributes. Similar to Sec. 7.2.1 learning of the hyper-parameters is performed by maximizing

$$\log p(\boldsymbol{Y}^{\{s,t\}},|\boldsymbol{X}^{\{s,t\}},\boldsymbol{\theta}^{\{s,t\}}) = \log p(\boldsymbol{Y}^{(t)}|\boldsymbol{X}^{(t)},\boldsymbol{\theta}^{(t)}) + \sum_{k=1}^{M} \log p_k(\boldsymbol{Y}^{(s_k)}|\boldsymbol{X}^{(s_k)},\boldsymbol{\theta}^{(s)}), \tag{7.11}$$

where each log-marginal is computed according to Eq. (7.3). The above factorization, apart from facilitating learning of the domain experts, allows for efficient GP training even with larger datasets, as shown in [43]. Note that the source experts can be learned independently from the target, which allows our model to generalize to unseen target domains without retraining.

Predictions. Once we have trained the GPDE, we need to combine the predictions from each expert to form an overall prediction. To achieve that, we follow the approach presented in [25], where we further readjust the predictions from the source experts using the conditional adaptation from GPA. Hence, the predictive distribution is given by

$$p(\boldsymbol{f}_*^{(t)}|\boldsymbol{x}_*^{(t)},\mathcal{D}) = \prod_{k=1}^{M} p_k^{\beta_{s_k}}(\boldsymbol{f}_*^{(t)}|\boldsymbol{x}_*^{(t)},\mathcal{D}^{(s_k)},\mathcal{D}^{(t)},\boldsymbol{\theta}^{(s)}) \cdot p^{\beta_t}(\boldsymbol{f}_*^{(t)}|\boldsymbol{x}_*^{(t)},\mathcal{D}^{(t)},\boldsymbol{\theta}^{(t)}), \tag{7.12}$$

where β_{s_k}, β_t control the contribution of each expert. In this work we equally weight the experts and normalize them such that $\beta_t + \sum \beta_{s_k} = 1$, as suggested in [43]. The predictive mean and variance are given by

$$\boldsymbol{\mu}_*^{\text{gpde}} = V_*^{\text{gpde}}\left[\beta_t {V_*^{(t)}}^{-1}\boldsymbol{\mu}_*^{(t)} + \sum\nolimits_k \beta_{s_k} {V_{ad}^{(s_k)}}^{-1}\boldsymbol{\mu}_{ad}^{(s_k)}\right] \tag{7.13}$$

$$V_*^{\text{gpde}} = \left[\beta_t {V_*^{(t)}}^{-1} + \sum\nolimits_k \beta_{s_k} {V_{ad}^{(s_k)}}^{-1}\right]^{-1}. \tag{7.14}$$

At this point the contribution of the GPDE becomes clear: Eq. (7.13) shows that the overall mean is the sum of the predictions from each expert, weighted by their precision (inverse

variance). Hence, the solution of the GPDE will favor the predictions of more confident experts. On the other hand, if the quality of a domain expert is poor (noisy predictions with large variance), GPDE will weaken its contribution to the overall prediction.

7.2.4 Weighted GP Domain Experts for imbalanced outputs

In the analysis we conducted so far, we treated the multiple outputs as i.i.d. samples from a joint Gaussian distribution. Hence, we assumed a shared covariance matrix among the multiple output dimensions, which results in the same weighting/variance in Eqs. (7.13)–(7.14). This could be problematic in cases where we have to deal with imbalanced data in the output, (*e.g.*, different AUs with different occurrence patterns). Thus, it is important in each expert to account for a different variance per output. To address this, we follow the approach presented in [75, 184], and introduce a weighting matrix to the log-marginal likelihood of each expert in Eq. (7.11), so that

$$\begin{aligned}\log p(\boldsymbol{Y}^{(v)}|\boldsymbol{X}^{(v)},\boldsymbol{\theta}^{(v)}) = -\frac{1}{2}\mathrm{tr}\left[(\boldsymbol{K}^{(v)}+\sigma_v^2\boldsymbol{I})^{-1}\boldsymbol{Y}^{(v)}\boldsymbol{\Lambda}^{(v)}\boldsymbol{Y}^{(v)^T}\right] \\ -\frac{C}{2}\log|\boldsymbol{K}^{(v)}+\sigma_v^2\boldsymbol{I}| + \frac{N_v}{2}\log|\boldsymbol{\Lambda}^{(v)}| + \text{const},\end{aligned} \tag{7.15}$$

where $\boldsymbol{\Lambda}^{(v)} = \mathrm{diag}(\lambda_1^{(v)},\cdots,\lambda_C^{(v)})$. This is equivalent to learning a GP with kernel covariance function $k^{(v)}(\cdot,\cdot) = k^{(v)}(\cdot,\cdot)/\lambda_c^{(v)}$ for each output dimension c. The term $1/\lambda_c^{(v)}$ accounts for the different variances in the output dimensions and gives more flexibility to the model, since more representative input-output mappings can be learned.

Note, however, that the predicted variance of a probabilistic model depends highly on the training data. A GP domain expert can have access to data with zero activations for a certain output, while other outputs may frequently co-occur together. This suggests that there exists an intrinsic structure between the outputs, which we do not account for within the GPDE. To ameliorate this, we re-parameterize $\lambda_c^{(v)}$ as

$$\frac{1}{\lambda_c^{(v)}} = \frac{w_c^{(v)}}{\sum_c w_c^{(v)}}, \tag{7.16}$$

where $w_c^{(v)}$ is the new parameter to learn. As we can see from Eq. (7.16), the variance of each output is now proportional to the amount of the total variance. Such a re-parameterization correctly enforces the total variance of the GP to be distributed to the various outputs. It can be also regarded as a straightforward way to rectify the assumption of having i.i.d. outputs, since now frequently co-occurring outputs will be assigned similar weights, and, hence, a similar covariance function. We name the approach presented here as *weighted* Gaussian process domain experts (wGPDE) to differentiate it from the single variance GPDE.

Algorithm 3 Domain adaptation with (w)GPDE

Inputs: $\mathcal{D}^{(s)} = \{\boldsymbol{X}^{(s)}, \boldsymbol{Y}^{(s)}\}, \mathcal{D}^{(t)} = \{\boldsymbol{X}^{(t)}, \boldsymbol{Y}^{(t)}\}$
Training:
Learn the hyper-parameters $\boldsymbol{\theta}^{\{s,t\}}$ by maximizing Eq. (7.11).
Adaptation:
Adapt the posterior from the source experts via Eq. (7.8)–(7.9).
Predictions of Experts:
Combine the prediction from each GP domain expert via
Eq. (7.13)–(7.14) for GPDE or Eq. (7.17)–(7.18) for wGPDE.
Output: $\boldsymbol{y}_* = sign(\boldsymbol{\mu}_*^{gpde})$.

Re-weighted Predictions. By propagating the weighting matrix $\boldsymbol{\Lambda}^{(v)}$ to the predictive distribution of the proposed wGPDE, we can derive the re-weighted predictions for the c-th output

$$\boldsymbol{\mu}_{*_c}^{\text{gpde}} = V_{*_c}^{\text{gpde}} \left[\beta_t \lambda_c^{(t)} {V_*^{(t)}}^{-1} \boldsymbol{\mu}_{*_c}^{(t)} + \sum\nolimits_k \beta_{s_k} \lambda_c^{(s_k)} {V_{ad}^{(s_k)}}^{-1} \boldsymbol{\mu}_{ad_c}^{(s_k)} \right] \tag{7.17}$$

$$V_{*_c}^{\text{gpde}} = \left[\beta_t \lambda_c^{(t)} {V_*^{(t)}}^{-1} + \sum\nolimits_k \beta_{s_k} \lambda_c^{(s_k)} {V_{ad}^{(s_k)}}^{-1} \right]^{-1}. \tag{7.18}$$

By comparing Eqs. (7.13)–(7.14) to Eqs. (7.17)–(7.18) we see that the combined predictions from all the experts depend on the predicted variance of each output. This allows the re-weighted experts to be confident (higher contribution to the overall prediction) for certain outputs, while remaining 'silent' for outputs that have not seen. On the contrary, Eqs. (7.13)–(7.14) assign the same weight to all outputs, a fact that increases the bias in the predictions. Algorithm 3 summarizes the adaptation procedure of the proposed (w)GPDE.

7.3 Relation to Prior Work on Domain Adaptation

Domain adaptation is a well studied problem in machine learning (for an extensive survey, see [144]). The adaptation can be performed either in an *unsupervised* or a *(semi-)supervised* setting, based on the availability of labeled target domain data. The approaches that operate on the first setting, usually focus on deriving a common subspace where the distribution mismatch between source and target data is diminished. For instance, a manifold learning approach has been proposed in [71], where labeled data from the source domain and unlabeled data from target domain are first mapped on the Grassmann manifold, before learning a classifier. Similarly, [70] treats the source and target domains as connected points in the Grassmann manifold. The intermediate points (domains) in the path are integrated out in order to propagate the information from the source to the target domain data. More recently, [65, 5] proposed to align the eigenspaces from the two domains and train a classifier

on the aligned source domain data. In a similar attempt, [173] proposed to whiten the data in order to align the correlations between the source and target domain, before applying the classification. The above approaches can be very effective in cases where we do not have access to labeled target data. However, even when few labels from the target domain become available, unsupervised methods should not be preferred, since they fail to integrate the class knowledge from the target domain to the adaptation step.

The (semi-)supervised setting is more appropriate to our target task, since the available labels can be used to enhance the classification. One of the first attempts toward this directions has been presented in [41]. The authors proposed to replicate the input features to produce shared and domain-specific features, which are then fed into a classifier. Although straightforward, this approach has been proven effective for the adaptation task. [101] learns a transformation that maximizes similarity between data in the source and target domains by enforcing data pairs with the same labels to have high similarity, and pairs with different labels to be dissimilar. Then, a k-NN classifier is used to perform classification of target data. [82] is an extension of this approach to multiple source domains. The input data are assumed to be generated from category-specific local domain mixtures, the mixing weights of which determine the underlying domain of the data, classified using an SVM classifier. Similarly, [83] learns a linear asymmetric transformation to maximally align target features to the source domain. This is attained by introducing max-margin constraints that allow the learning of the transformation matrix and SVM classifier jointly. [46] extends the work in [83] by introducing additional constraints to the max-margin formulation. More specifically, unlabeled data from the target domain are used to enforce the classifier to produce similar predictions for similar target-source data. While these methods attempt to directly align the target to source features, several works attempted this through a shared manifold. For instance, [48] learns a non-linear transformation from both source and target data to a shared latent space, along with the target classifier. Likewise, [196] finds a low-dimensional subspace, which preserves the structure across the domains. The subspace is facilitated by projections that are learned jointly with the linear classifier. The structure preservation constraints are used to ensure that similar data across domains are close in the subspace.

All of the above methods tackle the adaptation problem in a deterministic fashion. Thus, they do not provide a measure of confidence in the target predictions. By contrast, our approach is fully probabilistic and non-parametric due to the use of GPs. Thus, the proposed method is more related to recent advances in the literature [69, 112, 96] that perform the domain adaptation in a Bayesian fashion. Specifically, in [69] a discriminative framework is

proposed to couple data from different domains in a shared subspace. Task-specific projections are learned simultaneously with the classifiers in order to couple all the task from the multiple domains in the obtained subspace. In [112], the predictive distribution of a GP trained on the source data is used as a prior for the joint distribution of the source and target domains. The information from the source domain can be analytically propagated to the inference of the target data by simply following the conditional properties of the GPs. Similarly, in [96] the authors proposed a two-layer GP that jointly learns separate discriminative functions from the source and target features to the labels. The intermediate layer facilitates the adaptation step, and a variational approximation is employed to integrate out this layer.

Compared to the aforementioned work, our approach has some key differences: In [69] the authors learn the classifier on a subspace shared among the data from source and target domains. This can be problematic in cases where access to target domain data is confined, since it can impose a bias on the manifold toward the source domain. In contrast to [112], our proposed approach defines a target specific expert, which is then combined with the source domain experts. The benefit of this is that the resulting classifier is not limited by the distribution of the source data. Also, in contrast to [96], the training of the experts is performed independently, and thus, we need not retrain the source classifier.

7.4 Experiments

7.4.1 Experimental Protocol

Datasets. We evaluate the proposed model on acted and spontaneous facial expressions from three publicly available datasets: MultiPIE [76], Denver Intensity of Spontaneous Facial Actions (DISFA) [122] and BP4D [208] (using the publicly available data subset from the FERA2015 [186] challenge). Specifically, MultiPIE contains images of 373 subjects depicting acted facial expressions of Neutral (NE), Disgust (DI), Surprise (SU), Smile (SM), Scream (SC) and Squint (SQ), captured at various pan angles. In our experiments, we used images from $0°$, $-15°$ and $-30°$. DISFA is widely used in the AU-related literature, due to the large amount of (subjects and AUs) annotated images. It contains video recordings of 27 subjects while watching YouTube videos. Each frame is coded in terms of the intensity of 12 AUs, coded on a six-point ordinal scale. For our experiments we treated each AU with intensity larger than zero as active. FERA2015 database includes video of 41 participants. There are 21 subjects in the training and 20 subjects in the development partition. Each video is annotated in terms of occurrence of 11 AUs. Example images of the three datasets are given in Fig. 7.2.

Figure 7.2: Example images from MultiPIE (top), DISFA (middle) and FERA2015 (bottom) datasets.

Features. We use both a set of geometric features derived from the facial landmark locations, as well as appearance features. Specifically, DISFA and FERA2015 datasets come with frame-by-frame annotations of 66 and 49 facial landmarks, respectively, while a set of 66 annotated points for MultiPIE were obtained from [154]. After removing the contour landmarks from DISFA and MultiPIE annotations, we end up with the same set of 49 facial points for all three datasets. These were then registered to a reference face (average face per view for MultiPIE, and average face for DISFA and FERA2015) using an affine transformation. We then extract LBP histograms [131] with 59 bins from patches centered around each registered point. Hence, we obtain 98D (geometric) and 2891D (appearance) feature vectors, commonly used in modeling of facial affect. For the high dimensional appearance features, in order to remove potential noise and artifacts, and also reduce the dimensionality, we applied PCA, retaining 95% of the energy, which resulted in approximately 200D appearance feature vectors.

Evaluation procedure. We evaluate GPDE and wGPDE on both multi-class (facial expression classification of basic emotions on MultiPIE) and multi-label (multiple AU detection on DISFA and FERA2015) scenarios. We also assess the adaptation capacity of the model with a single (view adaptation) and multiple (subject adaptation) source domains. For the task of emotion classification, images from $0°$, $-15°$ and $-30°$ served interchangeably as the source domain, while inference was performed via adaptation to the remaining views. For the AU detection task, the various subjects from the training data were used as multiple source domains, and adaptation was performed each time on the tested subject.

To evaluate the model's adaptation ability we strictly follow a training protocol, where for each experiment we vary the cardinality of the training target data (we always use all the

available source domain data). For MultiPIE, we first split the data in 5-folds (4 training, 1 testing and iterate over all folds) and then, we keep increasing the cardinality as: $N_t = 10, 30, 50, 100, 200, 300, 600, 1200$. For DISFA we follow a leave-one-subject-out approach (26 training source subjects and 1 target test subject at a time). For FERA2015 we followed the original partitioning suggested in [186] (20 training source subjects from the training partition, while each of the 20 subjects in the development partition served as an individual target domain). From the test subject's sequence in DISFA and FERA2015 the first 500 frames were used as target training data (with increasing cardinality $N_t = 10, 30, 50, 100, 200, 500$), while inference was performed on the rest frames of the sequence. This is in order to avoid the target model overfitting the temporally neighboring examples of the test subject. For the emotion classification experiments, we employ the classification ratio (CR) as the evaluation measure, while for the AU detection we report the F1 score and the area under the ROC curve (AUC). Both F1 and AUC are widely used in the literature as they quantify different characteristics of the classifiers' performance. Specifically, F1, defined as $F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$, is the harmonic mean between the precision and recall. It puts emphasis on the classification task, while being largely robust to imbalanced data (such as examples of different AUs). AUC quantifies the relation between true and false positives, showing the robustness of a classifier to the choice of its decision threshold.

Models compared. We compare the proposed approach with the two generic models GP_{source} and GP_{target}. The former is trained solely on the source data, while the latter on the target data used for the adaptation. Furthermore, we compare to the state-of-the-art models based on GPs for supervised domain adaptation, *i.e.*, the GPA [112] and the asymmetric transfer learning with deep GP (ATL-DGP) [96]. The GPA is an instance of the proposed GPDE, with only a source domain expert (no target) and predictions given by Eqs. (7.8)–(7.9). ATL-DGP employs an intermediate GP to combine the predictions of GP_{source} and GP_{target}. Apart from the GP-based domain adaptation techniques, we further compare to the deterministic max-margin domain transfer (MMDT) [83], that adjusts the SVM classifier to the domain adaptation scenario, and kernelized Bayesian transfer learning (KBTL) [69] that finds a shared subspace appropriate for the classification of various tasks (domains) in a probabilistic manner. Finally, we compare to state-of-the-art domain adaptation methods from the field of action unit analysis, *i.e.*, the dynamic SVM (dynSVM) [11] that performs the adaptation by neutral calibration (*e.g.*, removing the average, per subject, neutral image from the input data), and the confidence preserving machine (CPM) [200] that reweights the source classifier based on a confidence measure, before applying it to the data from the target

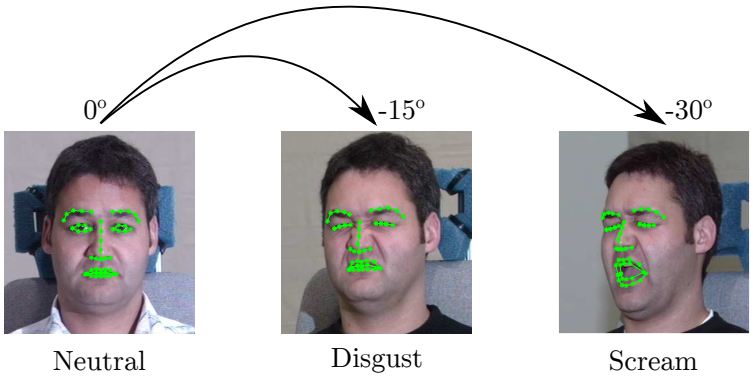


Figure 7.3: View adaptation for emotion classification on the MultiPIE dataset.

subject. Note that implementation of (dynSVM) and CPM were not available, and thus, in our comparisons we report the available results from the authors' papers and websites. For the other compared methods, all relevant parameters were tuned based on a cross-validation strategy. On the other hand, the proposed (w)GPDE is a non-parametric model with no free parameters to tune.

7.4.2 View adaptation from a single source

In this experiment, we demonstrate the effectiveness of the proposed approach when the distributions between source and target domain ($0°$, $-15°$ and $-30°$) differ in an increasing non-linear manner. For this purpose we evaluate all considered algorithms in terms of their ability to perform accurate emotion classifiaction as we move away from the source pose. Example images for the specified task can be seen in Fig. 7.3. Notice that the weighted version of our method, *i.e.*, wGPDE is not evaluated on the current experiment since the emotion analysis is an intrinsic single output problem, and hence, there are no additional variances to be modeled. Furthermore, in this scenario we only considered the geometric features as inputs to the compared models since in Chapter 4 they have been proved efficient to model the global phenomena of the facial expressions.

Table 7.1 summarizes the results. The generic classifier GP_{source} exhibits the lowest performance, due to the fact that it has only been trained on source domain images. It is important to note the fluctuations in the classification rate when the source and target domain vary. We can clearly see that when the frontal pose, *i.e.*, $0°$ acts as the source domain, the symmetric nature of the face helps towards achieving a satisfactory performance on the target domains. Yet, the performance degrades when the symmetry is severely violated, *e.g.*, $0° \rightarrow -30°$. On the other hand when $-15°$ and $-30°$ serve as the source domain, these symmetric attributes cannot be

uncovered from the generic GP_{source}. Hence, we observe a significantly low performance for the target frontal view (around 55%). The above results clearly indicate the inefficiency of a generic classifier to deal with data of different characteristics.

On the other hand, the GP_{target} when trained with as few as 30–50 data points, in most of the cases, achieves similar performance to the GP_{source} since it benefits from modeling domain-specific attributes. A further increase of the cardinality of the target training data results in a significant improvement in the classification rate. This is even more pronounced in the scenario we have illustrated above, *i.e.*, the target frontal view. As we can see the generic classifier when trained on the 0° can reach the CR of 84.06%, compared to the achieved 53.82% and 56.56% when trained on -15° and -30°, respectively.

A similar trend can be observed in the performance of the adaptation methods, where the inclusion of 10–30 labeled data points from the target domain is adequate to shift the learned source classifier towards the distribution of the target data. The GPA uses the extra data to condition on the generic classifier GP_{source} and increase its prediction performance. Thus, it can reach its highest performance in situations where the generic classifier GP_{source} is already sufficient for the task of emotion classification (*i.e.*, -15° and -30°). ATL-DGP on the other hand facilitates a joint learning scheme where GP_{source} and GP_{target} are fused together in an intermediate latent space, via conditioning, in a deep architecture. The advantage of the latter is evidenced by the highest achieved accuracy in the situations where the source classifier performs averagely, *i.e.*, $0^\circ \rightarrow -30^\circ$, $-15^\circ \rightarrow 0^\circ$ and $-30^\circ \rightarrow 0^\circ$ for $N_t = 10$–50. However, the joint training scheme of ATL-DGP limits its adaptation ability, due to the high effect of the source prior. Consequently, its performance saturates and cannot reach that of the generic classifier GP_{target} for $N_t > 100$. A further disadvantage of ATL-DGP's joint learning is that it requires retraining of both source and target classifiers every time the target distribution changes.

An opposite pattern compared to ATL-DGP can be observed in the performance of both MMDT and KBTL. Both of these methods achieve, to some extent, to reach the accuracy of the generic GP_{target} classifier, when more and more target data become available. On the contrary their performance is problematic when dealing with quite few labeled target data, *i.e.*, $N_t < 50$. In such cases, the parametric[1] nature of MMDT does not allow for effective learning of the projections from the target to the source domain, and hence, the learned classifier fails to poor results. Similarly, KBTL cannot recover accurate projections from the target domain data to a low-dimensional space. The latter has a negative impact on the accuracy of KBTL.

[1] Parametric models require lots of data for their accurate training.

Table 7.1: Average classification rate across 5-folds on MultiPIE. The view adaptation is performed with increasing cardinality of labeled target domain data ($10 - 1200$).

	Target	$-15°$								$-30°$							
	N_t	10	30	50	100	200	300	600	1200	10	30	50	100	200	300	600	1200
Source $0°$	GP_{source}	81.65								76.94							
	GP_{target}	55.85	81.19	84.59	89.61	90.66	91.31	91.57	97.26	51.99	76.09	81.97	86.48	88.57	89.75	**92.16**	**98.43**
	GPA [112]	82.36	84.00	85.37	88.63	90.20	91.51	93.79	96.15	77.73	79.82	81.65	85.43	87.79	87.72	89.29	93.01
	ATL-DGP [96]	**83.32**	86.34	85.22	85.62	85.16	86.42	86.53	87.80	**79.82**	**82.93**	83.36	85.53	82.08	84.32	80.03	83.04
	MMDT [83]	21.75	66.88	82.63	88.11	89.81	91.25	90.73	90.46	27.37	71.39	80.47	86.48	87.59	88.70	89.16	90.53
	KBTL [69]	41.67	69.11	72.57	85.63	87.98	89.61	91.18	97.19	34.36	62.44	66.62	81.71	84.91	86.35	89.55	95.62
	GPDE	82.95	**86.35**	**87.52**	**92.10**	**93.73**	**94.64**	**95.36**	**97.84**	78.71	82.17	**84.65**	**87.85**	**88.83**	**90.01**	91.38	96.86

	Target	$0°$								$-30°$							
	N_t	10	30	50	100	200	300	600	1200	10	30	50	100	200	300	600	1200
Source $-15°$	GP_{source}	53.82								85.70							
	GP_{target}	52.91	61.27	64.60	71.96	**77.53**	**79.10**	**81.84**	**84.06**	51.99	76.09	81.97	86.48	88.57	89.75	92.16	**98.43**
	GPA [112]	55.00	57.67	59.70	63.10	65.51	68.26	72.83	78.31	88.37	92.16	93.21	93.86	94.45	**94.97**	**95.30**	97.52
	ATL-DGP [96]	**70.11**	**73.20**	**71.15**	72.21	73.48	74.68	74.33	73.41	78.33	79.95	82.68	85.12	83.79	86.16	85.28	86.08
	MMDT [83]	17.37	42.91	63.03	71.72	72.44	74.98	78.18	79.23	11.93	63.10	86.54	90.27	89.55	90.40	89.03	86.81
	KBTL [69]	22.08	35.99	59.24	67.28	70.35	71.39	75.11	79.03	32.20	64.21	70.35	82.89	87.00	87.85	90.73	96.41
	GPDE	56.11	63.23	66.82	**72.37**	75.64	76.94	80.40	83.80	**88.44**	**93.40**	**94.32**	**93.99**	**94.84**	94.64	94.97	98.04

	Target	$0°$								$-15°$							
	N_t	10	30	50	100	200	300	600	1200	10	30	50	100	200	300	600	1200
Source $-30°$	GP_{source}	56.56								91.38							
	GP_{target}	52.91	61.27	64.60	71.96	**77.53**	**79.10**	**81.84**	**84.06**	55.85	81.19	84.59	89.61	90.66	91.31	91.57	97.26
	GPA [112]	57.41	59.83	61.53	64.53	67.15	69.24	75.11	77.60	93.27	94.58	94.72	95.43	**95.89**	**96.54**	**96.47**	97.91
	ATL-DGP [96]	**70.13**	**75.38**	**73.45**	**74.79**	74.68	75.51	67.61	73.17	83.52	84.21	84.94	85.02	87.90	85.80	86.12	88.35
	MMDT [83]	20.77	46.11	60.81	69.76	72.63	76.55	78.71	79.69	23.97	72.11	86.41	92.36	92.36	92.68	93.08	92.42
	KBTL [69]	22.08	35.60	59.37	67.60	70.15	71.06	74.85	78.18	40.10	68.26	75.38	87.72	89.42	90.01	91.70	97.58
	GPDE	59.57	65.58	69.56	72.57	75.96	77.86	81.45	83.61	**93.60**	**94.64**	**94.84**	**94.58**	94.51	94.25	93.60	**98.37**

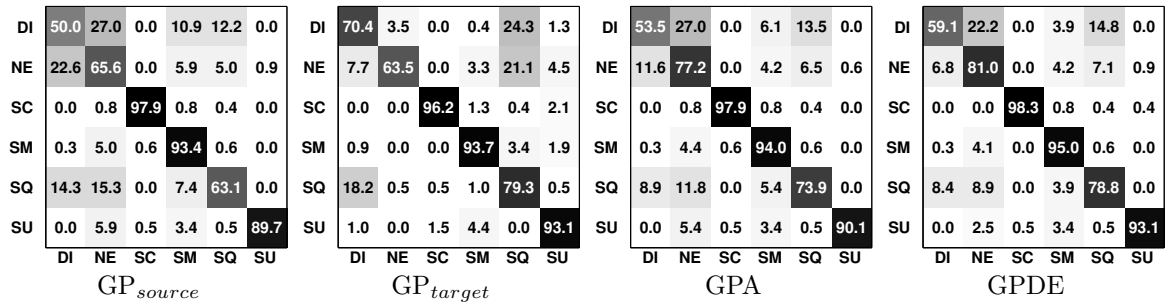


Figure 7.4: Confusion matrices averaged across the folds when using 50 target training data for $0° \rightarrow -30°$ adaptation.

Finally, the proposed GPDE, exhibits the most stable performance for varying cardinality of labeled target data. This can be attributed to the fact that it uses the notion of experts to unify GP_{source} and GP_{target} into a single classifier. To achieve so, GPDE measures the confidence of the predictions from each expert (by means of predictive variance), in contrast to GPA (uses source expert only) and ATL-DGP (uses an uninformative prior). This property of GPDE is more pronounced in the highly non-linear adaptation scenarios of $0° \rightarrow -30°$, $-30° \rightarrow 0°$ and $-15° \rightarrow 0°$ for $N_t > 200$, where GP_{target} achieves the highest classification ratio. GPDE performs similarly to the target expert while, GPA and ATL-DGP underestimate the prediction capacity of the target-specific classifier, and thus, attain lower results. The only situations where GPDE achieves inferior performance are the cases where GP_{source} performs poorly. Thus, as expected, GPDE cannot attain a reliable adaptation without having access to latent factors, opposed to ATL-DGP.

A better insight into the performance of the considered methods can be obtained from the confusion matrices in Fig. 7.4. The reported results are for $0° \rightarrow -30°$ adaptation with $N_t = 50$ (at which point the GP_{target} starts outperforming GP_{source}). The proposed GPDE takes advantage of the target-specific expert and significantly reduces the confusion between the subtle expressions of Disgust and Squint with the Neutral face.

7.4.3 Subject adaptation from multiple sources

In this section, we evaluate the models in a multi-label classification scenario, where the adaptation is performed from multiple source domains. This is also a natural setting to exhibit the importance of modeling different variances per output dimensions with the proposed wGPDE. In contrast to the view adaptation scenario for emotion classification, herein we report results for both geometric and appearance features, since different AUs are better explained from different type of features.

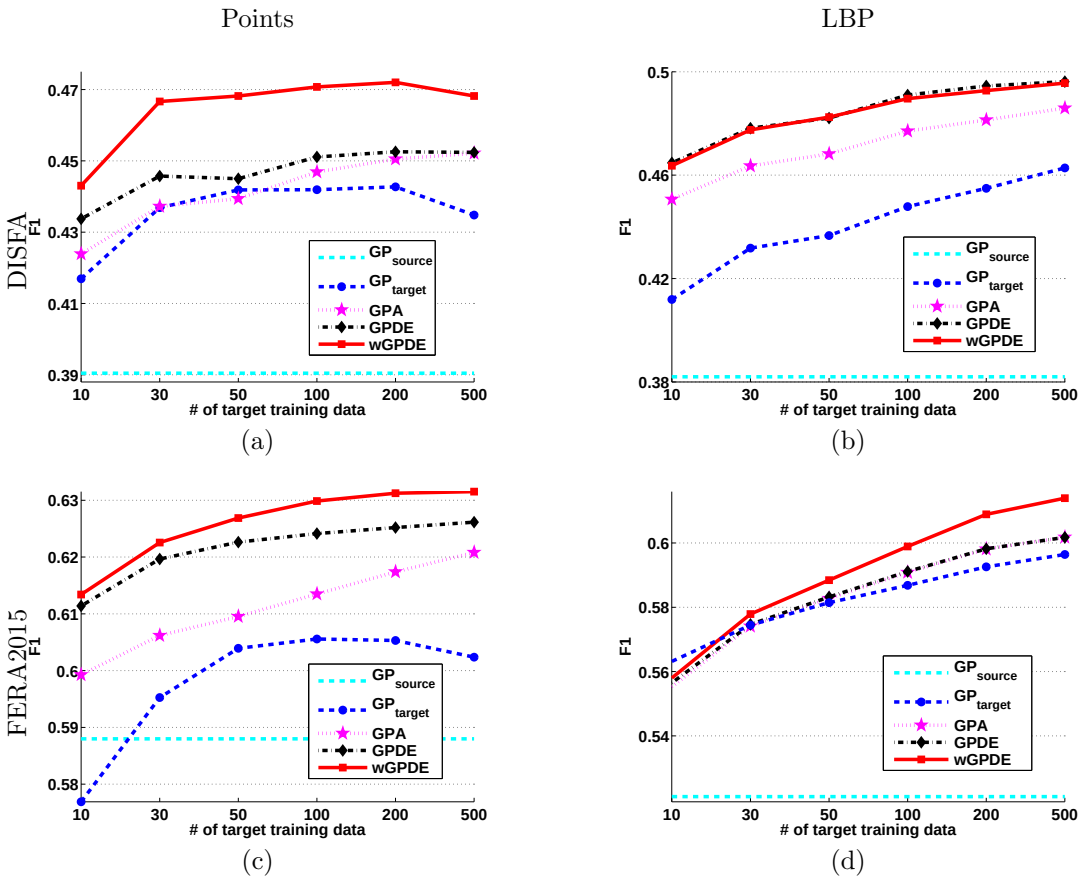


Figure 7.5: Average F1 score for joint AU detection with subject adaptation on DISFA (top) and FERA2015 (bottom) with increasing number of target domain data. The results are reported when using geometric (left) and appearance (right) features.

Overall, this is a more challenging setting, since the datasets are comprised of naturalistic facial expressions, and the recorded subjects are experiencing the affect in different ways and levels. The difficulty of the task can be seen in Fig. 7.5, where the subject-specific classifier GP_{target}, trained with 10–30 labeled data points, achieves a higher average F1 score than the generic classifier GP_{source}, which is trained on all available source subjects. The importance of this outcome gets more clear if we consider that it holds for both DISFA and FERA2015, when using either geometric or appearance features. This suggests that, no matter the nature of the inputs, personalized AU detectors are superior to generic classifiers, even when limited data are available. Another factor that is worth mentioning is that the average results are obtained over a large set of AUs (*i.e.*, 12 AUs for DISFA and 11 AUs for FERA2015). This fact, not only constitutes the results more reliable, but it also implies that even a small increase in the average performance (*e.g.*, 1-2%) can be attributed to an improved performance over several AUs.

By continuing our analysis of Fig. 7.5 we observe that the adaptation models, *i.e.*, GPA, GPDE and wGPDE achieve superior F1 score compared to the generic GP_{target}, under all scenarios. The latter implies that images from source and target subjects contain complementary information regarding the depicted facial expressions. Hence, the target classifier does not consist anymore an upper bound limit for the adaptation. This can be explained from the multi-modal nature of the problem , since we can have different AU combinations per sequence, contrary to the universal expressions appearing in the view adaptation scenario. Thus, expressions that are present only on the source sequences, can be used to improve the AU detection task for the target subject. The proposed GPDE and wGPDE benefit from modeling the target-specific information and can attain a better adaptation compared to GPA. Another reason for the difference in the performance between the proposed model and GPA is that the latter treats all training subjects as data from a single, *broader*, source domain. Hence, GPA smooths out the individual differences and lessens the contribution of the target domain, as the variations of the target data can be explained, on average, by the source domain.

Finally, the importance of modeling individual variances becomes clear by comparing the attained scores from wGPDE and GPDE. In 3 out of 4 scenarios, wGPDE achieves superior performance with more pronounced results appearing in DISFA dataset when geometric features are used (see Fig. 7.5(a)). On the other hand, when appearance features are used, as we can see in Fig. 7.5(b) both wGPDE and GPDE perform similarly. This can be explained from the fact that images from DISFA are not of high resolution. Hence, the local patches cannot explain adequately all the important variations that differ among the various outputs (*i.e.*, AUs). However, as we can see in Fig. 7.5(d) this is not the case with the high-resolution images from FERA2015. The input appearance features are of better quality, and thus, wGPDE can more accurately model the individual variances per output and attain higher scores.

For a deeper understanding of the efficacy of the adaptation task, in Tables 7.2–7.3 we report the detailed results (F1 score and AUC) per AU for the case of $N_t = 50$. Note that the setting of $N_t = 50$ is not always the most beneficial for our proposed approach. In most scenarios the gap in the performance between (w)GPDE and the other methods increases as we include more target data. However, we selected to demonstrate the performance on $N_t = 50$ because AU annotations are expensive and laborious. Thus, such a setting is a more reasonable choice for adaptation for the current task. The proposed (w)GPDE under the current setting, and using the geometric features as input (upper half of Tables 7.2–7.3), attains an average F1 improvement on both DISFA and FERA2015 of 3% and 2%, respectively. This small increase in the average performance translates to an improved F1 score on 9/12 and

Table 7.2: F1 score and AUC for joint AU detection on DISFA. Subject adaptation with $N_t = 50$.

		Dataset	DISFA											
		AU	1	2	4	5	6	9	12	15	17	20	25	26	Avg.
Points	F1	GP_{source}	33.1	31.6	54.8	10.5	44.8	31.6	57.3	24.4	35.8	13.7	**79.5**	51.5	39.0
		GP_{target}	37.2	41.4	62.2	21.7	57.3	30.2	59.3	25.9	38.3	**20.5**	76.0	60.1	44.2
		GPA [112]	36.0	37.2	62.4	21.3	52.7	36.4	**67.3**	**27.1**	38.7	16.2	77.1	54.8	43.9
		GPDE	36.8	38.3	**63.2**	22.7	54.3	**36.8**	66.4	26.8	38.9	16.5	77.4	55.9	44.5
		wGPDE	**41.2**	**52.9**	61.7	**25.3**	**60.9**	32.8	58.8	**27.1**	**40.7**	16.7	77.6	**65.2**	**46.8**
	AUC	GP_{source}	71.3	73.2	64.1	56.3	70.7	71.8	77.3	61.6	65.7	57.4	80.2	67.7	68.2
		GP_{target}	72.6	77.2	75.2	63.3	81.6	66.8	75.7	61.3	69.0	**69.3**	77.8	74.3	72.0
		GPA [112]	74.9	76.8	75.3	68.1	79.9	73.7	81.2	66.3	71.1	63.1	79.7	73.6	73.6
		GPDE	**75.5**	77.6	**76.2**	68.3	81.2	**73.9**	**81.3**	**66.4**	**71.5**	63.8	80.3	74.6	74.2
		wGPDE	73.7	**83.2**	75.0	**71.4**	**82.9**	72.3	77.0	64.2	70.6	60.8	**80.4**	**79.4**	**74.3**
LBP	F1	GP_{source}	31.0	27.0	52.2	11.7	35.5	29.3	52.4	31.1	38.6	23.8	73.4	52.4	38.2
		GP_{target}	35.4	40.9	58.7	10.5	55.4	30.6	56.2	28.9	40.7	23.0	79.7	64.1	43.7
		GPA [112]	38.5	37.3	63.4	13.6	62.0	32.4	63.8	30.9	44.9	24.4	83.1	67.7	46.8
		GPDE	39.8	41.1	65.1	17.2	**62.2**	**34.5**	**64.3**	32.5	44.9	**25.5**	**83.4**	**68.2**	**48.2**
		wGPDE	**41.0**	**41.8**	**65.6**	**20.8**	60.7	34.1	60.9	**34.5**	**46.3**	24.4	82.1	66.7	**48.2**
	AUC	GP_{source}	67.2	66.4	57.3	66.3	60.2	68.7	69.7	68.6	69.4	73.6	75.2	68.7	67.6
		GP_{target}	75.8	77.9	71.1	60.8	81.3	71.8	75.0	68.3	72.1	71.5	84.0	80.4	74.2
		GPA [112]	78.3	80.0	77.5	70.2	84.4	73.2	81.4	72.1	75.4	74.9	88.2	83.0	78.2
		GPDE	79.7	**82.2**	79.6	76.1	**84.5**	75.2	**82.3**	74.6	75.4	**75.3**	**88.5**	**83.4**	79.7
		wGPDE	**80.4**	82.1	**81.0**	**79.4**	83.7	**75.3**	80.2	**76.1**	**76.0**	73.9	87.4	82.0	**79.8**

8/11 AUs, respectively. The robustness on the results of (w)GPDE is further supported by both per AU and average AUC. We can see that (w)GPDE achieves higher AUC even in the AUs that reports inferior F1 score, resulting in 11/12 and 10/11 improved AUs on DISFA and FERA2015, respectively. Thus, it is evidenced that the proposed (w)GPDE constitutes a more reliable classifier, under the current settings. Regarding the appearance features (lower half of Tables 7.2–7.3) the average improvement of (w)GPDE is marginal, especially on FERA2015 dataset. Yet, if we look again individually each AU, we observe that the proposed model attains increased F1 score on 12/12 (12/12 in terms of AUC) and 7/11 (12/12 in terms of AUC), on DISFA and FERA2015, respectively.

By comparing wGPDE to GPDE we can further observe that modeling of individual variances results in improved average performance, which translates to an improvement on certain AUs. An indicative example is the increase in F1 score of AUs $1, 2, 5, 6$ on DISFA dataset, especially when using the geometric features. On all these 4 AUs, the standard GPDE fails to reach the performance of the generic GP_{target} classifier. However, the proposed weighting allows the GPDE to model output-specific attributes, or 'pair' the variances that are associated with co-occurring outputs, *e.g.*, AUs $1, 2$. Similar pattern can be observed in the results for AU2, for geometric, and AUs $2, 4, 6$, for appearance features on FERA2015. Especially for AUs $4, 6$ the increase in F1 score is further supported by an increase in AUC of 2% and 4%,

Table 7.3: F1 score and AUC for joint AU detection on FERA2015. Subject adaptation with $N_t = 50$.

		Dataset	FERA2015											
		AU	1	2	4	6	7	10	12	14	15	17	23	Avg.
Points	F1	GP_{source}	49.5	34.5	57.9	73.9	77.2	79.5	82.2	62.6	32.1	60.2	37.2	58.8
		GP_{target}	43.4	38.5	53.3	72.2	78.3	83.7	80.7	64.6	**48.5**	60.8	41.0	60.5
		GPA [112]	**54.6**	37.8	**60.4**	74.9	77.9	81.5	83.1	64.6	34.7	61.4	39.7	61.0
		GPDE	52.6	38.8	57.8	**75.7**	**79.2**	**84.9**	**84.5**	**65.9**	39.1	65.2	40.7	62.3
		wGPDE	53.4	**41.2**	58.5	75.1	79.0	84.2	83.4	65.6	40.9	**65.7**	**43.1**	**62.7**
	AUC	GP_{source}	75.5	65.9	81.5	81.5	68.9	76.1	85.9	66.7	57.5	68.5	65.6	72.1
		GP_{target}	67.6	68.9	77.0	76.5	73.1	82.6	79.1	70.8	73.2	68.6	68.1	73.2
		GPA [112]	**79.1**	68.7	83.4	83.0	72.2	81.4	87.1	70.1	63.3	69.8	68.5	75.1
		GPDE	72.7	69.3	83.2	**83.3**	**76.7**	85.5	**88.4**	**73.7**	68.6	**75.2**	70.5	77.0
		wGPDE	74.1	**70.6**	**83.6**	82.7	76.6	**85.6**	87.6	**73.7**	**71.0**	74.9	**72.2**	**77.5**
LBP	F1	GP_{source}	35.8	29.9	36.0	63.3	75.8	78.1	73.1	60.5	30.6	58.0	32.1	52.1
		GP_{target}	**41.6**	36.4	48.1	64.9	**78.0**	80.9	74.7	63.0	**50.0**	58.8	43.2	58.1
		GPA [112]	41.2	36.5	46.8	66.9	77.4	80.3	76.8	62.6	47.6	60.1	**44.7**	58.3
		GPDE	41.4	36.6	47.0	66.8	77.4	80.5	76.7	62.6	47.7	60.1	44.7	58.3
		wGPDE	41.4	**37.3**	**48.7**	**68.6**	77.6	**81.6**	**77.6**	**63.2**	47.4	**60.6**	44.4	**58.9**
	AUC	GP_{source}	56.3	58.5	54.0	41.5	47.2	40.4	42.3	47.8	51.5	47.5	55.3	49.3
		GP_{target}	65.4	65.3	72.3	62.6	71.5	75.0	63.5	68.6	76.0	62.8	71.0	68.5
		GPA [112]	66.8	65.9	72.6	71.1	73.1	77.6	74.2	69.5	74.0	65.5	72.0	71.1
		GPDE	65.9	**66.6**	**74.7**	**74.7**	**73.6**	**79.6**	**77.4**	**70.3**	73.9	**66.9**	71.9	**72.3**

Table 7.4: F1 score for joint AU detection on DISFA. Comparison to state-of-the-art. Subject adaptation for wGPDE has been performed with $N_t = 50$.

Dataset	DISFA												
AU	1	2	4	5	6	9	12	15	17	20	25	26	Avg.
wGPDE (pts.)	**41.2**	**52.9**	61.7	**25.3**	**60.9**	32.8	58.8	27.1	40.7	16.7	77.6	65.2	46.8
wGPDE (app.)	41.0	41.8	**65.6**	20.8	60.7	34.1	60.9	34.5	**46.3**	24.4	82.1	66.7	**48.2**
dynSVM [11]	30.0	26.0	34.0	16.0	45.0	**45.0**	**77.0**	**47.0**	41.0	**25.0**	**84.0**	**75.0**	48.0
CPM [200]	29.5	24.8	56.8	–	41.7	31.5	71.9	–	–	–	81.6	51.3	–

Table 7.5: F1 score for joint AU detection on FERA2015. Comparison to state-of-the-art. Subject adaptation for wGPDE has been performed with $N_t = 50$.

Dataset	FERA2015											
AU	1	2	4	6	7	10	12	14	15	17	23	Avg.
wGPDE (pts.)	**53.4**	**41.2**	**58.5**	75.1	**79.0**	84.2	83.4	65.6	40.9	**65.7**	43.1	**62.7**
wGPDE (app.)	41.4	37.3	48.7	68.6	77.6	81.6	77.6	63.2	**47.4**	60.6	44.4	58.9
dynSVM [11]	43.0	39.0	46.0	**77.0**	77.0	**85.0**	**87.0**	**67.0**	44.0	62.0	**45.0**	61.0
CPM [200]	46.6	38.7	46.5	68.4	73.8	74.1	84.6	62.2	44.3	57.5	41.7	58.0

respectively.

We next compare the proposed (w)GPDE to state-of-the-art models from the literature of AU analysis that attempt to perform the adaptation. For the purposes of this experiment, and in order to have fair comparisons, we do not include unsupervised models that do not use the available labels of the target data. Thus, we compare to the supervised dynSVM [11]

and the semi-supervised CPM [200].[2] dynSVM attempts to perform the adaptation at the feature level (combination of geometric and appearance features), where the input data from each subject (domain) are normalized by removing the dynamics of the expression. CPM on the other hand tries to adjust the classifier to the target domain. It achieves so by taking into account the confidence/agreement in the predictions of source soft classifiers, when assessing the target data.

Tables 7.4–7.5 summarize the results. At first we can see that the proposed wGPDE outperforms both dynSVM and CPM on both DISFA and FERA2015. The improvement over dynSVM on DISFA is marginal. However, the authors in [11], before applying the dynSVM, attempted to re-balance the data in order to account for the mismatch in the distribution of activated AUs. This explains the superior performance of dynSVM on less frequently occurring AUs, *i.e.*, AUs $9, 15, 20$ on DISFA and AUs, $14, 23$ on FERA2015. On the other hand, CPM reports lower results, both on average and per AU, on both datasets. This is partly attributed to the fact that CPM is a semi-supervised method and uses soft labels (*i.e.*, the predictions of the source classifier) as ground truth labels for the target data during training. Another reason for its low performance is the 'virtual' way that CPM utilizes in order to measure the confidence. In contrast, the proposed wGPDE has a well determined probabilistic way to correctly estimate the confidence in the predictions of the various experts. This allows the wGPDE to weight the contribution of each expert in the final classification, which results in more accurate predictions.

7.4.4 Assessing the confidence in the predictions

Herein, we assess the ability of (w)GPDE to measure the confidence in the output labels, by means of predicted variance. As an evaluation measure we use the negative log-predictive density (NLPD). It is a measure commonly used in probabilistic models, since apart from the predictive mean it also takes into account the predictive variance. In Fig. 7.6 we see the NLPD for the baseline generic classifiers, *i.e.*, GP_{source} and GP_{target}, as well as the proposed (w)GPDE, on both DISFA and FERA2015 datasets. First of all we observe that all the models (apart from the GP_{target} on DISFA) increase their variance in the predictions (NLPD is increasing), as we include more training target data. This is expected since by increasing the training set, we observe more variations in the input data (different AU combinations). Hence, the predicted variance in the outputs also increases. In the case of DISFA, (Fig. 7.6(left)) the target expert becomes more confident for $N_t > 10$. We attribute this to the nature of the

[2]Note again that implementations to the current algorithms are not available, and hence, the results are directly taken from the corresponding papers and the authors' websites.

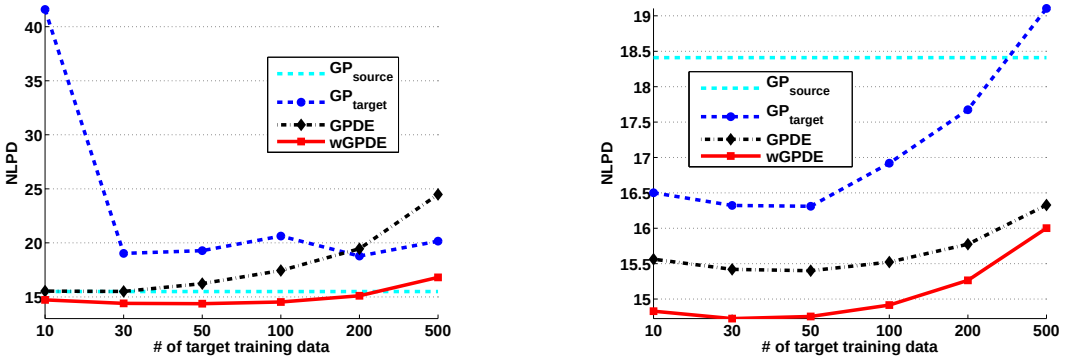


Figure 7.6: Quantification of the confidence in the probabilistic predictions in terms of NLPD for DISFA (left) and FERA2015 (right) with increasing number of target domain data.

videos in DISFA, which contain less frequently varying expressions over time. Thus, the generic personalized classifier has seen most of the available variations – on average – which results in reduced uncertainty. On the other hand, the events on FERA2015 are shorter, hence, more frequent variations. Thus, the relevant NLPD at first decreases, but as more data become available (more AU combinations) the uncertainty increases. Eventually, in both situations the generic GP_{target} becomes less confident than GP_{source}.

When we compare GPDE to wGPDE we observe a similar behavior between the two. However, GPDE without the weighting can only produce a single variance for all outputs. This has a negative impact on the NLPD, since the model is equally confident for all the outputs. Thus, GPDE results in being confident even for false predictions. On the other hand, the extra weighting term allows the wGPDE to produce different variance for each predicted output.

The above claims for the difference between GPDE and wGPDE are better explained from Fig. 7.7. In Fig. 7.7 (upper) we see an example where both GPDE and wGPDE predict the exact same labels (almost the same predicted means). However, GPDE (Fig. 7.7 (left)) suffers from heavier tails. This results in less accurate estimation of the mass probability for AUs $1, 2, 10, 12$, which can be interpreted by also a higher NLPD. The same behavior of heavier tails can be observed in another example in Fig. 7.7 (lower). However, now GPDE and wGPDE disagree on their predictions for AUs $6, 17$. wGPDE can better estimate the probability mass for the quite uncertain AUs $6, 17$, which results in their correct prediction compared to the unweighted GPDE.

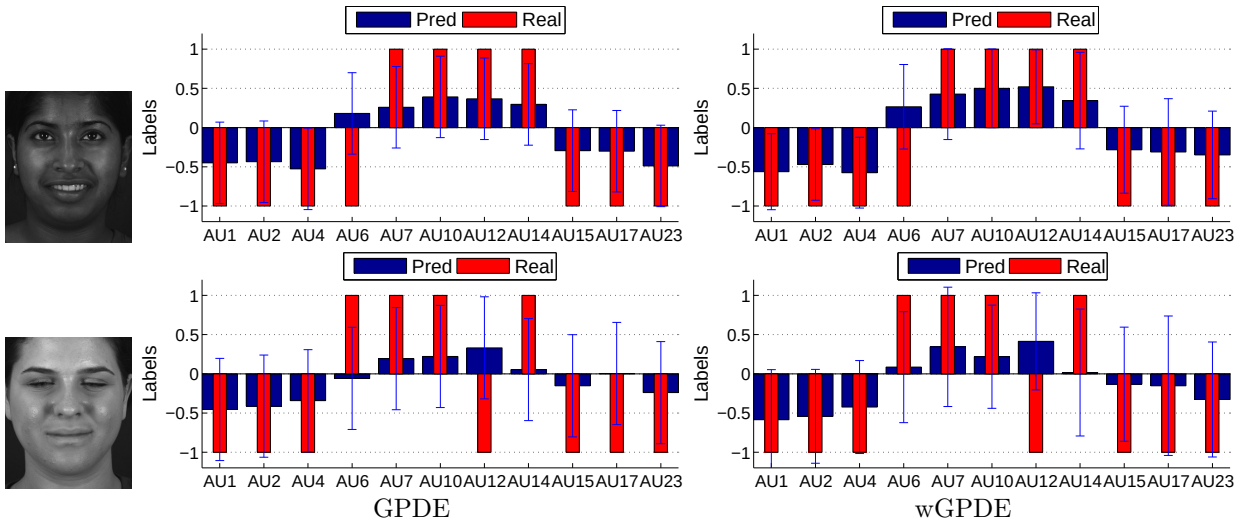


Figure 7.7: Probabilistic prediction of joint AU activations on FERA2015 from GPDE (left) and wGPDE (right). The reported tails account for the predicted standard deviation. Shorter tails correspond to more confident predictions. Both GPDE and wGPDE are trained with $N_t = 50$.

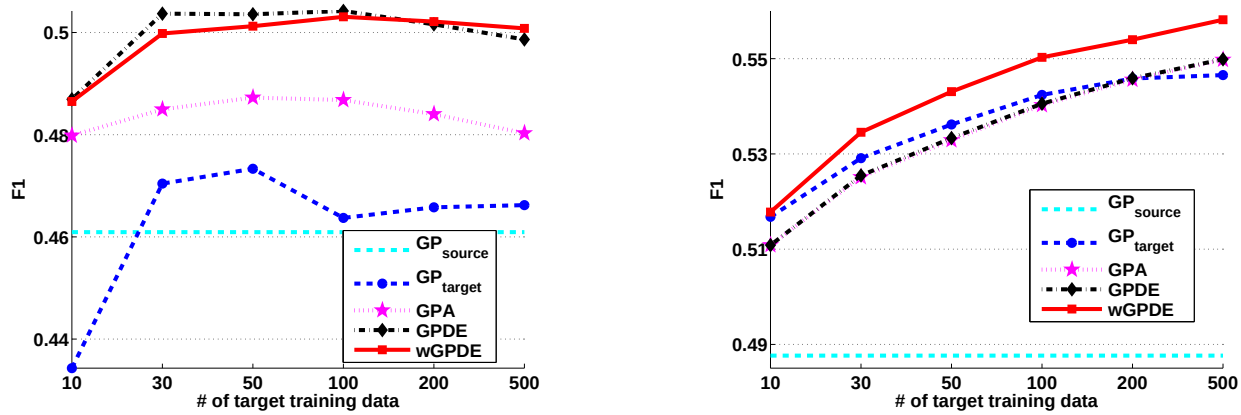


Figure 7.8: Cross-dataset evaluations. Average F1 score of the 7 common AUs present in both DISFA and FERA2015 datasets. The models are trained on data from FERA2015 and tested on data from DISFA (left), and the other way around (right). The reported results are obtained with geometric features and increasing cardinality of labeled target domain data.

7.4.5 Cross dataset adaptation

In this section, we evaluate the robustness of the models in a cross dataset experiment. Specifically, we perform two different cross-dataset experiments, FERA2015 → DISFA and DISFA→FERA2015.[3] We evaluate the models' performance on the 7 AUs (*i.e.*, $1, 2, 4, 6, 12, 15, 17$) that are present in both datasets. For the purposes of this experiment

[3]'A→B' denotes the training on dataset A and testing on dataset B.

Table 7.6: Cross-dataset evaluations on 7 AUs present in both DISFA and FERA2015 datasets. The models are trained on data from FERA2015 dataset and tested on data from DISFA dataset (F → D), and the other way around (D → F). Subject adaptation with $N_t = 50$.

		F1								AUC							
	AU	1	2	4	6	12	15	17	Avg.	1	2	4	6	12	15	17	Avg.
F → D	GP_{source}	**44.0**	43.9	56.4	49.1	54.8	28.9	45.6	46.1	77.3	81.0	65.2	73.7	72.5	66.4	**75.4**	73.1
	GP_{target}	39.2	46.4	58.2	61.0	57.3	**29.6**	39.7	47.3	74.4	81.8	70.8	81.1	73.0	65.8	68.0	73.6
	GPA [112]	41.3	44.7	61.9	57.2	62.9	28.7	44.4	48.7	78.3	80.7	74.6	82.0	79.4	67.6	73.5	76.6
	dynSVM [11]	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	GPDE	41.8	44.8	**63.9**	**61.7**	**66.5**	28.1	**45.8**	**50.4**	79.1	**81.9**	**76.5**	**85.0**	**82.4**	67.6	75.1	**78.2**
	wGPDE	43.4	**46.9**	62.4	61.5	63.9	**29.6**	43.2	50.1	**80.4**	81.7	75.1	84.5	80.3	**68.6**	73.2	77.7
D → F	GP_{source}	37.3	28.0	46.5	63.8	74.1	31.6	60.1	48.8	61.1	55.5	71.7	64.8	74.9	50.9	61.9	63.0
	GP_{target}	41.1	**37.5**	47.0	67.5	77.0	**45.8**	59.4	53.6	67.0	**66.4**	71.7	68.1	69.3	71.1	63.7	68.2
	GPA [112]	40.7	36.3	50.6	68.0	76.9	39.7	60.8	53.3	**67.3**	65.2	74.6	72.8	76.0	69.0	66.2	70.2
	dynSVM [11]	44.0	34.0	50.0	68.0	67.0	26.0	48.0	48.0	–	–	–	–	–	–	–	–
	GPDE	40.7	36.4	50.5	68.0	77.0	40.0	60.7	53.3	**67.3**	65.3	74.6	72.7	75.8	**69.2**	66.2	70.2
	wGPDE	**42.1**	35.9	**54.7**	**69.2**	**79.5**	36.9	**62.0**	**54.3**	66.3	64.3	**79.5**	**76.5**	**83.6**	66.5	**69.6**	**72.3**

we employ the geometric features, since the images from the two datasets differ significantly in resolution. However, even the geometric features are being affected by factors, such as, facial pose and size. This imposes a further difficulty on the alignment of the input facial features.

By analyzing the results in Fig. 7.8 we can draw two quick conclusions. First, FERA2015 is a more representative dataset for the task of AU detection. The generic classifier GP_{source} in Fig. 7.8 (left) achieves similar performance to the adaptation models in Fig. 7.5(a). This does not hold for the generic GP_{source} in the DISFA → FERA2015 experiment. The latter is further supported by the performance of GP_{target} which significantly outperforms the generic GP_{source} on the DISFA → FERA2015 adaptation. The second finding is related to the advantage of the joint modeling of the AUs. This is illustrated in the performance of the generic GP_{target} in both cross-dataset evaluations. We can see that the average results are lower than the average of the corresponding AUs from Tables 7.2–7.3.

Regarding the performance of the adaptation methods we observe that in the FERA2015 → DISFA scenario, all the compared models benefit from the presence of the additional target domain data. More interestingly, (w)GPDE consistently outperforms GPA and reaches the average performance of the corresponding AUs in the within dataset evaluations from Table 7.2. The importance of wGPDE is not evidenced in this scenario. However, in the DISFA → FERA2015 adaptation, wGPDE manages to correctly model the individual variances in the target data, and hence, achieves better performance than the generic GP_{target} (contrary to the simple GPDE).

Finally, the detailed results per AU for the cross dataset adaptation are presented in

Table 7.6. It is clear that the proposed approach, not only outperforms its counterparts on the current experiment, but also manages to achieve improved performance on most of the AUs (on FERA2015 → DISFA), compared to the within dataset evaluations. This is an indicator of the quality of the achieved adaptation, since the model becomes less sensitive to the input source data. On the other hand, the subject normalization of dynSVM does not attain a sufficient adaptation, and hence, it fails to lower results than the generic GP_{source}.

7.5 Conclusions

To conclude, in this chapter we have presented a method that exploits successfully the non-parametric probabilistic framework of GPs to perform domain adaptation for both multi-class and multi-label classification of human facial expressions. In contrast to existing adaptation approaches, which leverage solely the source distribution during adaptation, the proposed approach defines a target expert to model domain-specific attributes, and reduce that way the effect of negative transfer. As a purely probabilistic model, (w)GPDE explores also the variance in the predictions. The latter consists an accurate measure of confidence, and as such, it can be used to reevaluate the predictions from the various experts, in order to achieve an improved classification performance.

Chapter 8

Discussion and Conclusions

In this thesis we have presented a variety of methodologies, all stemming from the well studied framework of Gaussian processes (GPs) [146], in order to address some of the important challenges that are commonly encountered in automated analysis of facial expressions. Our main goal, when originally discussing the direction of this thesis, was to propose novel algorithms and learning strategies that would have an impact on both the domains of affective computing, via advancing the current modeling practices in a more learning-oriented scheme, as well as the field of machine learning, via designing novel methodologies, general enough for being applicable to a variety of tasks.

We started in Chapter 4 by tackling the problem of multi-view and view invariant facial expression classification of basic emotions, and we showed how this challenge can be addressed in a multi-view learning strategy. We introduced the discriminative shared Gaussian process latent variable model (DS-GPLVM), which is proven to be effective on a variety of tasks, including multi-view and view-invariant facial expression classification of basic emotions, smile detection on spontaneous displayed expressions, as well as fusion of complementary modalities in a shared manifold for more accurate facial expression analysis. From a modeling perspective the main novelty achieved in DS-GPLVM is the back-constraining of the latent space from multiple views. This not only resulted in learning a manifold which reflects the structure from the multiple observation spaces, but it further allowed us to perform inference under different settings (*i.e.*, view-invariant and multi-view). We showed that DS-GPLVM considerably outperforms the existing approach to mutli-view facial exrpession analysis, while it is also capable of generalizing to new images captured in uncontrolled environments.

Since the framework of shared GPs has been proven to be effective, in Chapter 5 we focused on the fusion of multiple modalities, and we experimentally demonstrated its importance on

facial expression analysis, and in particular on the task of multiple AU detection. Specifically, we showed that combining the information of both geometric and appearance features resulted in a better descriptor which enhanced the detection task. Modeling-wise, contrary to the DS-GPLVM, we proposed a multi-conditional approach, where the fusion of the input features was concurrently learned with the output classifiers in a joint generative and discriminative framework. This approach gave us the opportunity to balance the contribution and the effect of the discriminative/generative attributes of the manifold during the learning of the multi-conditional latent variable model (MC-LVM). Nevertheless, the key property that resulted in the superior performance of MC-LVM was the induction of the label's structure to the manifold, in the form of the proposed constraints. Consequently, the detection of more subtle AUs, as demonstrated in our results, has been considerably improved by accounting for the co-occurrence patterns.

Motivated by the good results of the MC-LVM on the aforementioned problem, we further explored the effect of the feature fusion on analyzing the intensities of multiple AUs. Hence, in Chapter 6 we introduced the variational Gaussian process auto-encoder (VGP-AE), where we focused on how to model the ordinal structure of the output labels and impose it in the latent space. It is important to note that the structure of the data was automatically imposed on the learned manifold via a novel GP auto-encoder, without the need for additional constraints, as in the MC-LVM. Probabilistic sampling from VGP-AE generated meaningful facial expressions, demonstrating good generalization capabilities of VGP-AE and effectiveness of our structure learning algorithm in capturing higher-order dependencies among the high-dimensional input features and target AU intensities. The proposed approach is among the first that explored, and actually achieved, simultaneous feature fusion and joint AU intensity estimation in the context of facial behavior analysis. Furthermore, it is the first fully probabilistic auto-encoder in the GP-literature.

Finally, in Chapter 7 we exploited the primitives of domain adaptation to perform adaptation of two contextual factors: '*who*' (subject) and '*where*' (view). The work on domain adaptation in facial behavior analysis is still in its early stage. The conducted experiments on various adaptation scenarios indicate several interesting facts: the source classifier trained on a large number of data can easily be outperformed by the classifier trained on as few as 50 examples from the target domain. Furthermore, the existing adaptation approaches try to adapt the target domain to the source domain by assuming that the two distributions can be matched. Yet, when more target data become available, a generic target classifier can largely outperform the existing adaptation approaches. In our proposed Gaussian process domain experts (GPDE)

we tried to address these challenges by introducing the target expert, allowing it to reach (and outperform) the full performance of either source or target classifiers with as few as 50 target samples.

Taken together, the methods proposed in this thesis solve some of the most important challenges in the field of facial expression analysis. This research can serve as a basis and trigger further work in the field. Thus, it is our responsibility to note some of the limitations of our proposed algorithms and draw directions for future work. The main limitation of all the proposed approaches is the inefficiency to deal with large data during training. As purely based on the framework of GPs, training of the proposed alogrithms scales in $\mathcal{O}(N^3)$, which typically imposes a restriction on using datasets of size $\mathcal{O}(10^4)$. However, this can be addressed by sparse [170] or distributed [43] computations, which scale GPs to $\mathcal{O}(10^7)$. This would be of extreme importance, now that we have officially entered the era of *deep learning* with *big data*. Toward this extension, we can employ the notion of deep Gaussian processes [39] to model hierarchical layers of GPs, which allow for learning more complex structures in the intermediate manifolds.

Another promising improvement would be to include temporal information in the inference process. All models within this thesis operate only on static images. However, intuitively, facial expressions show a characteristic development over time (*e.g.*, we do not expect rapid jumps between pain and happiness) and thus information from the past and future would be valuable to infer the present. Thus, a temporal extension of our models based on temporal priors, as in [192, 38], would be likely to improve the recognition performance. An even more interesting direction to pursue, would be to consider the temporal dependencies within our proposed adaptation strategy. Ideally, we should be focusing on designing a general framework for adaptation, where we would exploit the remaining contextual factors (*i.e.*, 'when', 'why','what' and 'how'), simultaneously. It would be very interesting to explore how we could design a unified model, where interactions between the various contextual factors could be modeled in context-specific generative subspaces. The design of such a model would be an important step toward achieving a holistic analysis of facial expressions. It is our hope that the research presented in this thesis provides a small contribution towards accomplishing that goal.

Appendix A

Appendices

A.1 Derivatives for the DS-GPLVM

During the optimization of DS-GPLVM, we need to update $\boldsymbol{X}$ and $\boldsymbol{\theta}_s$ by solving the problem in Eq. (4.16). The latter is a sum of two terms, the negative log-likelihood given by Eq. (4.9), and the norm term which, for convenience, we denote as

$$\boldsymbol{C} = \frac{\mu_t}{2} \sum_{v=1}^{V} \| IBP(\mathbf{X}, \mathbf{A}_t^{(v)}) + \frac{\boldsymbol{\Lambda}_t^{(v)}}{\mu_t} \|_F^2 \tag{A.1}$$

Because of the likelihood term, the defined problem does not have an exact solution, and thus, we need to apply the conjugate gradient algorithm. Hence, we have to compute the gradients of Eq. (4.9),(A.1) w.r.t. the latent positions $\boldsymbol{X}$ and the kernel parameters $\boldsymbol{\theta}_s$

- $\frac{\partial L_s}{\partial \boldsymbol{X}} = \sum_v \frac{\partial L^{(v)}}{\partial \boldsymbol{X}} + \beta \tilde{\boldsymbol{L}} \boldsymbol{X}$
- $\frac{\partial L_s}{\partial \boldsymbol{\theta}_s} = \begin{bmatrix} \frac{\partial L^{(1)}}{\partial \boldsymbol{\theta}^{(1)}} & \cdots & \frac{\partial L^{(V)}}{\partial \boldsymbol{\theta}^{(V)}} \end{bmatrix}^T$
- $\frac{\partial \boldsymbol{C}}{\partial \boldsymbol{X}} = \sum_v \mu_t (\boldsymbol{X} - \boldsymbol{A}_t^{(v)}) + \boldsymbol{\Lambda}_t^{(v)}$

- $\frac{\partial \boldsymbol{C}}{\partial \boldsymbol{\theta}_s} = \mathbf{0}$.

The likelihood term $L^{(v)}$ is a function of the kernel $\boldsymbol{K}^{(v)}$, thus, we need to apply the chain rule in order to find the derivatives w.r.t $\boldsymbol{X}$ and $\boldsymbol{\theta}^{(v)}$

- $\frac{\partial L^{(v)}}{\partial x_{ij}} = \text{tr}\left[(\frac{\partial L^{(v)}}{\partial \boldsymbol{K}^{(v)}})^T \frac{\partial \boldsymbol{K}^{(v)}}{\partial x_{ij}}\right]$
- $\frac{\partial L^{(v)}}{\partial \theta_i^{(v)}} = \text{tr}\left[(\frac{\partial L^{(v)}}{\partial \boldsymbol{K}^{(v)}})^T \frac{\partial \boldsymbol{K}^{(v)}}{\partial \theta_i^{(v)}}\right]$
- $\frac{\partial L^{(v)}}{\partial \boldsymbol{K}_v} = \frac{D}{2}(\boldsymbol{K}^{(v)})^{-1} - \frac{1}{2}(\boldsymbol{K}^{(v)})^{-1}\boldsymbol{Y}_v\boldsymbol{Y}_v^T(\boldsymbol{K}^{(v)})^{-1}$.

Finally, the derivatives of the selected kernel are

- $\frac{\partial k^{(v)}(\boldsymbol{x}_i,\boldsymbol{x}_j)}{\partial \theta_1^{(v)}} = \exp(-\frac{\theta_2}{2}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2)$
- $\frac{\partial k^{(v)}(\boldsymbol{x}_i,\boldsymbol{x}_j)}{\partial \theta_2^{(v)}} = -\frac{\theta_1^{(v)}}{2}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 \exp(-\frac{\theta_2}{2}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2)$
- $\frac{\partial k^{(v)}(\boldsymbol{x}_i,\boldsymbol{x}_j)}{\partial \theta_3^{(v)}} = 1$
- $\frac{\partial k^{(v)}(\boldsymbol{x}_i,\boldsymbol{x}_j)}{\partial \theta_4^{(v)}} = -\frac{1}{(\theta_4^{(v)})^2}\delta_{i,j}$

and

$$\frac{\partial \boldsymbol{k}^{(v)}(\boldsymbol{x}_i)}{\partial x_{ij}} = \begin{bmatrix} -\theta_2(x_{ij} - x_{1j})\ k^{(v)}(\boldsymbol{x}_i, \boldsymbol{x}_1) \\ \vdots \\ -\theta_2(x_{ij} - x_{Nj})\ k^{(v)}(\boldsymbol{x}_i, \boldsymbol{x}_N) \end{bmatrix}$$

A.2 LOO solution of the regression step in ADMM

Herein, we derive the solution for the more general form of the IBP case. The same steps can be followed to arrive at the solution of the SBP case. The optimal values of parameters $\boldsymbol{A}^{(v)}$ are given by the solution of the linear equation:

$$(\boldsymbol{K}_{bc}^{(v)} + \frac{\lambda^{(v)}}{\mu_t}\boldsymbol{I})\boldsymbol{A}^{(v)} = (\boldsymbol{X} + \frac{\boldsymbol{\Lambda}_t^{(v)}}{\mu_t}). \tag{A.2}$$

The system of linear equations defined by Eq. (4.20) is insensitive to permutations of the ordering of the equations and the variables. Thus, at each iteration of the LOO, the i-th left out sample and the corresponding equation can be placed on top, without affecting the result. This enables us to define the matrix $\boldsymbol{M}$ as in Eq. (4.21). By placing $\boldsymbol{M}$ back in Eq. (4.20), we end up with the following linear system of equations:

$$\begin{bmatrix} m_{ii} & \boldsymbol{m}_i^T \\ \boldsymbol{m}_i & \boldsymbol{M}_i \end{bmatrix} \boldsymbol{A}^{(v)} = \begin{bmatrix} \boldsymbol{x}_i + \boldsymbol{\Lambda}_i^{(v)}/\mu_t \\ \boldsymbol{X}^{(-i)} + \boldsymbol{\Lambda}_{-i}^{(v)}/\mu_t \end{bmatrix} \tag{A.3}$$

Now, the solution of the parameters of the regression with the i-th sample excluded is

$$\boldsymbol{A}_{-i}^{(v)} = \boldsymbol{M}_i^{-1}(\boldsymbol{X}^{(-i)} + \frac{\boldsymbol{\Lambda}_{-i}^{(v)}}{\mu_t}),$$

and the LOO prediction of the i-th sample is given by

$$\begin{aligned}
\hat{\boldsymbol{x}}_i^{(-i)} &= \boldsymbol{m}_i^T \boldsymbol{A}_{-i}^{(v)} = \boldsymbol{m}_i^T \boldsymbol{M}_i^{-1}(\boldsymbol{X}^{(-i)} + \frac{\boldsymbol{\Lambda}_{-i}^{(v)}}{\mu_t}) \\
&= \boldsymbol{m}_i^T \boldsymbol{M}_i^{-1} \begin{bmatrix} \boldsymbol{m}_i & \boldsymbol{M}_i \end{bmatrix} \boldsymbol{A}^{(v)} \\
&= \boldsymbol{m}_i^T \boldsymbol{M}_i^{-1} \begin{bmatrix} \boldsymbol{m}_i & \boldsymbol{M}_i \end{bmatrix} \begin{bmatrix} \boldsymbol{A}_i^{(v)} \\ \boldsymbol{A}_{-i}^{(v)} \end{bmatrix} \\
&= \boldsymbol{m}_i^T \boldsymbol{M}_i^{-1} \boldsymbol{m}_i \boldsymbol{A}_i^{(v)} + \boldsymbol{m}_i^T \boldsymbol{A}_{-i}^{(v)}.
\end{aligned}$$

From Eq. (A.3) we have

$$\boldsymbol{x}_i + \frac{\boldsymbol{\Lambda}_i^{(v)}}{\mu_t} = \begin{bmatrix} m_{ii} & \boldsymbol{m}_i^T \end{bmatrix} \begin{bmatrix} \boldsymbol{A}_i^{(v)} \\ \boldsymbol{A}_{-i}^{(v)} \end{bmatrix} = m_{ii} \boldsymbol{A}_i^{(v)} + \boldsymbol{m}_i^T \boldsymbol{A}_{-i}^{(v)} \tag{A.4}$$

and thus, the error between the prediction $\hat{\boldsymbol{x}}_i^{(-i)}$ and the actual output $\boldsymbol{x}_i$ is

$$\begin{aligned}
\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i^{(-i)} &= (m_{ii} - \boldsymbol{m}_i^T \boldsymbol{M}_i^{-1} \boldsymbol{m}_i) \boldsymbol{A}_i^{(v)} - \boldsymbol{\Lambda}_i^{(v)} / \mu_t \\
&= \frac{\boldsymbol{A}_i^{(v)}}{[\boldsymbol{M}^{-1}]_{ii}} - \frac{\boldsymbol{\Lambda}_i^{(v)}}{\mu_t},
\end{aligned}$$

where on the last equation we used the Shur complement from the block matrix inversion lemma, and $\boldsymbol{M}_{ii}$ denotes the i-th diagonal element of the matrix $\boldsymbol{M}$. Finally, we end up with the cost of the LOO for all samples, E_{LOO}, as defined in Eq. (4.24). For the SBP case we follow exact the same steps, with the difference that we drop from all the equations the dependencies on the view v and we replace the $\boldsymbol{K}_{bc}^{(v)}$ with

$$\tilde{\boldsymbol{K}} = \sum_{v=1}^{V} w_v \boldsymbol{K}_{bc}^{(v)}.$$

Our final goal is to find the optimal parameters $\gamma^{(v)}$ and $\lambda^{(v)}$ that minimize the error of the LOO cross validation, defined by Eq. (4.24). For this, we need to calculate the derivatives of E_{LOO} w.r.t. $\gamma^{(v)}$ and $\lambda^{(v)}$. We first define the diagonal matrix

$$\boldsymbol{D} = \begin{bmatrix} \frac{1}{[\boldsymbol{M}^{-1}]_{11}} & & \\ & \ddots & \\ & & \frac{1}{[\boldsymbol{M}^{-1}]_{NN}} \end{bmatrix}$$

that allows us to reformulate Eq. (4.24) into

$$E_{LOO} = \frac{1}{2}\|\boldsymbol{D}\boldsymbol{A}^{(v)} - \frac{\boldsymbol{\Lambda}^{(v)}}{\mu_t}\|^2. \tag{A.5}$$

Using the chain rule, the derivatives of Eq. (A.5) are given by

$$\frac{\partial E_{LOO}}{\partial \lambda^{(v)}} = \mathrm{tr}\left[\left(\frac{\partial E_{LOO}}{\partial \boldsymbol{A}^{(v)}}\right)^T \frac{\partial \boldsymbol{A}^{(v)}}{\partial \lambda^{(v)}} + \left(\frac{\partial E_{LOO}}{\partial \boldsymbol{D}}\right)^T \frac{\partial \boldsymbol{D}}{\partial \lambda^{(v)}}\right]$$

and

$$\frac{\partial E_{LOO}}{\partial \gamma^{(v)}} = \mathrm{tr}\left[\left(\frac{\partial E_{LOO}}{\partial \boldsymbol{A}^{(v)}}\right)^T \frac{\partial \boldsymbol{A}^{(v)}}{\partial \gamma^{(v)}} + \left(\frac{\partial E_{LOO}}{\partial \boldsymbol{D}}\right)^T \frac{\partial \boldsymbol{D}}{\partial \gamma^{(v)}}\right],$$

while the detailed derivatives inside the trace terms are

- $\frac{\partial E_{LOO}}{\partial \boldsymbol{A}^{(v)}} = \boldsymbol{D}^T(\boldsymbol{D}\boldsymbol{A}^{(v)} - \frac{\boldsymbol{\Lambda}^{(v)}}{\mu_t})$
- $\frac{\partial E_{LOO}}{\partial \boldsymbol{D}} = \left[\boldsymbol{D}\boldsymbol{A}^{(v)}(\boldsymbol{A}^{(v)})^T - \frac{1}{\mu_t}\boldsymbol{\Lambda}^{(v)}(\boldsymbol{A}^{(v)})^T\right] \odot \boldsymbol{I}$
- $\frac{\partial \boldsymbol{A}^{(v)}}{\partial \lambda^{(v)}} = -\boldsymbol{M}^{-1}\frac{\partial \boldsymbol{M}}{\partial \lambda^{(v)}}\boldsymbol{M}^{-1}(\boldsymbol{X} + \frac{\boldsymbol{\Lambda}_t^{(v)}}{\mu_t}) = -\frac{1}{\mu_t}\boldsymbol{M}^{-1}\boldsymbol{A}^{(v)}$
- $\frac{\partial \boldsymbol{A}^{(v)}}{\partial \gamma^{(v)}} = -\boldsymbol{M}^{-1}\frac{\partial \boldsymbol{M}}{\partial \gamma^{(v)}}\boldsymbol{M}^{-1}(\boldsymbol{X} + \frac{\boldsymbol{\Lambda}_t^{(v)}}{\mu_t}) = -\boldsymbol{M}^{-1}\frac{\partial \boldsymbol{K}_{bc}^{(v)}}{\partial \gamma^{(v)}}\boldsymbol{A}^{(v)}$
- $\frac{\partial \boldsymbol{D}}{\partial \lambda^{(v)}} = -(\boldsymbol{D} \odot \boldsymbol{D}) \odot \frac{\partial \boldsymbol{M}^{-1}}{\partial \lambda^{(v)}} = (\boldsymbol{D} \odot \boldsymbol{D}) \odot (\boldsymbol{M}^{-1}\boldsymbol{M}^{-1})$
- $\frac{\partial \boldsymbol{D}}{\partial \gamma^{(v)}} = -(\boldsymbol{D} \odot \boldsymbol{D}) \odot \frac{\partial \boldsymbol{M}^{-1}}{\partial \gamma^{(v)}} = (\boldsymbol{D} \odot \boldsymbol{D}) \odot (\boldsymbol{M}^{-1}\frac{\partial \boldsymbol{K}_{bc}^{(v)}}{\partial \gamma^{(v)}}\boldsymbol{M}^{-1})$

where the value of $\frac{\partial \boldsymbol{K}_{bc}^{(v)}}{\partial \gamma^{(v)}}$ for each element of the kernel is given in Appendix A.1 and $\odot$ denotes the Hadamard product of two matrices. Once we have obtained the optimal parameters $\gamma^{(v)}$ and $\lambda^{(v)}$, we can compute $\boldsymbol{A}^{(v)}$ from Eq. (4.20).

Bibliography

[1] Arvind Agarwal, Samuel Gerber, and Hal Daume. Learning multiple tasks using manifold regularization. In *Advances in Neural Information Processing Systems (NIPS)*, pages 46–54, 2010. 64, 73

[2] Alan Agresti. *Analysis of ordinal categorical data*. John Wiley & Sons, 2010. 29, 90, 93, 97, 100

[3] Michal Aharon, Michael Elad, and Alfred Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing (TSP)*, 54(11):4311–4322, 2006. 22

[4] Nasir Ahmed, T Natarajan, and Kamisetty R Rao. Discrete cosine transform. *IEEE Transactions on Computers (TC)*, 100(1):90–93, 1974. 5, 51

[5] Rahaf Aljundi, Rémi Emonet, Damien Muselet, and Marc Sebban. Landmarks-based kernelized subspace alignment for unsupervised domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition (CVPR)*, pages 56–63, 2015. 112

[6] Zara Ambadar, Jeffrey F Cohn, and Lawrence Ian Reed. All smiles are not created equal: Morphology and timing of smiles perceived as amused, polite, and embarrassed/nervous. *Journal of Nonverbal Behavior (JNB)*, 33(1):17–34, 2009. 6, 7

[7] Andreas Argyriou, Stéphan Clémençon, and Ruocong Zhang. Learning the graph of relations among multiple tasks. Technical Report hal-00940321, GALEN - INRIA Saclay, 2013. 74

[8] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Convex multi-task feature learning. *Machine Learning (ML)*, 73(3):243–272, 2008. 23

[9] Akshay Asthana, Roland Goecke, Novi Quadrianto, and Tom Gedeon. Learning based automatic face annotation for arbitrary poses and expressions from frontal images only. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition (CVPR)*, pages 1635–1642, 2009. 22

[10] Akshay Asthana, Stefanos Zafeiriou, Shiyang Cheng, and Maja Pantic. Robust discriminative response map fitting with constrained local models. In *Proceedings of the IEEE*

International Conference on Computer Vision & Pattern Recognition (CVPR), pages 3444–3451, 2013. 4

[11] Tadas Baltrušaitis, Marwa Mahmoud, and Peter Robinson. Cross-dataset learning and person-specific normalisation for automatic action unit detection. In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, 2015. 26, 116, 124, 125, 128

[12] Tadas Baltrušaitis, Daniel McDuff, Ntombikayise Banda, Marwa Mahmoud, Rana El Kaliouby, Peter Robinson, and Rosalind Picard. Real-time inference of mental states from facial expressions and upper body gestures. In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, pages 909–914, 2011. 18

[13] M Bartlett and Jacob Whitehill. Automated facial expression measurement: Recent applications to basic research in human behavior, learning, and education. *Handbook of Face Perception*, 2010. 2

[14] Marian Stewart Bartlett, Gwen Littlewort, Mark Frank, Claudia Lainscsek, Ian Fasel, and Javier Movellan. Recognizing facial expression: machine learning and application to spontaneous behavior. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition (CVPR)*, volume 2, pages 568–573, 2005. 17, 18, 19, 23

[15] Marian Stewart Bartlett, Gwen Littlewort, Mark Frank, Claudia Lainscsek, Ian Fasel, and Javier Movellan. Fully automatic facial action recognition in spontaneous behavior. In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, pages 223–230, 2006. 8

[16] Marian Stewart Bartlett, Gwen C Littlewort, Mark G Frank, Claudia Lainscsek, Ian R Fasel, and Javier R Movellan. Automatic recognition of facial actions in spontaneous expressions. *Journal of Multimedia (JMM)*, 1(6):22–35, 2006. 18

[17] Peter N Belhumeur, David W Jacobs, David J Kriegman, and Neeraj Kumar. Localizing parts of faces using a consensus of exemplars. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition (CVPR)*, pages 545–552, 2011. 50

[18] Dimitri P Bertsekas. Constrained optimization and Lagrange multiplier methods. *International Journal of Applied Mathematics and Computer Science (IJAMCS)*, 1, 1982. 41, 45

[19] C.M. Bishop. *Pattern recognition and machine learning*, volume 4. Springer New York, 2006. 21, 54, 66

[20] Michael J Black and Yaser Yacoob. Recognizing facial expressions in image sequences using local parameterized models of image motion. *International Journal of Computer Vision (IJCV)*, 25(1):23–48, 1997. 17

[21] Liefeng Bo and Cristian Sminchisescu. Supervised spectral latent variable models. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 33–40, 2009. 67, 71, 74

[22] Anna Bosch, Andrew Zisserman, and Xavier Munoz. Representing shape with a spatial pyramid kernel. In *ACM Conference on Image and Video Retrieval (CIVR)*, pages 401–408, 2007. 51, 61

[23] Roberto Calandra, Jan Peters, Carl E. Rasmussen, and Marc P. Deisenroth. Manifold Gaussian processes for regression. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 2016. 90

[24] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM*, 58(3), 2011. 25

[25] Yanshuai Cao and David J Fleet. Generalized product of experts for automatic and principled fusion of Gaussian process predictions. *arXiv preprint arXiv:1410.7827*, 2014. 14, 106, 110

[26] Jixu Chen, Xiaoming Liu, Peter Tu, and Amy Aragones. Learning person-specific models for facial expression and action unit recognition. 34(15):1964–1970, 2013. 18, 27, 29, 106

[27] Wei Chu and Zoubin Ghahramani. Gaussian processes for ordinal regression. In *Journal of Machine Learning Research (JMLR)*, pages 1019–1041, 2005. 94

[28] Wen-Sheng Chu, Fernando De La Torre, and Jeffery F Cohn. Selective transfer machine for personalized facial action unit detection. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition (CVPR)*, pages 3515–3522, 2013. 18, 26, 30, 106, 107

[29] Fan RK Chung. Spectral graph theory. *American Mathematical Society (AMS)*, 1997. 36, 42, 69

[30] Ira Cohen, Nicu Sebe, Ashutosh Garg, Lawrence S Chen, and Thomas S Huang. Facial expression recognition from video sequences: temporal and static modeling. *Computer Vision and Image Understanding (CVIU)*, 91(1):160–187, 2003. 17

[31] Jeffrey F Cohn and Paul Ekman. Measuring facial action. *The new handbook of methods in nonverbal behavior research*, pages 9–64, 2005. 6

[32] Timothy F Cootes, Gareth J Edwards, Christopher J Taylor, et al. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 23(6):681–685, 2001. 4

[33] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning (ML)*, 20(3):273–297, 1995. 19

[34] Zhenwen Dai, Andreas Damianou, Javier González, and Neil Lawrence. Variational auto-encoded deep Gaussian processes. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016. 95, 97, 98, 100, 101, 102, 103

[35] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition (CVPR)*, pages 886–893, 2005. 5

[36] AC Damianou, Carl Henrik Ek, MK Titsias, and ND Lawrence. Manifold relevance determination. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 145–152, 2012. 74, 78, 81, 82, 90, 94, 95, 97, 100

[37] Andreas Damianou and Neil Lawrence. Semi-described and semi-supervised learning with Gaussian processes. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence (UAI)*, 2015. 95, 96

[38] Andreas Damianou, Michalis K Titsias, and Neil D Lawrence. Variational Gaussian process dynamical systems. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2510–2518, 2011. 133

[39] Andreas C Damianou and Neil D Lawrence. Deep Gaussian processes. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 207–215, 2013. 133

[40] Charles Darwin. *The expression of the emotions in man and animals*. John Marry, 1872. 1

[41] Hal Daumé III. Frustratingly easy domain adaptation. *Transactions of the Association for Computational Linguistics (TACL)*, page 256, 2007. 113

[42] Fernando De la Torre and Jeffrey F Cohn. Facial expression analysis. In *Visual analysis of humans*, pages 377–409. Springer, 2011. 4

[43] Marc P. Deisenroth and Jun Wei Ng. Distributed Gaussian processes. *Proceedings of the International Conference on Machine Learning (ICML)*, 2015. 14, 106, 110, 133

[44] Abhinav Dhall, Roland Goecke, Simon Lucey, and Tom Gedeon. Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark. In *Proceedings of the IEEE International Conference on Computer Vision, Workshops (ICCV-W)*, pages 2106–2112, 2011. 50, 51, 52, 61

[45] Tom Diethe, David Roi Hardoon, and John Shawe-Taylor. Constructing nonlinear discriminants from multiple data views. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD)*, pages 328–343. Springer, 2010. 49

[46] Jeff Donahue, Judy Hoffman, Erid Rodner, Kate Saenko, and Trevor Darrell. Semi-supervised domain adaptation with instance constraints. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition (CVPR)*, pages 668–675, 2013. 113

[47] Shichuan Du, Yong Tao, and Aleix M Martinez. Compound facial expressions of emotion. *Proceedings of the National Academy of Sciences*, 111(15):E1454–E1462, 2014. 6

[48] Lixin Duan, Dong Xu, and Ivor Tsang. Learning with augmented features for heterogeneous domain adaptation. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2012. 113

[49] Guillaume Benjamin Duchenne. *Mécanisme de la physionomie humaine*. F. Malteste, 1862. 1

[50] Carl Henrik Ek, Philip HS Torr, and Neil D Lawrence. Gaussian process latent variable models for human pose estimation. In *Proceedings of the International Workshop on Machine Learning for Multimodal Interaction (MLMI)*, pages 132–143. Springer, 2008. 12, 40, 95

[51] Paul Ekman. Facial expression and emotion. *American Psychologist*, 48(4):384, 1993. 2

[52] Paul Ekman. Darwin, deception, and facial expression. *Annals of the New York Academy of Sciences*, 1000(1):205–221, 2003. 2, 8

[53] Paul Ekman and Wallace V Friesen. Constants across cultures in the face and emotion. *Journal of Personality and Social Psychology*, 17(2):124, 1971. 2, 6

[54] Paul Ekman, Wallace V Friesen, and Joseph C Hager. Facial action coding system. *Salt Lake City, UT: A Human Face*, 2002. 2, 6, 7, 10, 19

[55] Paul Ekman, WV Friesen, and P Ellsworth. Emotion in the human face. 2nd edition, 1982. 2

[56] Paul Ekman and Erika L Rosenberg. *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System*. 2005. 2, 7, 8

[57] Stefanos Eleftheriadis, Ognjen Rudovic, Marc P. Deisenroth, and Maja Pantic. Gaussian process domain experts for model adaptation in facial behavior analysis. *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition (CVPR), Workshops*, 2016. 107

[58] Stefanos Eleftheriadis, Ognjen Rudovic, Marc P. Deisenroth, and Maja Pantic. Variational gaussian process auto-encoder for ordinal prediction of facial action units. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2016. 90

[59] Stefanos Eleftheriadis, Ognjen Rudovic, and Maja Pantic. Shared Gaussian process latent variable model for multi-view facial expression recognition. In *Proceedings of the International Symposium on Visual Computing (ISVC)*, pages 527–538, 2013. 41

[60] Stefanos Eleftheriadis, Ognjen Rudovic, and Maja Pantic. View-constrained latent variable model for multi-view facial expression classification. In *Proceedings of the International Symposium on Visual Computing (ISVC)*, pages 292–303, 2014. 41

[61] Stefanos Eleftheriadis, Ognjen Rudovic, and Maja Pantic. Discriminative shared Gaussian processes for multiview and view-invariant facial expression recognition. *IEEE Transactions on Image Processing (TIP)*, 24(1):189–204, 2015. 41

[62] Stefanos Eleftheriadis, Ognjen Rudovic, and Maja Pantic. Multi-conditional latent variable model for joint facial action unit detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3792–3800, 2015. 65

[63] Stefanos Eleftheriadis, Ognjen Rudovic, and Maja Pantic. Joint facial action unit detection and feature fusion: A multi-conditional learning approach. *IEEE Transactions on Image Processing (TIP)*, 25(12):5727–5742, 2016. 65

[64] Theodoros Evgeniou and Massimiliano Pontil. Regularized multi–task learning. In *ACM Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 109–117. ACM, 2004. 73

[65] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2960–2967, 2013. 112

[66] Thomas Finley and Thorsten Joachims. Training structural SVMs when exact inference is intractable. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 304–311, 2008. 73

[67] S Georgoulis, S Eleftheriadis, D Tzionas, K Vrenas, Panagiotis Petrantonakis, and Leontios J Hadjileontiadis. Epione: An innovative pain management system using facial expression analysis, biofeedback and augmented reality-based distraction. In *Proceedings of the IEEE International Conference on Intelligent Networking and Collaborative Systems (INCOS)*, pages 259–266, 2010. 2

[68] Jeffrey M. Girard, Jeffrey F. Cohn, László A. Jeni, Simon Lucey, and Fernando De la Torre. How much training data for facial action unit detection? In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, 2015. 12, 19

[69] Mehmet Gönen and Adam A Margolin. Kernelized Bayesian transfer learning. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*, 2014. 113, 114, 116, 119

[70] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition (CVPR)*, pages 2066–2073, 2012. 112

[71] Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 999–1006, 2011. 112

[72] Iris Gordon, Matthew D Pierce, Marian S Bartlett, and James W Tanaka. Training facial expression production in children on the autism spectrum. *Journal of Autism and Developmental Disorders*, 44(10):2486–2498, 2014. 7

[73] John C Gower. Generalized procrustes analysis. *Psychometrika*, 40(1):33–51, 1975. 4

[74] Arthur Gretton, Alex Smola, Jiayuan Huang, Marcel Schmittfull, Karsten Borgwardt, and Bernhard Schölkopf. Covariate shift by kernel mean matching. *Dataset shift in machine learning*, 2009. 26

[75] Keith Grochow, Steven L Martin, Aaron Hertzmann, and Zoran Popović. Style-based inverse kinematics. 23(3):522–531, 2004. 111

[76] Ralph Gross, Iain Matthews, Jeffrey Cohn, Takeo Kanade, and Simon Baker. MultiPIE. *Image and Vision Computing (IMAVIS)*, 28(5):807–813, 2010. 20, 50, 114

[77] Amogh Gudi, H Emrah Tasli, Tim M den Uyl, and Andreas Maroulis. Deep learning based facs action unit occurrence and intensity estimation. In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, volume 6, pages 1–5, 2015. 24, 25

[78] Jens Hainmueller and Chad Hazlett. Kernel regularized least squares: reducing misspecification bias with a flexible and interpretable machine learning approach. *Political Analysis*, 22(2):143–168, 2014. 44

[79] David R Hardoon, Sandor Szedmak, and John Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation (NC)*, 16(12):2639–2664, 2004. 49, 51, 56

[80] Xiaofei He, Deng Cai, Yuanlong Shao, Hujun Bao, and Jiawei Han. Laplacian regularized Gaussian mixture model for data clustering. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 23(9):1406–1418, 2011. 71

[81] Nikolas Hesse, Tobias Gehrig, Hua Gao, and Hazim Kemal Ekenel. Multi-view facial expression recognition using local appearance features. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pages 3533–3536, 2012. 21

[82] Judy Hoffman, Brian Kulis, Trevor Darrell, and Kate Saenko. Discovering latent domains for multisource domain adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 702–715, 2012. 113

[83] Judy Hoffman, Erik Rodner, Jeff Donahue, Kate Saenko, and Trevor Darrell. Efficient learning of domain-invariant image representations. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2013. 113, 116, 119

[84] Harold Hotelling. Relations between two sets of variates. *Biometrika*, 28(3):321–377, 1936. 49, 51

[85] Y. Hu, Z. Zeng, L. Yin, X. Wei, J. Tu, and TS Huang. A study of non-frontal-view facial expressions recognition. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pages 1–4, 2008. 21

[86] Yuxiao Hu, Zhihong Zeng, Lijun Yin, Xiaozhou Wei, Xi Zhou, and Thomas S Huang. Multi-view facial expression recognition. In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, 2008. 20, 21

[87] Anil Jain and Stan Z Li. Encyclopedia of biometrics: I–z. 1, 2009. 5

[88] Mahdi Jampour, Thomas Mauthner, and Horst Bischof. Multi-view facial expressions recognition using local linear regression of sparse codes. In *Proceedings of the Computer Vision Winter Workshop (CVWW)*, 2015. 22

[89] László A Jeni, Jeffrey M Girard, Jeffrey F Cohn, and Fernando De La Torre. Continuous AU intensity estimation using localized, sparse facial feature space. In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, pages 1–7, 2013. 19, 24, 89

[90] LáSzló A Jeni, András Lőrincz, Tamás Nagy, Zsolt Palotai, Judit Sebők, Zoltán Szabó, and Dániel Takács. 3D shape estimation in video sequences provides high precision evaluation of facial expressions. *Image and Vision Computing (IMAVIS)*, 30(10):785–795, 2012. 18

[91] Bihan Jiang, Michel Valstar, Brais Martinez, and Maja Pantic. A dynamic appearance descriptor approach to facial actions temporal modeling. *IEEE Transactions on Cybernetics (TCYB)*, 44(2):161–174, 2014. 18

[92] S. Kaltwang, O. Rudovic, and M. Pantic. Continuous pain intensity estimation from facial expressions. In *Proceedings of the International Symposium on Visual Computing (ISVC)*, pages 368–377, 2012. 18, 19, 24, 89

[93] Sebastian Kaltwang, Sinisa Todorovic, and Maja Pantic. Doubly sparse relevance vector machine for continuous facial behavior estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 38(9):1748–1761, 2015. 19, 24, 89

[94] Sebastian Kaltwang, Sinisa Todorovic, and Maja Pantic. Latent trees for estimating intensity of facial action units. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition (CVPR)*, pages 296–304, 2015. 24, 25, 29, 89, 97, 100, 101

[95] Meina Kan, Shiguang Shan, Haihong Zhang, Shihong Lao, and Xilin Chen. Multi-view discriminant analysis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 808–821. 2012. 49, 51, 62

[96] Melih Kandemir. Asymmetric transfer learning with deep Gaussian processes. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2015. 113, 114, 116, 119

[97] B Michael Kelm, Chris Pal, and Andrew McCallum. Combining generative and discriminative methods for pixel classification with multi-conditional learning. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, volume 2, pages 828–832, 2006. 64, 80

[98] Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2013. 91, 92, 94, 95

[99] Sander Koelstra, Maja Pantic, and Ioannis Patras. A dynamic texture-based approach to recognition of facial actions and their temporal models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 32(11):1940–1954, 2010. 18, 19, 23

[100] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1097–1105, 2012. 3

[101] Brian Kulis, Kate Saenko, and Trevor Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition (CVPR)*, pages 1785–1792, 2011. 113

[102] Abhishek Kumar and Hal Daumé III. Learning task grouping and overlap in multi-task learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1383–1390, 2012. 73

[103] Abhishek Kumar and Hal D Iii. A co-training approach for multi-view spectral clustering. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 393–400, 2011. 49

[104] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 282–289, 2001. 19

[105] N.D. Lawrence. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *Journal of Machine Learning Research (JMLR)*, 6:1783–1816, 2005. 34, 35, 75, 95

[106] Neil D. Lawrence and Joaquin Quiñonero Candela. Local distance preservation in the GP-LVM through back constraints. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 148, pages 513–520, 2006. 37, 68, 75, 95

[107] Yann LeCun, Corinna Cortes, and Christopher JC Burges. The MNIST database of handwritten digits, 1998. 96

[108] Tai Sing Lee. Image representation using 2d gabor wavelets. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 18(10):959–971, 1996. 5

[109] Yongqiang Li, S Mohammad Mavadati, Mohammad H Mahoor, and Qiang Ji. A unified probabilistic framework for measuring the intensity of spontaneous facial action units. In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, 2013. 24, 25, 89

[110] James J Lien, Takeo Kanade, Jeffrey F Cohn, and Ching-Chung Li. Automated facial expression recognition based on FACS action units. In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, pages 390–395, 1998. 17

[111] Gwen C Littlewort, Marian Stewart Bartlett, and Kang Lee. Automatic coding of facial expressions displayed during posed and genuine pain. *Image and Vision Computing (IMAVIS)*, 27(12):1797–1803, 2009. 8, 18

[112] Bo Liu and Nuno Vasconcelos. Bayesian model adaptation for crowd counts. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4175–4183, 2015. 106, 108, 109, 113, 114, 116, 119, 123, 124, 128

[113] D.G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 1150–1157, 1999. 5, 21

[114] Patrick Lucey, Jeffrey F Cohn, Takeo Kanade, Jason Saragih, Zara Ambadar, and Iain Matthews. The extended Cohn-Kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition (CVPR), Workshops*, pages 94–101, 2010. 10, 18, 19, 23, 75, 76

[115] Patrick Lucey, Jeffrey F Cohn, Iain Matthews, Simon Lucey, Sridha Sridharan, Jessica Howlett, and Kenneth M Prkachin. Automatically detecting pain in video through facial action units. *IEEE Transactions Systems, Man and Cybernetics, Part B (TSMCB)*, 41(3):664–674, 2011. 18, 19, 23

[116] Patrick Lucey, Jeffrey F Cohn, Kenneth M Prkachin, Patricia E Solomon, and Iain Matthews. Painful data: The UNBC-McMaster shoulder pain expression archive database. In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, pages 57–64, 2011. 8, 75, 76, 77

[117] Michael J Lyons, Julien Budynek, and Shigeru Akamatsu. Automatic classification of single facial images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 21(12):1357–1362, 1999. 17

[118] Mohammad H Mahoor, Steven Cadavid, Daniel S Messinger, and Jeffrey F Cohn. A framework for automated measurement of the intensity of non-posed facial action units. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition (CVPR), Workshops*, pages 74–80, 2009. 18, 19, 24, 89

[119] Mohammad H Mahoor, Mu Zhou, Kevin L Veon, Seyed Mohammad Mavadati, and Jeffrey F Cohn. Facial action unit recognition with sparse representation. In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, pages 336–342, 2011. 18, 19, 23

[120] I. Matthews and S. Baker. Active appearance models revisited. *International Journal of Computer Vision (IJCV)*, 60(2):135–164, 2004. 77

[121] Andreas Maurer. Bounds for linear multi-task learning. *Journal of Machine Learning Research (JMLR)*, 7:117–139, 2006. 81, 82

[122] Seyed Mohammad Mavadati, Mohammad H Mahoor, Kevin Bartlett, Philip Trinh, and Jeffrey F Cohn. DISFA: A spontaneous facial action intensity database. *IEEE Transactions on Affective Computing (TAC)*, 4(2):151–160, 2013. 8, 19, 24, 75, 76, 89, 96, 114

[123] Andrew McCallum, Chris Pal, Greg Druck, and Xuerui Wang. Multi-conditional learning: Generative/discriminative training for clustering and classification. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*, volume 21, page 433, 2006. 67, 74

[124] Yun-Qian Miao, Roberto Araujo, and Mohamed S Kamel. Cross-domain facial expression recognition using supervised kernel mean matching. In *Proceedings of the International Conference on Machine Learning and Applications (ICMLA)*, pages 326–332, 2012. 26, 106

[125] Zuheng Ming, Aurélie Bugeau, Jean-Luc Rouas, and Takaaki Shochi. Facial action units intensity estimation by the fusion of features with multi-kernel support vector machine. In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, volume 6, pages 1–6, 2015. 18, 19, 24, 89

[126] Mohammad Reza Mohammadi, Emad Fatemizadeh, and Mohammad H Mahoor. Intensity estimation of spontaneous facial action units based on their sparsity properties. *IEEE Transactions on Cybernetics (TCYB)*, 46(3):817–826, 2016. 24, 25, 89

[127] S. Moore and R. Bowden. Local binary patterns for multi-view facial expression recognition. *Computer Vision and Image Understanding (CVIU)*, 115(4):541–558, 2011. 20, 58, 59, 60

[128] Trung V Nguyen and Edwin V Bonilla. Collaborative multi-output Gaussian processes. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 643–652, 2014. 73

[129] Jeremie Nicolle, Kevin Bailly, and Mohamed Chetouani. Facial action unit intensity prediction via hard multi-task metric learning for kernel regression. In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, pages 1–6, 2015. 24, 25, 29, 89

[130] X Niyogi. Locality preserving projections. In *Advances in Neural Information Processing Systems (NIPS)*, volume 16, page 153, 2004. 49

[131] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 24(7):971–987, 2002. 5, 20, 51, 77, 96, 115

[132] Ville Ojansivu and Janne Heikkilä. Blur insensitive texture classification using local phase quantization. In *Proceedings of the International Conference on Image and Signal Processing (ICISP)*, pages 236–243. Springer, 2008. 51, 61

[133] Nuria Oliver, Alex Pentland, and François Bérard. LAFTER: a real-time face and lips tracker with facial expression recognition. *Pattern Recognition*, 33(8):1369–1382, 2000. 17

[134] Javier Orozco, Brais Martinez, and Maja Pantic. Empirical analysis of cascade deformable models for multi-view face detection. *Image and Vision Computing (IMAVIS)*, 42:47–61, 2015. 4

[135] Takahiro Otsuka and Jun Ohya. Recognizing multiple persons' facial expressions using HMM based on automatic extraction of significant frames from image sequences. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, volume 2, pages 546–549, 1997. 17

[136] Curtis Padgett and Garrison W Cottrell. Representing face images for emotion classification. *Advances in Neural Information Processing Systems (NIPS)*, pages 894–900, 1997. 17

[137] 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 (IJAACS)*, 1(2):168–187, 2008. 2

[138] M. Pantic and I. Patras. Dynamics of facial expression: recognition of facial actions and their temporal segments from face profile image sequences. *IEEE Transactions Systems, Man and Cybernetics, Part B (TSMCB)*, 36(2):433–449, 2006. 9, 56

[139] Maja Pantic. Machine analysis of facial behaviour: Naturalistic and dynamic behaviour. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1535):3505–3513, 2009. 2, 3, 8, 10

[140] Maja Pantic and Ioannis Patras. Detecting facial actions and their temporal segments in nearly frontal-view face image sequences. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (ICSMC)*, volume 4, pages 3358–3363, 2005. 18

[141] Maja Pantic, Alex Pentland, Anton Nijholt, and Thomas S Huang. Human computing and machine understanding of human behavior: a survey. In *Artifical Intelligence for Human Computing*, pages 47–71. 2007. 11

[142] Maja Pantic and Leon J. M. Rothkrantz. Automatic analysis of facial expressions: The state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 22(12):1424–1445, 2000. 6

[143] Maja Pantic and Leon JM Rothkrantz. Facial action recognition for facial expression analysis from static face images. *IEEE Transactions Systems, Man and Cybernetics, Part B (TSMCB)*, 34(3):1449–1461, 2004. 17

[144] Vishal M Patel, Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Visual domain adaptation: A survey of recent advances. *IEEE Signal Processing Magazine (SPM)*, 32(3):53–69, 2015. 112

[145] Rosalind W Picard and Roalind Picard. *Affective computing*, volume 252. MIT Press Cambridge, 1997. 2

[146] C.E. Rasmussen and C.K.I. Williams. *Gaussian processes for machine learning*, volume 1. MIT press Cambridge, MA, 2006. 12, 31, 33, 48, 64, 65, 68, 69, 90, 93, 97, 100, 131

[147] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1278–1286, 2014. 91, 92, 94, 95

[148] O. Rudovic, M. Pantic, and I. Patras. Coupled Gaussian processes for pose-invariant facial expression recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(6):1357–1369, 2013. 22, 59, 60

[149] O. Rudovic, I. Patras, and M. Pantic. Regression-based multi-view facial expression recognition. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pages 4121–4124, 2010. 22

[150] Ognjen Rudovic. *Machine Learning Techniques for Automated Analysis of Facial Expressions*. PhD thesis, Imperial College London, 2013. 7

[151] Ognjen Rudovic, Vladimir Pavlovic, and Maja Pantic. Context-sensitive dynamic ordinal regression for intensity estimation of facial action units. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 37(5):944–958, 2015. 11, 18, 19, 20, 24, 89

[152] Håvard Rue and Leonhard Held. *Gaussian Markov random fields: theory and applications*, volume 104. Chapman & Hall, 2005. 42

[153] Jan Rupnik and John Shawe-Taylor. Multi-view canonical correlation analysis. In *Proceedings of the International Conference on Data Mining and Data Warehouses (SiKDD)*, pages 1–4, 2010. 49

[154] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. A semi-automatic methodology for facial landmark annotation. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition (CVPR), Workshops*, 2013. 51, 115

[155] Mathieu Salzmann and Raquel Urtasun. Implicitly constrained Gaussian process regression for monocular non-rigid pose estimation. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2065–2073, 2010. 43

[156] Georgia Sandbach, Stefanos Zafeiriou, and Maja Pantic. Markov random field structures for facial action unit intensity estimation. In *Proceedings of the IEEE International Conference on Computer Vision, Workshops (ICCV-W)*, pages 738–745, 2013. 18, 24, 25, 89, 97, 100

[157] Enver Sangineto, Gloria Zen, Elisa Ricci, and Nicu Sebe. We are not all equal: Personalizing models for facial expression analysis with transductive parameter transfer. In *ACM Conference on Multimedia (MM)*, pages 357–366, 2014. 27, 106

[158] Arman Savran, Bulent Sankur, and M Taha Bilge. Regression-based intensity estimation of facial action units. *Image and Vision Computing (IMAVIS)*, 30(10):774–784, 2012. 18, 19, 24, 89

[159] Klaus R Scherer and Paul Ekman. *Handbook of methods in nonverbal behavior research*, volume 2. Cambridge University Press, 1982. 10

[160] Bernhard Schölkopf, Ralf Herbrich, and Alex J Smola. A generalized representer theorem. In *ACL Conference on Computational Learning Theory (COLT)*, pages 416–426. Springer, 2001. 45

[161] Matthias Seeger. *Bayesian Gaussian process models: PAC-Bayesian generalisation error bounds and sparse approximations*. PhD thesis, University of Edinburgh, 2003. 106

[162] T. Senechal, V. Rapp, H. Salam, R. Seguier, K. Bailly, and L. Prevost. Facial action recognition combining heterogeneous features via multikernel learning. *IEEE Transactions Systems, Man and Cybernetics, Part B (TSMCB)*, 42(4):993–1005, 2012. 18, 77

[163] Caifeng Shan, Shaogang Gong, and Peter W McOwan. Facial expression recognition based on local binary patterns: A comprehensive study. *Image and Vision Computing (IMAVIS)*, 27(6):803–816, 2009. 17

[164] Lifeng Shang and Kwok-Ping Chan. Nonparametric discriminant HMM and application to facial expression recognition. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition (CVPR)*, pages 2090–2096, 2009. 17

[165] Abhishek Sharma, Abhishek Kumar, H Daume, and David W Jacobs. Generalized multiview analysis: A discriminative latent space. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition (CVPR)*, pages 2160–2167, 2012. 49, 51, 52, 62

[166] Rishit Sheth, Yuyang Wang, and Roni Khardon. Sparse variational inference for generalized GP models. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1302–1311, 2015. 97, 100, 101, 102

[167] Aaron Shon, Keith Grochow, Aaron Hertzmann, and Rajesh Rao. Learning shared latent structure for image synthesis and robotic imitation. *Advances in Neural Information Processing Systems (NIPS)*, 18:1233–1240, 2006. 12, 37, 40, 41, 64, 95

[168] Patrick E Shrout and Joseph L Fleiss. Intraclass correlations: uses in assessing rater reliability. *APA Psychological Bulletin*, 86(2):420, 1979. 97

[169] Alex Smola and Vladimir Vapnik. Support vector regression machines. *Advances in Neural Information Processing Systems (NIPS)*, 9:155–161, 1997. 19

[170] Edward Snelson and Zoubin Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1257–1264, 2005. 133

[171] Yale Song, Daniel McDuff, Deepak Vasisht, and Ashish Kapoor. Exploiting sparsity and co-occurrence structure for action unit recognition. In *Proceedings of the IEEE*

International Conference on Automatic Face and Gesture Recognition (FG), 2015. 23, 24, 28, 78, 82, 83, 84, 85

[172] Mohammad S Sorower. A literature survey on algorithms for multi-label learning. *Oregon State University, Corvallis*, 2010. 73

[173] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. *arXiv preprint arXiv:1511.05547*, 2015. 113

[174] S Sundararajan and S Sathiya Keerthi. Predictive approaches for choosing hyperparameters in gaussian processes. *Neural Computation (NC)*, 13(5):1103–1118, 2001. 46

[175] U. Tariq, J. Yang, and T. Huang. Multi-view facial expression recognition analysis with generic sparse coding feature. In *Proceedings of the European Conference on Computer Vision (ECCV), Workshops*, pages 578–588, 2012. 21, 58, 59, 60

[176] Ying-li Tian. Evaluation of face resolution for expression analysis. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition (CVPR), Workshops*, 2004. 17

[177] Ying-Li Tian, Takeo Kanade, and Jeffrey F Cohn. Facial expression analysis. In *Handbook of face recognition*, pages 247–275. 2005. 6

[178] Michalis K Titsias and Neil D Lawrence. Bayesian Gaussian process latent variable model. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 844–851, 2010. 90, 95, 98

[179] Yan Tong, Jixu Chen, and Qiang Ji. A unified probabilistic framework for spontaneous facial action modeling and understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 32(2):258–273, 2010. 18

[180] Yan Tong, Wenhui Liao, and Qiang Ji. Facial action unit recognition by exploiting their dynamic and semantic relationships. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 29(10):1683–1699, 2007. 23, 25

[181] Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the International Conference on Machine Learning (ICML)*, page 104, 2004. 73

[182] Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining (IJDWM)*, 3(3):1–13, 2007. 73

[183] Raquel Urtasun and Trevor Darrell. Discriminative Gaussian process latent variable model for classification. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 927–934. ACM, 2007. 12, 36, 40, 42, 43, 51, 52, 62

[184] Raquel Urtasun, David J Fleet, and Pascal Fua. 3D people tracking with gaussian process dynamical models. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition (CVPR)*, pages 238–245, 2006. 111

[185] Raquel Urtasun, Ariadna Quattoni, Neil Lawrence, and Trevor Darrell. Transferring nonlinear representations using Gaussian processes with a shared latent space. Technical Report MIT-CSAIL-TR-08-020, 2008. 74, 78, 81, 82, 84, 90, 97, 100

[186] Michel F Valstar, Timur Almaev, Jeffrey M Girard, Gary McKeown, Marc Mehu, Lijun Yin, Maja Pantic, and Jeffrey F Cohn. FERA 2015 - second facial expression recognition and analysis challenge. In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, volume 6, pages 1–8, 2015. 19, 24, 25, 89, 96, 114, 116

[187] Michel F Valstar, Hatice Gunes, and Maja Pantic. How to distinguish posed from spontaneous smiles using geometric features. In *Proceedings of the International Conference on Multimodal Interaction (ICMI)*, pages 38–45, 2007. 8

[188] Michel F Valstar, Bihan Jiang, Marc Mehu, Maja Pantic, and Klaus Scherer. The first facial expression recognition and analysis challenge. In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, pages 921–926, 2011. 20

[189] Michel F Valstar and Maja Pantic. Fully automatic recognition of the temporal phases of facial actions. *IEEE Transactions Systems, Man and Cybernetics, Part B (TSMCB)*, 42(1):28–43, 2012. 10, 17, 18, 19, 23

[190] A. Vinciarelli, M. Pantic, and H. Bourlard. Social signal processing: Survey of an emerging domain. *Image and Vision Computing (IMAVIS)*, 27(12):1743–1759, 2009. 2

[191] Paul Viola and Michael J Jones. Robust real-time face detection. *International Journal of Computer Vision (IJCV)*, 57(2):137–154, 2004. 4

[192] Jack M Wang, David J Fleet, and Aaron Hertzmann. Gaussian process dynamical models for human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 30(2):283–298, 2008. 74, 133

[193] Jun Wang, Lijun Yin, Xiaozhou Wei, and Yi Sun. 3D facial expression recognition based on primitive surface feature distribution. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition (CVPR)*, volume 2, pages 1399–1406, 2006. 20

[194] Ziheng Wang, Yongqiang Li, Shangfei Wang, and Qiang Ji. Capturing global semantic relationships for facial action unit recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3304–3311, 2013. 18, 23, 28, 63, 64, 78, 82, 83, 84, 85

[195] Jianchao Yang, Kai Yu, Yihong Gong, and Thomas Huang. Linear spatial pyramid matching using sparse coding for image classification. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition (CVPR)*, pages 1794–1801, 2009. 21

[196] Ting Yao, Yingwei Pan, Chong-Wah Ngo, Houqiang Li, and Tao Mei. Semi-supervised domain adaptation with subspace learning for visual recognition. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition (CVPR)*, pages 2142–2150, 2015. 113

[197] Mohammed Yeasin, Baptiste Bullot, and Rajeev Sharma. Recognition of facial expressions and measurement of levels of interest from video. *IEEE Transactions on Multimedia (TMM)*, 8(3):500–508, 2006. 17

[198] Matthew D Zeiler. ADADELTA: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012. 95

[199] Gloria Zen, Enver Sangineto, Elisa Ricci, and Nicu Sebe. Unsupervised domain adaptation for personalized facial emotion recognition. In *Proceedings of the International Conference on Multimodal Interaction (ICMI)*, pages 128–135, 2014. 27, 106

[200] Jiabei Zeng, Wen-Sheng Chu, Fernando De la Torre, Jeffrey F Cohn, and Zhang Xiong. Confidence preserving machine for facial action unit detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3622–3630, 2015. 27, 30, 106, 116, 124, 125

[201] Z. Zeng, M. Pantic, G.I. Roisman, and T.S. Huang. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 31(1):39–58, 2009. 4, 9

[202] Jian Zhang, Zoubin Ghahramani, and Yiming Yang. Flexible latent variable models for multi-task learning. *Machine Learning (ML)*, 73(3):221–242, 2008. 64, 73

[203] Min-Ling Zhang and Zhi-Hua Zhou. Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 18(10):1338–1351, 2006. 73, 78, 82

[204] Min-Ling Zhang and Zhi-Hua Zhou. ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7):2038–2048, 2007. 73, 78, 82

[205] Xiao Zhang and Mohammad H Mahoor. Simultaneous detection of multiple facial action units via hierarchical task structure learning. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pages 1863–1868, 2014. 23, 24, 29

[206] Xiao Zhang, Mohammad H Mahoor, S Mohammad Mavadati, and Jeffrey F Cohn. An lp-norm MTMKL framework for simultaneous detection of multiple facial action units. In *Proceedings of the IEEE International Winter Conference on Applications of Computer Vision (WACV)*, pages 1104–1111, 2014. 23, 24, 29, 63, 64, 78, 82, 84, 85

[207] Xiao Zhang, Mohammad H Mahoor, and Rodney D Nielsen. On multi-task learning for facial action unit detection. In *Proceedings of the International Conference on Image and Vision Computing New Zealand (IVCNZ)*, pages 202–207, 2013. 23, 24

[208] Xing Zhang, Lijun Yin, Jeffrey F Cohn, Shaun Canavan, Michael Reale, Andy Horowitz, Peng Liu, and Jeffrey M Girard. BP4D-spontaneous: a high-resolution spontaneous 3D dynamic facial expression database. *Image and Vision Computing (IMAVIS)*, 32(10):692–706, 2014. 96, 114

[209] Kaili Zhao, Wen-Sheng Chu, Fernando De la Torre, Jeffrey F Cohn, and Honggang Zhang. Joint patch and multi-label learning for facial action unit detection. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition (CVPR)*, pages 2207–2216, 2015. 23, 24, 29, 69, 78, 82, 83

[210] W. Zheng, H. Tang, Z. Lin, and T. Huang. Emotion recognition from arbitrary view facial images. *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 490–503, 2010. 21

[211] Zhonglong Zheng, Fan Yang, Wenan Tan, Jiong Jia, and Jie Yang. Gabor feature-based face recognition using supervised locality preserving projection. *EURASIP Journal of Signal Processing (JSP)*, 87(10):2473–2483, 2007. 51

[212] Guoqiang Zhong, Wu-Jun Li, Dit-Yan Yeung, Xinwen Hou, and Cheng-Lin Liu. Gaussian process latent random field. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*, 2010. 12, 36, 40, 42, 43, 48, 51, 62

[213] Xiangxin Zhu and Deva Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition (CVPR)*, pages 2879–2886, 2012. 4

[214] Xiaojin Zhu, John Lafferty, and Zoubin Ghahramani. Semi-supervised learning: from Gaussian fields to Gaussian processes. Technical Report CMU-CS-03-175, CMU, 2003. 43

[215] Yachen Zhu, Shangfei Wang, Lihua Yue, and Qiang Ji. Multiple-facial action unit recognition by shared feature learning and semantic relation modeling. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pages 1663–1668, 2014. 18, 23, 25, 63, 64

[216] Zhiwei Zhu and Qiang Ji. Robust real-time face pose and facial expression recovery. In *Proceedings of the IEEE International Conference on Computer Vision & Pattern Recognition (CVPR)*, volume 1, pages 681–688, 2006. 9