



Reading Faces. Using Hard Multi-Task Metric Learning for Kernel Regression

Jérémie Nicolle

► To cite this version:

Jérémie Nicolle. Reading Faces. Using Hard Multi-Task Metric Learning for Kernel Regression. Machine Learning [cs.LG]. Université Pierre et Marie Curie - Paris VI, 2016. English. NNT : 2016PA066043 . tel-01365433

HAL Id: tel-01365433

<https://theses.hal.science/tel-01365433>

Submitted on 7 Jul 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Reading Faces

Using Hard Multi-Task Metric Learning for Kernel Regression



Jérémie Nicolle

Institut des Systèmes Intelligents et de Robotique

Université Pierre et Marie Curie

This dissertation is submitted for the degree of

Doctor of Computer Science

January 2016

Abstract

Collecting and labeling various and relevant data for training automatic facial information prediction systems is both hard and time-consuming. As a consequence, available data is often of limited size compared to the difficulty of the prediction tasks. This makes overfitting a particularly important issue in several face-related machine learning applications. In this PhD, we introduce a novel method for multi-dimensional label regression, namely Hard Multi-Task Metric Learning for Kernel Regression (H-MT-MLKR). Our proposed method has been designed taking a particular focus on overfitting reduction. The Metric Learning for Kernel Regression method (MLKR) that has been proposed by Kilian Q. Weinberger in 2007 aims at learning a subspace for minimizing the quadratic training error of a Nadaraya-Watson estimator. In our method, we extend MLKR for multi-dimensional label regression by adding a novel multi-task regularization that reduces the degrees of freedom of the learned model along with potential overfitting. We decided in our method to include a non-linear filter-based feature selection step based on conditional entropy for reducing the training time because of the quadratic complexity of MLKR with respect to the number of features.

We evaluate our regression method on two different applications, namely landmark localization and Action Unit intensity prediction. We also present our work on automatic emotion prediction in a continuous space which is based on the Nadaraya-Watson estimator as well. Two of our frameworks let us win international data science challenges, namely the Audio-Visual Emotion Challenge (AVEC'12) and the fully continuous Facial Expression Recognition and Analysis challenge (FERA'15).

Table of contents

| | |
|--|-----------|
| List of figures | ix |
| List of tables | xi |
| 1 Introduction | 1 |
| 1.1 What can be inferred from faces? | 2 |
| 1.1.1 Facial Action Coding System | 2 |
| 1.1.2 Towards high level information | 4 |
| 1.2 Automatic facial analysis: applications and challenges | 8 |
| 1.2.1 A few applications | 8 |
| 1.2.2 How do automatic facial analysis systems work? | 9 |
| 1.2.3 Difficulties in automatic facial expression analysis | 10 |
| 1.2.4 International challenges | 13 |
| 1.3 From landmark detection towards emotion recognition | 14 |
| 1.3.1 Landmark detection | 15 |
| 1.3.2 AU prediction | 20 |
| 1.3.3 Mental state recognition | 22 |
| 1.4 Outline and contributions | 25 |
| 2 Hard Multi-Task Metric Learning for Kernel Regression | 27 |
| 2.1 Introduction to machine learning | 28 |
| 2.1.1 A few definitions | 28 |
| 2.1.2 Different methods | 29 |
| 2.1.3 Understanding overfitting | 30 |
| 2.2 Different model types | 33 |
| 2.2.1 Non-parametric models | 34 |
| 2.2.2 Parametric models | 35 |
| 2.2.3 Semi-parametric models | 38 |

| | | |
|----------|--|-----------|
| 2.3 | Metric Learning for Kernel Regression | 40 |
| 2.3.1 | The MLKR method | 40 |
| 2.3.2 | About MLKR convexity | 41 |
| 2.3.3 | About MLKR complexity | 45 |
| 2.3.4 | About overfitting | 46 |
| 2.3.5 | About Nadaraya-Watson extrapolation capabilities | 46 |
| 2.4 | Our extensions | 49 |
| 2.4.1 | Feature selection | 50 |
| 2.4.2 | Stochastic gradient descent | 52 |
| 2.4.3 | Lasso-regularization | 53 |
| 2.4.4 | Multi-dimensional label extensions | 53 |
| 2.5 | Conclusion | 58 |
| 3 | Facial landmark detection | 61 |
| 3.1 | Introduction | 61 |
| 3.2 | Commonly used appearance features | 62 |
| 3.3 | The 300W database | 68 |
| 3.4 | Our facial landmark prediction framework | 69 |
| 3.4.1 | Feature extraction | 70 |
| 3.4.2 | Proposed regression method | 71 |
| 3.4.3 | Experimental setup | 73 |
| 3.5 | Results on the 300W dataset | 73 |
| 3.5.1 | HOG normalizations | 74 |
| 3.5.2 | Comparison to CS-MLKR | 75 |
| 3.5.3 | Embedding more training data samples | 76 |
| 3.5.4 | Comparison to global PCA and Linear Regression | 77 |
| 3.5.5 | Comparison to state-of-the-art methods | 77 |
| 3.6 | Conclusion | 79 |
| 4 | Action Unit prediction | 81 |
| 4.1 | Introduction | 81 |
| 4.2 | The BP4D dataset | 83 |
| 4.3 | AU prediction framework | 84 |
| 4.3.1 | Feature extraction | 84 |
| 4.3.2 | About learning with video data | 86 |
| 4.3.3 | Experimental setup | 88 |
| 4.4 | Results on the BP4D dataset | 89 |

| | | |
|----------|--|------------|
| 4.4.1 | Analysis of feature impact | 89 |
| 4.4.2 | Evaluations and results on the BP4D dataset | 90 |
| 4.4.3 | Evaluation of regularization impact | 90 |
| 4.4.4 | Comparison to baseline systems on the FERA'15 development set . | 91 |
| 4.4.5 | Comparison to baseline systems on the FERA'15 test set | 91 |
| 4.4.6 | Comparison to other participants on the FERA'15 test set | 92 |
| 4.5 | Conclusion | 93 |
| 5 | Conclusion and future works | 95 |
| 5.1 | Conclusion | 95 |
| 5.2 | Future works | 97 |
| 5.2.1 | Towards coupling database design and model training | 97 |
| 5.2.2 | Towards smart system adaptation | 98 |
| 5.2.3 | Towards handling big data sets | 98 |
| | References | 101 |
| | Appendix A Iterative Regularized Metric Learning | 113 |
| | Appendix B Emotion Prediction in a Continuous Space | 131 |
| | Appendix C Binary Map based Landmark Localization | 141 |

List of figures

| | | |
|------|---|----|
| 1.1 | Examples of upper face AU (figure extracted from [95]) | 3 |
| 1.2 | The different temporal phases of an AU activation | 4 |
| 1.3 | Six basic emotions. 1: disgust, 2: fear, 3: happiness, 4: surprise, 5: sadness and 6: anger (figure extracted from [13]) | 5 |
| 1.4 | Mapping of 24 emotions into a two-dimensional space valence-potency (figure extracted from [36]) | 6 |
| 1.5 | Scheme of a supervised machine learning system (figure extracted from <i>http://www.astroml.org/</i>) | 10 |
| 1.6 | Same facial expression in different lighting conditions | 11 |
| 1.7 | Scheme of automatic face-related information extraction | 15 |
| 1.8 | Landmarks localized on a face | 16 |
| 1.9 | Model alignment in ASM (figure extracted from [19]) | 17 |
| 1.10 | Model alignment in AAM (figure extracted from [17]) | 18 |
| 2.1 | Polynomial regressions of different degrees on sampled data points | 30 |
| 2.2 | Influence of the number of data points on the intensity of overfitting | 31 |
| 2.3 | Illustration of our toy example | 42 |
| 2.4 | Mean and variance of the training errors for different kernel variances | 43 |
| 2.5 | Prediction of the label with a 0.1 mean squared error | 44 |
| 2.6 | Convergence rate for 600 samples and different number of noisy features | 44 |
| 2.7 | Convergence rate for different number of data points and 10 noisy features | 45 |
| 2.8 | Training and test errors for different reduced space dimensions | 47 |
| 2.9 | Distribution of our toy example training points and corresponding labels represented as colors | 48 |
| 2.10 | Label estimation using the SVR prediction function and the Nadaraya-Watson estimator | 49 |
| 2.11 | Toy examples for understanding conditional entropy | 51 |

| | | |
|------|---|----|
| 2.12 | Comparison between MLKR and Lasso-MLKR parameter matrices on the <i>pumadyn</i> dataset | 54 |
| 3.1 | Convolution of an image by four oriented gradient filters | 63 |
| 3.2 | Examples of images and corresponding HOG histogram | 64 |
| 3.3 | Illustration of LBP class computation | 65 |
| 3.4 | Examples of images and corresponding LBP histograms | 65 |
| 3.5 | Convolution of an image by a Gabor filter bank | 66 |
| 3.6 | Illustration of different histogram normalizations | 67 |
| 3.7 | Images and landmarks extracted from the 300W challenge database | 69 |
| 3.8 | Illustration of our landmark detection framework HOG extraction process on a 2 by 2 patch | 70 |
| 3.9 | Definition of the five different point sets | 71 |
| 3.10 | Comparison between local and global HOG normalizations for landmark prediction on 300W dataset | 75 |
| 3.11 | Comparison between Hard MT-MLKR and CS-MLKR for landmark prediction on the 300W dataset | 76 |
| 3.12 | Impact of the number of shape initializations used for performing the Nadaraya-Watson regressions | 77 |
| 3.13 | Comparison between H-MT-MLKR and PCA+LR for landmark prediction on the 300W dataset | 78 |
| 3.14 | Cumulative error distribution and examples of images and located landmarks with corresponding errors | 79 |
| 4.1 | Examples of images extracted from the BP4D dataset | 83 |
| 4.2 | Two images extracted from the BP4D dataset with corresponding activated AU intensities | 84 |
| 4.3 | Image pre-processing step of our AU prediction framework | 86 |
| 4.4 | On the left: patches defined without the landmarks. On the right: centers of the patches defined using the landmarks. | 87 |
| 4.5 | Two kernel matrices corresponding to different estimators | 88 |
| 4.6 | Illustration of the four most impacting features for each learned subspace. White lines indicate point triplet angles and black arrows indicate HOG features. | 90 |

List of tables

| | | |
|-----|--|----|
| 1.1 | Review of recent international challenges on facial analysis | 14 |
| 2.1 | Time until MLKR convergence for different number of features and data points | 46 |
| 2.2 | Comparison between stochastic and standard gradient descent in MLKR on the <i>pumadyn</i> regression dataset | 52 |
| 3.1 | Comparison between H-MT-MLKR and three state-of-art methods on the 300W dataset | 78 |
| 4.1 | Comparison between standard MLKR and the two proposed multi-task extensions of MLKR in terms of Pearson's correlation coefficient (in percentage) | 91 |
| 4.2 | Comparison between baseline system (B) and proposed H-MT-MLKR (H) in terms of Pearson's correlation coefficient in percentage for different feature subsets (only geometric (G), only appearance (A) and the fusion of both (F)) | 92 |
| 4.3 | Comparison between geometric and appearance baseline systems (G and A) and proposed H-MT-MLKR (H) in terms of ICC (I) in percentage and MSE (M) | 92 |
| 4.4 | Results of the FERA'2015 fully continuous challenge in terms of ICC. . . . | 93 |

Chapter 1

Introduction

Faces convey essential information for human interaction and communication. By only looking at a face, even without taking its movement into account, we, as humans, are able to deduce a person's gender, her ethnicity, an approximation of her age, assumptions about her cultural affiliations, sometimes clues about her tiredness or mood and so on. By analyzing the dynamics of the face (head movements and facial expressions), we are able to deduce an impressive amount of precious information for social interactions. Paying attention to facial expressions help us assess feelings (e.g. slow head movements and brow lowering may evoke sadness). Facial expressions can also help conveying information which is linked to the interaction, namely back-channel signals (e.g. a polite smile or a head nod may indicate engagement and agreement). All these various signals convey fundamental information that we interpret during a conversation and to which we adapt, sometimes unconsciously.

Because of the central role of face in human interaction, automatic facial analysis has gained lots of attention in research and industry during past decades. As an example, it may be useful for designing robots able to interact with humans in a natural manner. It can also help medical research, e.g. by objectively quantifying emotional production issues in autistic patients [41] or hypersensitivity in schizophrenic ones [51].

In this PhD, we worked on designing systems to predict facial expressions and emotional states, as well as to locate facial landmarks, which is a crucial step in many face-related machine learning systems. In this chapter, we introduce automatic facial expression analysis. In section 1.1, we present a system for describing facial expressions and discuss different information that humans infer by analyzing them. In section 1.2, we discuss the applications as well as the challenges we have to face when dealing with automatic facial analysis. We review state-of-art methods in section 1.3 before concluding, highlighting our contributions and presenting the outline of this PhD dissertation in section 1.4.

1.1 What can be inferred from faces?

Communication between people involves various channels that can be separated into verbal (word messages) and non-verbal communication. The latest corresponds to all non-word messages conveying meaning, including gestures, body posture, facial expressions, eye movements as well as paralinguistic (voice intonation, rhythm). In [31], Ekman proposes a taxonomy of non-verbal behavioral cues. The first category gathers signals conveying information about cognitive states (e.g. eye gazing may convey information about attention or smiling may give insights about emotional states). The second category contains emblems, defined as culture-specific interactive signals (e.g. winking or raising thumbs up that can have different meanings in different cultures). The third one contains manipulators, that are actions used to act on objects in the environment and include self-manipulative actions (e.g. lip-biting, scratching or playing with a pen). The fourth one gathers illustrators, that are actions accompanying speech (e.g. raising eyebrows or pointing fingers) and the fifth one contains regulators, defined as conversational mediators (e.g. head nodding or exchanging a look). These categories show that facial expressions are not only useful for expressing our emotional states but at every step of an interaction process, to emphasize our message or express our understanding and opinion about the messages of other people as they speak. All those face-related non-verbal cues are linked to local facial movements, that, when analyzed, help inferring more high-level information such as emotional states. In this section, we first present the coding system used for describing those local facial movements in an objective manner. Second, we discuss a few high-level information that can be inferred with the analysis of those local facial movements, especially focusing on mental states.

1.1.1 Facial Action Coding System

In order to be able to describe facial expressions in an objective manner, several researchers have worked on classifying all potential movements of human faces. In the 19th century, Guillaume Duchenne was one of the first physician to study the impact of facial muscle activations on the appearance of the face using electrical stimuli. In 1969, the anatomist Hjorstjö described the visible appearance changes for each muscle by photographing his own face when firing his facial muscles voluntarily [43]. In 1981, inspired by those works, Ekman and Friesen proposed the Facial Action Coding System (FACS) [30]. In this system, facial expressions are divided into 44 Action Units (AU), each corresponding to one or several facial muscle activations. For instance, AU1 and AU2 correspond to Inner and Outer Brow Raiser respectively and AU10 corresponds to Upper Lip Raiser. Some examples of upper face AU are represented on figure 1.1. In order to extend the system for including the possibility

to encode head and eye movements, a revised version of FACS has been proposed in 2002 [28].
















| NEUTRAL | AU 1 | AU 2 | AU 4 | AU 5 |
|---|---|---|--|---|
|  |  |  |  |  |
| Eyes, brow, and cheek are relaxed. | Inner portion of the brows is raised. | Outer portion of the brows is raised. | Brows lowered and drawn together | Upper eyelids are raised. |
| AU 6 | AU 7 | AU 1+2 | AU 1+4 | AU 4+5 |
|  |  |  |  |  |
| Cheeks are raised. | Lower eyelids are raised. | Inner and outer portions of the brows are raised. | Medial portion of the brows is raised and pulled together. | Brows lowered and drawn together and upper eyelids are raised. |
| AU 1+2+4 | AU 1+2+5 | AU 1+6 | AU 6+7 | AU 1+2+5+6+7 |
|  |  |  |  |  |
| Brows are pulled together and upward. | Brows and upper eyelids are raised. | Inner portion of brows and cheeks are raised. | Lower eyelids and cheeks are raised. | Brows, eyelids, and cheeks are raised. |

Fig. 1.1 Examples of upper face AU (figure extracted from [95])

However, it may not be sufficient to describe facial expressions as a simple combination of activations (e.g. saying that a man raised its eyebrows, then smiled). In order to be able to describe a facial expression and its dynamics more precisely, different characteristics of facial expressions can be measured in FACS. The first one defines which AU are activated. In order to be able to describe how much each AU is activated, it is possible to encode the associated intensities. The strength of each AU can be scored using a five-level scale, from A (trace of activation) to E (maximum activation). A few AU as eyebrow raising can be activated in an asymmetrical manner (e.g. raising only one eyebrow and keeping the other down). The FACS includes the possibility of encoding the laterality of each AU, that may be scored for whether it is bilateral, unilateral, or asymmetrical. Because of the importance of the dynamics, it is possible to encode the length of the activations (e.g. describing during how much time a man has been raising its eyebrows), which is encoded in FACS using locations, corresponding to the precise moments in time when each AU begins and ends. And, finally, the possibility of encoding timing is also included in FACS for a more precise dynamic description. The

timing characteristics describes the three different phases of activation. The onset phase goes from the start to the apex, which corresponds to maximum activation. The offset phase goes from the end of apex to the disappearance of activation. A precise description of those temporal phases leads to relevant information for inferring the meaning of facial expressions. In figure 1.2, we represented those three different phases of activation of an AU.

The FACS lets to objective and precise description of human facial expressions. In our proposed facial expression prediction system, we assess the intensities of AU encoded in FACS (details are given in chapter 4). In the next paragraph, we discuss a few high-level information that can be inferred by analyzing AU activations.

Note : In this dissertation, we use 'information' to refer to any characteristic one can see or infer about a person in an interaction e.g her age, her facial movements, her engagement, her mood. We use 'facial expression' to refer to objective description of facial appearance using FACS, as opposed to 'mental states' or 'emotions', corresponding to higher-level interpretations.

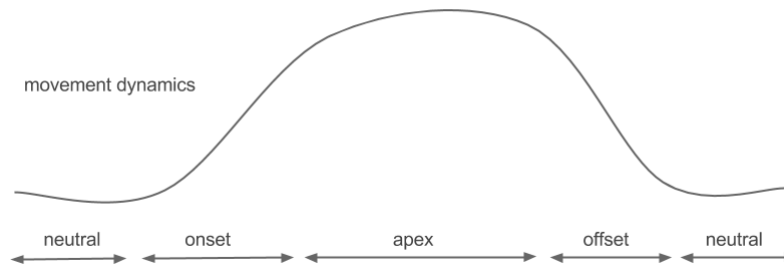


Fig. 1.2 The different temporal phases of an AU activation

1.1.2 Towards high level information

As we previously pointed out, facial expressions and their dynamics convey (consciously or unconsciously) clues about our mental states. Thus, their study can be useful for inferring higher level information such as emotional states, deception, pain or depression among others. We discuss some of those mental states in this subsection.

Emotions

Lots of works have focused on linking facial expressions to emotional states. In [63], Mehrabian studied face-to-face communication of feelings. One of his goals was to quantify

the amount of information conveyed by three distinct elements that are words, para-language and facial expressions. His conclusions led to the famous 7%-38%-55% rule stressing the primary importance of the face in human feeling expression. According to the study, more than half of the information for expressing our feelings comes from facial expressions.

In the book *The Expression of the Emotions in Man and Animals* [23] published in 1872, Charles Darwin notes the universal nature of emotion expression: 'the young and the old of widely different races, both with man and animals, express the same state of mind by the same movements.' Following that lead, in [29], Paul Ekman provided evidence in support of the hypothesis that the links between some facial expressions and six emotions were universal. The six basic emotions studied in the paper correspond to Happiness, Anger, Sadness, Disgust, Surprise and Fear. For instance, he linked Happiness to AU12 (Lip Corner Puller) and AU6 (Cheek Raiser) and Anger to AU4 (Brow Lowerer), AU5 (Upper Lid Raiser), AU7 (Lid Tightener) and AU23 (Lip Tightener). We provide an illustration of those six basic emotions on figure 1.3. In [32], Ekman supported the fact that Contempt may also be universal.

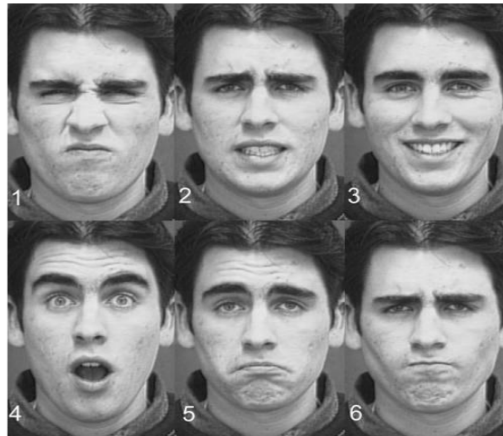


Fig. 1.3 Six basic emotions. 1: disgust, 2: fear, 3: happiness, 4: surprise, 5: sadness and 6: anger (figure extracted from [13])

However, those six emotions are not sufficient to describe the richness of human emotional states. Thus, researchers have proposed other representations, leading to finer descriptions of human emotions. For example, continuous spaces in which emotions can be mapped have been proposed. Many researchers have focused on the two-dimensional valence-arousal model [122]. Valence corresponds to pleasantness; it quantifies the positive or negative emotional charge of an event. Arousal corresponds to activity; it quantifies the excitation provoked by an event. Fontaine et al. [36] introduced a protocol for evaluating the optimal number of dimensions that should have a continuous space representing emotions. They

asked more than 500 people (Dutch-speaking, English-speaking and French-speaking) to score the likelihood of 144 emotion-related features for 24 different emotional terms. Then, they performed a Principal Component Analysis (PCA), succeeding to represent more than 75.5% of the total variance using a four-dimensional space. The first dimension (named pleasantness and corresponding to valence) accounted for 35.3% of the variance. The second dimension (named potency) accounted for 22.8%. The third dimension (named activation and corresponding to arousal), accounted for 11.4%, and the last dimension (named unpredictability) accounted for 6.0%. The potency dimension is characterized by appraisals of control, leading to feelings of power or weakness; it is linked to interpersonal dominance or submission. The fourth dimension is characterized by appraisals of novelty and opposes expectedness to unpredictability. In this study, the two-dimensional valence-arousal model is questioned as the potency dimension accounts for more variance than the arousal dimension. In figure 1.4, we illustrate the mapping of the 24 chosen emotions into a two-dimensional space valence-potency. We can notice that the valence axis opposes for instance joy and love (positive feelings) to jealousy or anger (negative feelings). The potency axis separate emotions like anxiety or shame, that are submission feelings, to hate or contempt, that are dominant feelings. During the first months of this PhD, we designed a framework for predicting emotions in this four-dimensional continuous space. This work has been published in the International Conference on Multimodal Interaction (ICMI 2012 [75]). The corresponding paper can be found in appendix B.

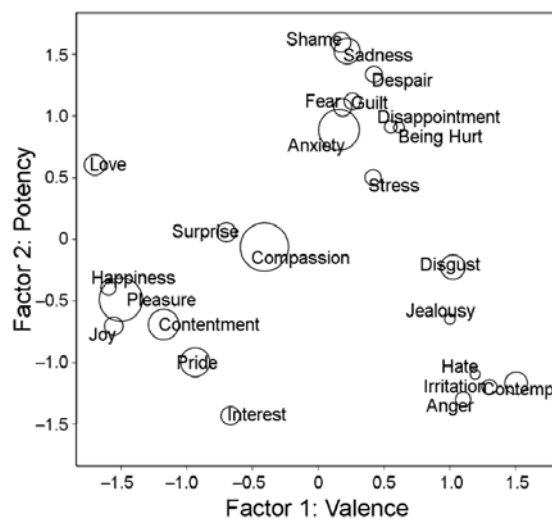


Fig. 1.4 Mapping of 24 emotions into a two-dimensional space valence-potency (figure extracted from [36])

Other information related to facial expressions

Aside from emotions, many other information regarding human behavior involve specific facial expressions. In past years, researchers have begun to study the relationship between facial expressions and deception, pain or depression among others.

In [27], Ekman presented a study on deception using facial expressions. Among his conclusion, he found that spontaneous emotional expressions were more symmetrical than those made deliberately. Moreover, differences in the dynamics of expressions exist. In deliberate expressions, the onset is often abrupt, the apex is longer, and the offset appears irregular rather than smooth. He also supported the fact that some micro-expressions, which are AU activations within a very brief time, may be hard to inhibit and thus may reveal emotions that subjects try to hide. His research points out the importance of precise analysis of AU intensities over time for being able to distinguish natural from acted facial expressions, or to infer high-level information such as deception.

Recently, many efforts have been made for trying to identify valid facial indicators of pain. In 2008, Prkachin and Solomon [78] proposed a pain intensity evaluation rule based on AU intensities, which states that:

$$Pain = AU4 + \max(AU6, AU7) + \max(AU9, AU10) + AU43$$

We recall that AU4 corresponds to Brow Lowerer, AU6 to Cheek Raiser, AU7 to Lid Tightener, AU9 to Nose Wrinkler, AU10 to Upper Lip Raiser and AU43 to Eyes Closed. Indeed, intense pain is frequently associated with those specific facial muscle contractions. This rule makes the hypothesis that physical pain can be entirely assessed using AU.

In addition to deception and pain, researchers have also tried in recent studies to link some facial behaviors to depression states. Major Depression Disorder (MDD) is the most prevalent mood disorder and concerns between 10 and 20% of women and between 5 and 12% of men during lifetime. Neuro-physiological changes in people with MDD can modify speech and facial expression production. Thus, being able to precisely analyze the dynamics of head movements, eye gaze and facial expressions could be of help for trying to detect the emergence of the first signs of these mood disorders for treating them earlier.

Those examples of information that may be inferred using facial expressions show the wide range of applications that may have automatic facial expression assessment. In the next section, we review some of those applications and discuss the main difficulties we encounter in automatic facial expression analysis.

1.2 Automatic facial analysis: applications and challenges

The wide range of information that can be conveyed by faces and the increase of computational power of machines made research on automatic facial analysis emerged since past decades. In this section, we first present a few applications involving automatic facial expression recognition. Then, we briefly describe how automatic systems work before discussing the main difficulties that we have to deal with when trying to automatically predict facial expressions. Afterwards, we present some international challenges that have been organized in order to accelerate research in this domain and make fair comparisons between different methods possible.

1.2.1 A few applications

There are many applications that could benefit from automatic facial expression recognition, in domains such as robotics, security, medicine or marketing among others.

Designing systems able to discuss with humans naturally is a challenging task. The role of the face for conveying emotions or sending back-channel signals during interaction has made automatic facial analysis of primary importance for those systems. Applications are numerous, from conversational agents to social robotics. Recently, the European project 'CompanionAble'¹ led to Hector, a robot designed for assisting elderly people living alone. Among other abilities, it includes a personalized dialog system displaying emotional intelligence to avoid feelings of loneliness and offer cognitive stimulation through games.

Automatic facial expression detection is also important in security-related applications. As an example, driver's drowsiness is one of the most frequent causes of traffic accident. Several fatigue detection systems are already used in cars recording and analyzing driving-related features. However, the earlier we can detect fatigue, the more accidents could be prevented, which explains current research on other ways of detecting drowsiness. By using Electrooculography and Electromyography for recording eyelid and head movements of people in a driving simulator, Hu and Zheng [44] have successfully linked facial dynamic information to drowsiness scores, showing that a precise tracking of head and eyelids using video cameras embedded in cars could increase passenger security.

In the medical domain, automatic facial expression recognition could also be useful, for instance in the context of patients' pain assessment. Pain is usually measured by patient self-report. Experiences related to pain are subjective and can be affected by physical, psychological, and social factors [14]. Thus, it is a complicated issue to deal with for the

¹www.companionable.net/

medical profession. The EPSRC Emo & Pain project² aims at assisting patients with lower back pain using automatic recognition systems able to interpret and act to human affective states related to pain.

The industry has also begun to take interest in automatic facial expression recognition. Several companies (as Affectiva³ or RealEyes⁴) have designed systems for emotional recognition from facial expressions, which are used to analyze the impact of advertisements. By showing advertisement videos to a panel of users and by recording their webcam, Affectiva, for instance, proposes to analyze the emotional responses of the users along time (detecting smiles, laughs, eyebrow frowning), letting companies better understand and quantify the relevance of each sequence of their advertisements.

All those applications related to widely different domains explain the emergence of automatic facial expression research. In the next subsection, we explain how automatic facial analysis systems work before discussing some of the main difficulties that we have to deal with when designing those systems.

1.2.2 How do automatic facial analysis systems work?

In order to design face-related machine learning systems (e.g. face recognition from images or automatic emotion recognition from video sequences), supervised learning algorithms have been used. Those algorithms make use of labeled databases (e.g. a set of face images labeled with corresponding people's names, or a set of video sequences to which experts have attached emotional labels). Using those databases, models are learned in order to be able to predict those information from new data samples. Let us explain the main steps of an automatic supervised machine learning system. First, the feature extraction step aims at collecting and summarizing information describing data samples. Then, during the training step, the parameters of the model are optimized for predicting as well as possible a part of the database, namely the training set. Finally, the testing step aims at evaluating the learned model; it is performed on another labeled part of the database, namely the test set. We illustrate supervised prediction systems with a scheme in figure 1.5. An introduction to machine learning and discussions about the different types of potential models can be found in chapter 2.

²www.emo-pain.ac.uk/

³<http://www.affectiva.com/>

⁴<https://www.realeyesit.com/>

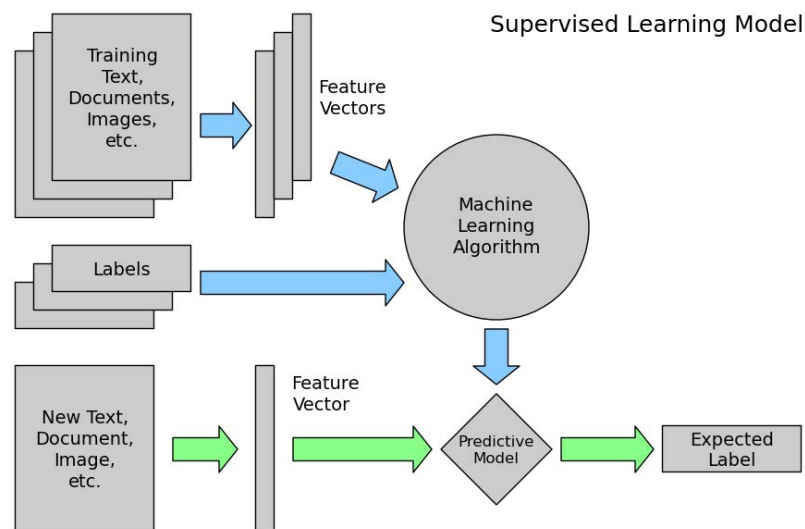


Fig. 1.5 Scheme of a supervised machine learning system (figure extracted from <http://www.astroml.org/>)

1.2.3 Difficulties in automatic facial expression analysis

In this subsection, we present the main difficulties for designing automatic prediction systems based on facial analysis. First, we discuss difficulties related to the training data. Then, we discuss difficulties related to the feature choice. Afterwards, we discuss difficulties related to the learning methods before finally discussing difficulties resulting from real-life use of those systems.

About the training data

Difficulties related to the training data can come from the data complexity, the labeling quality or the database size.

First, some difficulties come from the data complexity which is induced by the recording conditions (e.g. lighting or resolution), the subjects movements (e.g. head pose) as well as the subjects specificities (e.g. age or facial morphology). Regarding the recording conditions, several elements in the recording environment have impact on the images, increasing data complexity and making the learning process more difficult. As an example, recordings of the same facial expression with various lighting conditions can induce important modifications of the image pixels intensity values. Figure 1.6 illustrates those modifications induced by lighting changes. The camera resolution is also of important matter. Low resolutions can lead to a difficult characterization of the facial appearance and high resolutions can increase the time complexity of the feature extraction step. Regarding the subjects movements,



Fig. 1.6 Same facial expression in different lighting conditions

the orientation of the head relative to the camera can increase data complexity. In natural behaviors, important head movements can occur, leading to images with important pose variations. The impact of head pose on the appearance of faces is highly important and hard to deal with. Regarding subjects' specificities, even in an adequate recording environment and controlled movements, the appearance of different people faces can be extremely various. First, people have specific facial morphologies; many differences exist among people of different age, sex, weight or skin color. Second, beards, mustaches, hairs, glasses, hats, tattoos, piercings, make-up and so-on add complexity to the data and thus make the learning process harder.

Second, another difficulty related to the training data is to know how much we can rely on the data labels. How close are they to the truth? In some cases, labels can be entirely trusted and correspond without doubt to reality (e.g. the subjects' age when having their identity information). However, in other cases, there can be a need to label more subjective information (e.g. the emotional state of a subject along time). In this case, it is often necessary to have several experts labeling the data. This way, the label can better represent the characteristic that is aimed to be predicted as it is averagely perceived by people. For some complex prediction tasks, there exist large differences between the different experts annotations. This uncertainty in the labels can be a difficulty for training relevant models.

Finally, the database size is also a limiting factor. Depending on the complexity of the prediction task, system performance is highly linked to the training data size. The main difficulty for training a precise system is to carefully choose features and adapt the learning methods to the relationship between the training data size and the prediction task complexity. We define this prediction task complexity as being both linked to the complexity of the task itself (its ease of assessment by humans) and to the data complexity (coming from the recording conditions, the subjects movements as well as the subjects specificities).

About feature choice

In order to learn precise systems given previously presented difficulties, feature choice is of major importance. Features need to be relevant for the considered task.

As an example, they can be designed to be invariant to some data variations. In order to reduce the impact of luminosity changes, several works have focused on designing characteristics that are the most insensitive to lighting changes as possible (e.g. Local Binary Patterns LBP [76], that are presented in chapter 3).

Let us give two examples of questions related to feature choice in facial analysis applications. A common question regards the kinds of information to extract. If we have video data, will it be useful to extract audio characterizations? Do we need to extract geometric features, appearance features or both? Geometric features characterize relationships between key points on faces (e.g. the centers of the eyes or the mouth corners). Appearance features describe the texture of the skin. Another common question regards dynamic information inclusion. In order to predict information that vary smoothly along time (as emotional states), is it better to use static or dynamic features? Indeed, it is possible to use static features combined with a learning system that is able to model the dynamics (e.g. a Hidden Markov Model), or to encode dynamic information directly within features (e.g. by using a Fourier spectrum decomposition as we proposed to do in [75]) and use a static learning method (e.g. Support Vector Machines).

About the learning models

It is important to take into account features and data characteristics when designing models. In machine learning, different models may be more or less complex, i.e. they can model more or less complex functions between features and labels. This complexity is linked to the degrees of freedom of the model, i.e. the number of parameters that have to be estimated. More details are given in chapter 2. It is important to choose methods that will be able to generalize well when trained on the training data (i.e. the system performance must be high on data that has not been used for training). This difficulty of choosing the adequate degrees of freedom of a model (and thus its modeling ability) given the training dataset is related to what is called overfitting. In order to reduce overfitting, it may be relevant to reduce the number of parameters that have to be optimized while keeping a sufficient complexity. That is the goal of our proposed Hard Multi-Task MLKR method that we introduce in chapter 2. It frequently occurs that databases are labeled with several labels. It can then be relevant to try to make use of those multi-dimensional labels. Multi-task regularization aims at making use of all the information available in the databases for training the prediction models. More

details about this regularization are given in chapter 2. The choice of the method relative to the extracted features, the database and the prediction task is one of the main difficulties for designing precise facial analysis systems. Indeed, data is often hard to collect and label which results in data that can be of limited size compared to the complexity of the prediction task.

About real use of systems

As we previously discussed, numerous applications can benefit from face-based learning systems. However, it is necessary when designing systems to take into account the constraints linked to real use of the systems. In systems as fatigue detection in cars or social robotics, real-time is a necessary constraint. This constraint restricts the allowed complexity of the algorithms for the feature extraction and for the prediction function computation. Memory required by the algorithms can also be an important issue, e.g. in embedded systems for robotics where available RAM may be limited. In order to try to solve those previously presented issues, numerous methods have been proposed in the past decades for automatically extracting information on faces like facial landmark locations, activations of facial muscles or emotional states. We present some of the main methods in section 1.3. In the next subsection, we present some recent international facial analysis challenges.

1.2.4 International challenges

In order to stimulate research on automatic facial analysis and be able to compare methods, several international challenges have recently been organized. Papers corresponding to winning systems are reported on table 1.1.

The 300 faces in-the-wild challenges (300W) aimed at comparing methods for landmark localization. There were organized in conjunction with the International Conference on Computer Vision (ICCV 2013 and ICCV 2015). The participants had to predict 68 landmarks on images coming from the 300W dataset (a mix of several existing datasets such as the Helen dataset or the LFPW dataset). More details on the 300W dataset along with the evaluation of our landmark detection method can be found in chapter 3. In the EmotiW challenges, participants had to predict six discrete emotions plus neutral on the AFEW dataset, where images come from movies showing close-to-real-world conditions. Five audio-visual emotion challenges (AVEC) were organized each year since 2011 focusing on mental state prediction. In 2011, the participants had to predict emotions on the SEMAINE database in a quantized four-dimensional space; they had to predict along time whether valence, arousal, expectancy and power were above or below average. In 2012, the participants had to

predict emotions on the same dataset in a continuous four-dimensional space. In 2013 and 2014, participants had to predict affects as well as a depression indicator on the AViD corpus. In 2015, participants had to predict emotions on the RECOLA database in a continuous two-dimensional space. Our proposed method [75] won the first place of the AVEC'12 fully continuous challenge. We present this work in appendix B. Two facial expression recognition and analysis challenges (FERA) were held in 2011 and 2015 in conjunction with the IEEE International conference on Face and Gesture Recognition. In 2015, the participants had to predict the activations and intensities of AU on the BP4D and the SEMAINE databases. The FERA'15 fully continuous challenge has been won by our proposed Hard Multi-Task Metric Learning for Kernel Regression method [73]. This work is presented in chapter 4.

| Name | Year | Database | Winner(s) |
|-------------|------|----------------|--|
| 300W [83] | 2013 | 300W mix | [118] (academia), [131] (industry) |
| 300W [91] | 2015 | 300W mix | [119], [116] |
| Emotiw [25] | 2013 | AFEW | [47] |
| Emotiw [24] | 2014 | AFEW | [56] |
| AVEC [88] | 2011 | SEMAINE | [64] (audio), [79] (video) |
| AVEC [89] | 2012 | SEMAINE | [75] (fully continuous), [86] (word level) |
| AVEC [102] | 2013 | AViD-Corpus | [66] (affects), [113] (depression) |
| AVEC [101] | 2014 | AViD-Corpus | [46] (affects), [112] (depression) |
| AVEC [81] | 2015 | RECOLA | [42] |
| FERA [103] | 2011 | GEMEP-FERA | [90] (AUs), [121] (emotions) |
| FERA [99] | 2015 | BP4D / SEMAINE | [73] (fully continuous), [123] (occurrence), [2] (pre-segmented) |

Table 1.1 Review of recent international challenges on facial analysis

In next section, we present state-of-the-art methods in three subdomains of facial analysis, namely landmark detection, automatic AU prediction and emotion recognition.

1.3 From landmark detection towards emotion recognition

The first step of many face-related computer vision systems is to localize facial areas in images. The most used real-time face detector has been proposed by Paul Viola and Michael J. Jones in 2001 [106]. In their method, Haar-like features are extracted at different scales using integral images to accelerate the process. Then, the Adaboost method [37] is used for training classifiers that are combined in a cascaded way. The detector has been widely used because of its low computation time and its high performance. However, since 2001,

numerous other face detection methods have been proposed (e.g. a robust multi-pose face detection [115]). For more details about face detection, one can refer to the very recent review of Zafeiriou et al. [124].

After this step, the face is coarsely localized in the image. Then, a need for a finer description of the face leads to detecting facial landmarks (that are key points in faces as the centers and contours of the eyes, the eyebrows, the nose and the mouth). Those landmarks can be used afterwards for extracting geometric and appearance information in order to predict the activations of facial muscles along time. Afterwards, dynamic information can be extracted from signals representing head and facial movements for predicting higher level information such as emotional states. On figure 1.7, we represented a scheme of automatic face-related information extraction. In this section, we present some of the main methods proposed in the literature for landmark detection, AU prediction and mental state recognition.

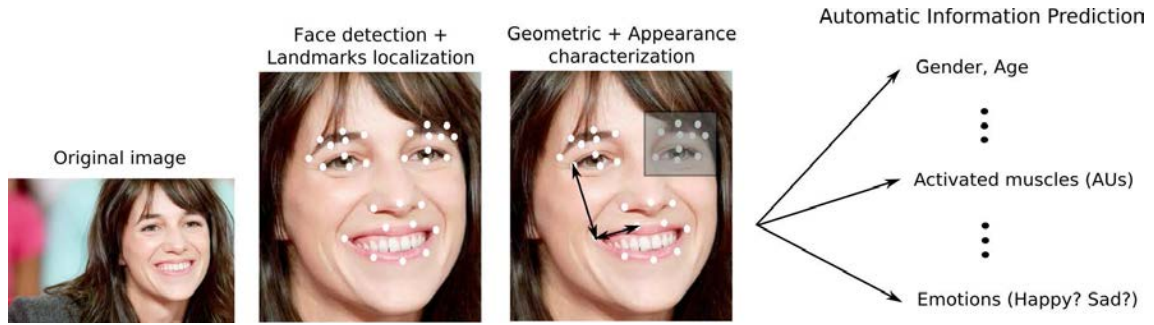


Fig. 1.7 Scheme of automatic face-related information extraction

1.3.1 Landmark detection

Landmark detection is a fundamental step in many facial-related computer vision systems. The problem is to precisely locate a set of facial key points in images. On figure 1.8, we represented 49 landmarks on a face. In most methods, facial landmark detection is performed through an iterative alignment process. A mean model is first initialized on the image and is then fitted step after step using appearance characterization. Different types of methods have been proposed in the literature. In this paragraph, we introduce four of the main classes of landmark detection methods, namely Active Shape Models, Active Appearance Models, Constrained Local Models and Cascaded-Regression methods.

Active Shape Models

In their seminal work of 1992, Cootes and Taylor [19] proposed Active Shape Models (ASM) for solving the task of landmark localization. Because different points of an object may

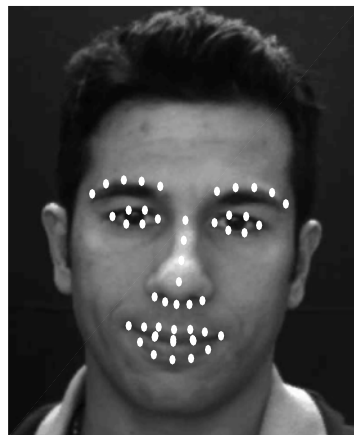


Fig. 1.8 Landmarks localized on a face

have a very alike local appearance (as the eye corner and the mouth corner at a small scale), and because of potential occlusions, detecting each landmark separately would not lead to taking advantage of the knowledge of the positions of the points relative to each other. Thus, it is important to include constraints relative to the global shape of the object in order to increase robustness. For doing so, in [19], the authors proposed the Point Distribution Model (PDM) for representing shapes. In this model, a shape is represented as a mean shape plus a distortion defined as a linear combination of different modes of variations as:

$$\mathbf{x} = \bar{\mathbf{x}} + \mathbf{P}\mathbf{b}$$

The vector \mathbf{x} corresponds to the coordinates of the points, $\bar{\mathbf{x}}$ corresponds the mean positions, \mathbf{P} is the matrix containing the modes of variations and \mathbf{b} corresponds to the associated weights. The model parameters $\bar{\mathbf{x}}$ and \mathbf{P} are learned using a Principal Component Analysis (PCA) on a set of labeled samples. During test, for each image and each landmark, a gradient vector is extracted along a line orthogonal to the shape boundary at the landmark location (as represented on figure 1.9) and the movement is predicted towards the location of the maximum of the gradient vector proportionally to its intensity. In [20], profile models are used in order to describe the appearance of each landmark. Mean profile vectors and covariance matrices are calculated for each landmark. During test, each landmark tends to move along the orthogonal line towards the pixel whose profile has the smallest Mahalanobis distance to the corresponding landmark model. Afterwards, the displacements of the parameters \mathbf{b} are calculated for minimizing the error between the set of landmarks generated using the manifold learned by PCA and the set of potential candidates.

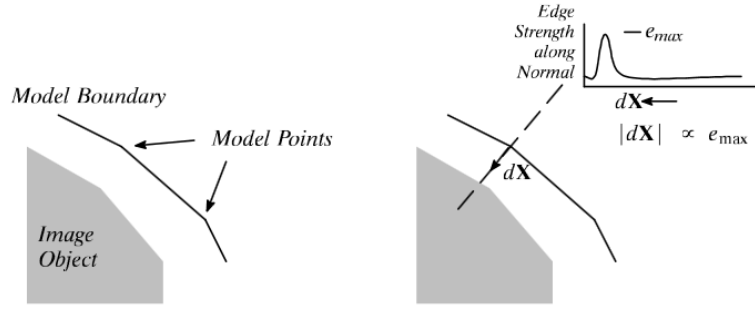


Figure 2 : Part of a model boundary approximating to the edge of an image object.

Figure 3 : Suggested movement of point is along normal to boundary, proportional to maximum edge strength on normal.

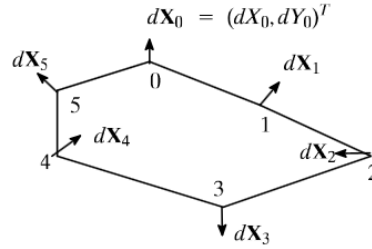


Figure 4 : Adjustments to a set of points

Fig. 1.9 Model alignment in ASM (figure extracted from [19])

Because the characterization of the appearance along a line may not be sufficient, Milborrow and Nicolls [67] extend ASM by proposing (among other modifications) two-dimensional profiles for each point, for being able to capture more appearance variations than the initial model.

Active Appearance Models

In 1998, Cootes et al. [16] proposed Active Appearance Models (AAM). In this work, the shape model and the profile model of ASM are merged into a single appearance model. For each training image, a warping is performed in order to obtain a shape-free patch where landmarks are projected into the mean shape. A global model characterizing both shape and appearance is then learned using PCA. For a test image, the alignment is performed in order to minimize the quadratic error between the synthesized model and the sample. The optimization is performed in the parameter space making a prior assumption of a fixed Jacobian matrix. An illustration of the process is represented on figure 1.10.

In [15], a comparison is performed between ASM and AAM on 400 face images labeled with more than a hundred landmarks. The authors conclude that ASM is faster and leads to results that are comparable to those obtained with AAM.

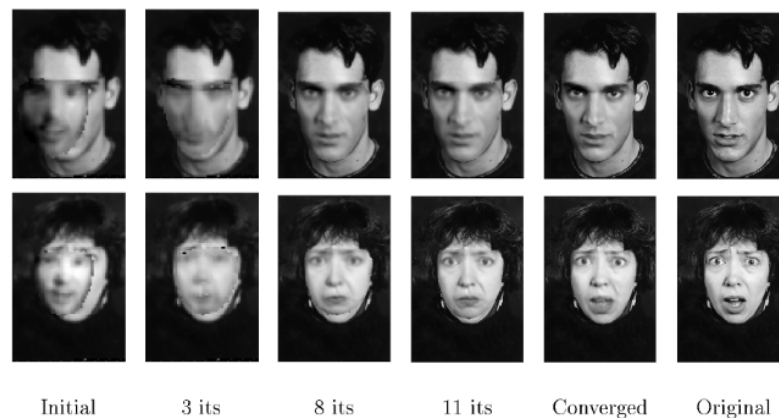


Fig. 1.10 Model alignment in AAM (figure extracted from [17])

Numerous modifications of AAM have been proposed in the literature since 1998. Matthews and Baker [60] proposed an efficient fitting for AAM based on the *inverse compositional* algorithm. Batur and Hayes [3] proposed an algorithm that linearly adapts the gradient matrix according to the texture of the target image to obtain a better estimate of the gradient. Tzimiropoulos et al. [98] proposed Active Orientation Models where (among other modifications) gradient maps are modeled instead of gray-level patches. All those modifications have made AAM very efficient, especially person-specific AAM that still continue to be used with success because of their ability to efficiently adapt to a specific subject by capturing texture very precisely.

Constrained Local Models

At each iteration of an ASM, each landmark is attracted to one candidate location for next step and the parameters of the model are computed for minimizing the error between the shape generated with the PDM and the set of candidates. In Constrained Local Models (CLM [84]), probability maps are calculated for each landmark and the optimization of the shape parameters is performed in order to find the best overall match combining all point probabilities.

Several methods have been proposed for estimating the probability maps of each landmark. Discriminative methods have been used for training local experts that estimate the probability maps of each landmark. In discriminative methods, the experts are learned by trying to classify positive patches (centered on the landmarks) and negative patches (with centers displaced from their accurate locations). For instance, SVM have been used in [84] and [109]. Another way of learning local probability maps is to use regressors. Cootes et al. [18] propose to use a random forest for learning the displacements from the centers of local

patches towards the true location of the landmark and then to use a voting method to compute the probability maps.

Several other CLM have been proposed in the literature in which the shape constraints are included without using the PDM model. In [100], Support Vector Regression (SVR) is used for generating local probability maps and the shape constraints are included using Markov Random Fields. In [4], the shape constraints are included using a RANSAC-like algorithm. CLM are very competitive landmark detection algorithms. They have been extensively used until the emergence of Cascaded-Regression methods, that we present in next paragraph.

Cascaded-Regression methods

Over the past few years, Cascaded-Regression methods have shown impressive results in automatic facial landmark detection in the wild [6] [117] [94] as compared to previously presented methods. In those methods, no explicit shape models are estimated. However, in order to implicitly include the shape constraints, the landmark displacements are estimated in a joint manner.

Let I be the image and \mathbf{x} be a vector containing the coordinates of the landmarks to be predicted. Let $\hat{\mathbf{x}}_t$ be the estimations of the landmarks at step t . At each step, a predictor

$$P_t : (I, \hat{\mathbf{x}}_{t-1}) \rightarrow \delta$$

is learned. δ corresponds to the current landmark displacements towards ground truth defined as:

$$\delta = \mathbf{x} - \hat{\mathbf{x}}_{t-1}$$

The choice of the features and the regression method are very important questions. In Supervised Descent Method (SDM [117]), SIFT descriptors are extracted on patches centered on the landmarks. A PCA is applied for dimensionality reduction and linear regressors are learned at each step. Yan et al. [118] compared different features for landmark detection (HOG, SIFT, Gabor and LBP) showing that HOG were the best ones on the LFPW database [4].

Ren et al. [80] proposed a cascaded regression method with local discriminative features. Their method achieves a significant error reduction of 22% with respect to SDM on the challenging IBUG dataset⁵.

The framework we present in chapter 3 also lets us use local features combined with a non-linear regression method.

⁵<http://ibug.doc.ic.ac.uk/resources/300-W/>

1.3.2 AU prediction

Numerous AU prediction methods have been proposed during the past decade along with the growing interest in this domain. Along all the data processing chain, from the acquisition sensors to the prediction method, many questions have been highlighted by past works. First, the availability of affordable three-dimensional sensors has attracted many researchers to focus on the utility and contribution of depth-related data for facial muscle activation prediction. This has made the data type a relevant question. Second, the choice of the areas used for feature extraction is also an important matter. Third, including prior human knowledge when designing task-specific high-level features can increase performance but may lead to less generic systems. In a similar way, including prior knowledge within the models (e.g. about AU co-occurrences in natural facial expressions) has also raised questions. Finally, the choice of the learning methods used for data modeling has also been an active topic in past works.

About three-dimensional information

The relevance of using 3D data for facial expression recognition has been investigated by several researchers. Sun et al. [93] used three-dimensional motion vectors and Hidden Markov Models (HMM) for predicting AU and discrete emotions on the Dynamic 3D Facial Expression Database. Savran et al. [87] extracted local 3D shape features (mean and Gaussian curvatures, shape index and curvedness among others). They used SVM for predicting AU on the Bosphorus database. However, 3D sensors are not yet widely democratized and many applications have a need of 2D data solutions, which explains the numerous recent 2D approaches for AU prediction [53] [61] [107]. Most of those 2D approaches can be easily extended to 3D data, extracting complementary features on depth maps the same way than on gray-level or color images.

About appearance patch location

A few methods [120] [11] extract features on more or less global regions defined using the area obtained with the face detector (commonly using Viola and Jones algorithm [106]). Yang et al. [120] extracted dynamic Haar-like features after a rescaling of the detected face image and then encoded it with binary patterns before classifying using Adaboost [37]. Chuang and Shih [11] divided the face region in upper and lower parts before using SVM on Independent Component Analysis (ICA) projections. Other methods use eye localization for defining feature extraction areas [97] [90]. In Jeni et al. [45] and Chu et al. [10], all fiducial points describing the facial shape are used to define local patches for feature extraction in order

to predict AU. By definition, AU correspond to local movements of the facial appearance. This is why extracting features on local areas defined from fiducial points lead to relevant information for the task. However, using more global areas defined only using the face region or the centers of the eyes (which are the most accurately located points in landmark detection methods) can avoid the spread of possible errors of facial point tracking. The recent improvement of facial point localization systems can explain the fact that local areas defined from landmarks are more and more used in AU prediction systems [45] [10] [107].

About prior assumptions for feature design

AU prediction methods also differ regarding the amount of human prior knowledge included when designing the features. Some methods use data-driven features, which often makes the framework more generic, as Chuang and Shih [11] that use Independent Component Analysis (ICA) or Jeni et al. [45] that use Non-negative Matrix Factorization (NMF). Even if it introduces a loss of genericity, other methods use handcrafted features, that may lead to relevant invariance and characterizations. Rudovic et al. [82] use Local Binary Patterns (LBPs) that are invariant to illumination changes. Gabor wavelets are commonly used [97] [90] [87] and have shown very promising results for AU prediction as pointed by Littlewort et al. [54]. However, dense computation of those features for different scales and orientations quickly becomes time-consuming and may be unsuited for real-time algorithms. This explains the choice of Histograms of Oriented Gradients (HOG) made by McDuff et al. [62] which encode relevant information for expression-relative wrinkle characterization while being less time-consuming to compute.

About prior assumptions for model design

Human prior knowledge can also be included in the data modeling. Several researchers focused on learning dynamic relationships and co-occurrences between AU in order to increase the system performance for natural behavior analysis, as Tong et al. [97] and Li et al. [53] that use Dynamic Bayesian Networks (DBN) or Zhao et al. [130] that introduced JPML (Joint Patch and Multi-Label learning) for solving this issue. These approaches are able to take into account correlations between AU in natural facial expressions. For instance, eyebrow raising (AU1+AU2) and upper lid raising (AU5) are often activated simultaneously. However, AU correspond to facial muscles that can be activated in an independent manner, making prior knowledge about co-occurrences between AU questionable for some applications. For instance, in the context of facial reeducation for patients that had a cerebrovascular accident (CVA), different muscles may need to be re-trained, and thus separately activated by the

patient (and separately recognized by the system). A prior knowledge inclusion in this case could bias the prediction system.

About model choice

Finally, there is the question of the machine learning algorithms used for building prediction frameworks. In many databases (Cohn-Kanade [48], GEMEP-FERA [103]) AU are labeled as activated or not, stating the problem as a classification one. Thus, Support Vector Machines (SVM) have been widely used in the facial expression domain [90] [104] [62]. A need for a greater precision in AU recognition systems has motivated the availability of new databases approaching the task as a regression one (Bosphorus [85], CK+ [58], UNBC-McMaster [59], DISFA [61]). However, the choice of the optimal machine learning algorithms for AU intensity prediction stays an open question. For solving this task, Savran et al. [87] adapt a classification learning machine for regression. For doing that, they use SVM and afterwards perform a logistic regression on the non-thresholded SVM outputs. On the contrary, Jeni et al. [45] directly use a regression learning method, Support Vector Regression (SVR), for AU intensity prediction, obtaining excellent results on Enhanced Cohn-Kanade (CK+) database. Recently, Girard [38] focused on smile intensity prediction, showing that SVR outperformed multi-class SVM.

Our choices

Considering all these remarks, in our proposed AU intensity prediction framework, we decided to use 2D data for it to be more generic. We extract patches both centered using the landmarks and only using the facial detected area. Our model is based on the regression method that we present in chapter 2, that aims at reducing potential overfitting issues. In next subsection, we present state-of-the-art methods on automatic mental state recognition.

1.3.3 Mental state recognition

Several works have focused on automatic prediction of high-level information regarding human behavior. They aim at going further than a description of facial expressions, trying to infer a meaning from those, in order to respond to more easily interpretable questions as: 'Is this man happy?' or 'Is he tired?'. When designing those systems, several questions are relevant and have been studied in the literature. First, is an image enough to give insights about high-level information as a person's mood? If not, how to include dynamic information? Facial information may not be enough. Then, how to combine those information in a multi-modal way with voice or gesture? A few works tried to link static description

of a face (at a frame level) to high level information such as mental states. However, most works on mental states prediction focused on how to characterize the dynamics of facial movements in a relevant manner. For including these dynamic information, some researchers have used dynamic features combined with static classifiers. The features are often designed for capturing global human behaviors within more or less long time windows. However, time windows of fixed lengths may not be sufficient for characterizing the richness of facial movements. Thus, many researchers use dynamic classifiers or regressors combined with static features. In this state-of-the-art, we review a few static and dynamic methods for mental state prediction and discuss a few works that make use of multi-modal information.

Static prediction

Several researchers have been trying to predict emotions or other high-level information at a frame level. Wang and Guan [108] worked on classifying the six basic emotions in 2005 by extracting Gabor wavelets on faces and using Fisher Linear Discriminant Analysis (LDA) for classification. Kanluan et al. [49] proposed a system for predicting emotions in a 3D continuous space using Support Vector Regression (SVR) on 2D Discrete Cosine Transform (DCT) of different patches in facial images. Littlewort et al. [55] described static facial expressions using AU classifiers to design an automatic system for discriminating genuine pain from posed pain using SVM. Those static approaches are often limited to prototypical images. Indeed, for subtle mental states, it is often necessary to describe the subject movements within a longer time window. Recently, because in-the-wild predictions of subtle mental states can be difficult at a frame level, more and more works have been focusing on including dynamic information.

Dynamic features

Several works have been focusing on how to include relevant dynamic information in the features. In [105], the authors tried to classify posed versus spontaneous eyebrow activations. Among other features, they chose to extract the durations of AU activation. Indeed, acted behaviors may have different dynamics. Afterward, they performed feature selection using the Gentleboost algorithm and classified using a Relevant Vector Machine (RVM). Baltrusaitis et al. [1] used Local Binary Patterns on Three Orthogonal Planes (LBP-TOP) for describing the texture of a set of successive frames. Pantic et al. [77] used Motion History Images (MHI) for characterizing facial movements. Those dynamic feature approaches often characterize a set of images on a fixed temporal window. However, facial movements can have different dynamics and different durations. A solution for characterizing movements on different

scales is to use several time windows for dynamic feature extraction. During this PhD, we proposed a method for continuous emotion prediction that includes dynamic information extracting Fourier spectra (on different time windows from one to four seconds) of signals characterizing head and facial movements [75]. This work won the first place of the Audio-Visual Emotion Challenge AVEC'12. We present this work in appendix B. In order to have more flexibility for characterizing facial expression dynamics, many researchers have used dynamic machine learning models.

Dynamic models

In this paragraph, we present some of the main methods that have been used to predict high-level information from faces using dynamic learning models. Graphical models that capture the probability of transition between different states have been widely used for mental state prediction as Hidden Markov Models (HMM) or Dynamic Bayesian Networks (DBN). Those graphical probabilistic models have been widely used for classification tasks. In [8], geometric parameters are extracted using an AAM and then data is projected into a reduced space using a Lipschitz embedding. Afterwards, the classification of six discrete emotions is performed with a probabilistic model. The same method has also been used on 3D data in [9]. In 2003, Cohen et al. [12] used HMM for segmenting six discrete emotions after using Gaussian Tree-Augmented Naive Bayes (TAN) classifiers to learn the dependencies among different facial features. HMM have also been used for classifying eleven mental states in [125]. In [50], HMM have been used for detecting head nods and head checks from pupil tracking in order to predict frustration in intelligent tutoring systems. In 2005, El Kaliouby and Robinson [33] extracted AU and head movements using pre-defined geometric rules and used DBN for classifying six mental states (agreeing, concentrating, disagreeing, interested, thinking and unsure). Zhang and Ji [128] also combines DBN with FACS for modeling the dynamic behaviors of spontaneous facial expressions. Meng and Bianchi-Berthouze [65] worked on predicting mental states in a discretized four-dimensional space. Each mental state was either classified as positive or negative for each dimension. They used HMM for capturing the dynamics of facial movements and predicting each dimension independently and then fused the results using another layer of HMM. Their approach outperformed all other systems in the first Audio-Visual Emotion Challenge (AVEC'11).

HMM and DBN are used for modeling different classes separately. They aim at training one model for each class and the classification is performed afterwards using the likelihoods of generation of the test sequences by the different learned models. Recently, researchers have been focusing on predicting mental states in continuous spaces, considering the problem as a regression task. Various machine learning methods have been used for tackling this issue. In

[114], the authors extracted optical flow features and used Long Short Term Memory Neural Networks (LSTM) in order to capture dynamic information. Nicolaou et al. [72] proposed to use Correlated-Spaces Regression (CSR) on AAM parameters. Baltrusaitis et al. [1] included dynamic information using Conditional Random Fields (CRM) combined with SVR.

We can notice that various methods have been used for dynamic data modeling in order to predict mental states (HMM, DBN, CRM, LSTM). Even if those methods lead to a more flexible modeling of facial dynamics, they can be harder to learn than static methods. The number of parameters is often higher and the training requires an important number of labeled samples for them to be efficiently learned. Moreover, some of those methods (as HMM) can be hard to use with an important number of features because of the increased number of hidden parameters that have to be learned. This explains why dynamic information inclusion within the features or within the models are still both used for mental state prediction [65], [75]. In next paragraph, we present a few works that have been including different modalities for learning to predict mental states.

Multi-modal approaches

Because prosody and body language may contain relevant information about human mental states, many researchers used several modalities for designing prediction systems. Wang and Guan [108] used Mel-Frequency Cepstral Coefficients (MFCC) in order to characterize prosody for classifying six basic emotions. In 2008, Castellano et al. [7] worked on classifying eight mental states (anger, despair, interest, pleasure, sadness, irritation, joy and pride) using multi-modal information relative to the face, the body as well as prosody. They used Bayesian classifiers and showed that their multi-modal approach led to an improvement of 10% compared to their best unimodal system. Nicolaou et al. [71] used multi-modal information extracted from the voice and the face and shoulder areas and used Output-Associative RVM as regressor for predicting emotions in the four-dimensional continuous space. In our continuous emotion prediction method, we used MFCC features and proposed a fusion with facial features that is performed a posteriori.

1.4 Outline and contributions

In this chapter, we discussed different signals that may be extracted or inferred via automatic facial analysis, from low-level signals as facial landmarks that describe facial shape in a geometric manner to higher-level signals as mental states passing by mid-level signals as facial muscles activations. We presented some applications of machine learning systems based on facial analysis and discussed the main difficulties, as image-related issues (as

luminosity) and subject-related issues (as facial morphology or head pose). We presented a few recent challenges that have been organized on several topics, as landmark detection, AU prediction or emotional state prediction in order to fasten the evolution of the domain and make possible objective comparisons of the methods. Finally, We presented state-of-the-art methods that have been proposed in recent literature.

All the difficulties discussed in this chapter show the complexity of automatically predicting information from face-centered data. The data that has to be predicted for making work in-the-wild prediction systems is highly heterogeneous. Moreover, designing databases that are relevant for the tasks and labeling them is a long and hard work. As a consequence, the number of samples in databases for learning those systems may be limited, which adds complexity to the task.

We believe that the extreme variability of the data and the limited learning data size makes overfitting one of the main issues. In this PhD, we propose different regression frameworks based on a metric learning algorithm, namely Metric Learning for Kernel Regression (MLKR). Our methods have been designed for taking a particular focus on reducing potential overfitting in face-centered machine learning systems. It introduces a novel multi-task regularization that lets us reduce overfitting compared to a standard multi-task regularization by learning fewer parameters and similar complexity predictors. On chapter 2, we present our method and evaluate its relevance on synthetic regression datasets. We present complete prediction frameworks in chapters 3 and 4 that use our proposed extensions for landmark detection and AU intensity prediction. Finally, we conclude our work and discuss a few prospects in chapter 5 of this dissertation.

Appendix A contains a paper presenting a previous system we designed for AU intensity prediction that is based on a Lasso extension of MLKR. Appendix B contains our work on emotion prediction in a continuous space. Finally, appendix C contains a previous work we did on landmark localization. Two of our systems let us win two challenges, namely the second Facial Expression Recognition Challenge (FERA'15 [99]) and the second Audio-Visual Emotion Challenge (AVEC'12 [89]).

Chapter 2

Hard Multi-Task Metric Learning for Kernel Regression

In the first chapter, we discussed difficulties in automatic prediction of information from face-centered data. Among them, the choice of the features as well as the choice of an adequate model according to the task and the amount of available data are of primary importance. When the dataset size is small, training too much high complexity models can lead to what is called overfitting. The regression methods that we designed during this PhD aim at taking this overfitting issue into account.

In this chapter, we first introduce basic principles of machine learning. We define overfitting and present widely-used machine learning methods. Then, we present the algorithm on which our methods are based, namely Metric Learning for Kernel Regression (MLKR). In order to explain our choice of using that method, we analyze some of its characteristics relative to its training ease (its convexity), to the constraints resulting from its use in real-life applications (temporal and memory complexities) and to its power of generalization (robustness to overfitting, extrapolation capabilities of its prediction function). For that, we use synthetic data as well as a standard regression dataset called *pumadyn-32nh* that we introduce in subsection 2.3.4. Afterwards, we discuss the extensions that we propose for keeping the advantages of MLKR while trying to get rid of its drawbacks as much as possible. Among other modifications, we propose the use of a stochastic gradient descent, a Lasso-regularization and of a non-linear feature selection step. Moreover, in numerous face-related databases, several labels are at our disposal (e.g. labels corresponding to the intensities of several coded AU). Thus, we have been taking interest in what is called multi-task regularization. Multi-task regularization aims at reducing overfitting by virtually increasing the number of data samples by learning the different tasks in a joint manner (more details are given in subsection 2.4.4). We present in this chapter different extensions of

MLKR using multi-task regularization. One of them (Hard Multi-Task MLKR) lets us reduce the number of model parameters while training similar complexity models and thus reduce overfitting. In chapters 3 and 4, we evaluate the proposed algorithm on real data.

2.1 Introduction to machine learning

2.1.1 A few definitions

Machine learning is the study of algorithms that can learn from data. In computer vision, data are often images or video sequences. Let us consider each sample as a point in an euclidean space. For instance, a 240 by 240 pixels gray-level image can be considered as a point in a space of dimension 57600 whose coordinates are the intensity of each pixel. Some machine learning algorithms directly work in this raw space, where the image is described by its gray-level pixel values. However, different information (namely the features) can be extracted from an image in order to describe it. Features lead to projecting raw data into another space, called the feature space. For simple and fully understood tasks, it is possible to manually define direct links between features and labels but for complex tasks, it is often relevant to automatically estimate those links by analyzing labeled data.

Machine learning algorithms can be separated in two categories: supervised and unsupervised. Unsupervised machine learning algorithms work with data samples without labels on the contrary to supervised machine learning algorithms, that work with labeled data samples. As an example, an age prediction system where you make use of a dataset of images of people and their age is supervised. However, if you have unlabeled images and aim at organizing them into different groups for being able to access them more easily afterwards, you have an unsupervised learning task called a clustering task. In this PhD, we focused on supervised machine learning, where we had to learn to predict information using labeled data samples. There exist different categories of supervised learning algorithms. If the system aims at recognizing different classes (for instance different animals), the task is said to be a classification task. The labels are then discrete (corresponding to the different classes). The goal is to optimally separate the samples of the different classes for being able to easily recognize them. If the system aims at estimating the value of a continuous label (for instance the age of a subject), the task is said to be a regression task. The goal is then to learn a model leading to predictions that are close to the labels of the training data, namely the ground truth. In next subsection, we present different categories of commonly-used machine learning methods.

2.1.2 Different methods

In this subsection, we introduce three types of methods: one-class modeling methods, multi-class discriminative methods and regression methods.

One-class modeling methods let us learn models for simplifying the representation of a set of data samples. This means that they aim at modeling the data set using a reduced set of parameters. In many of them, the parameters of the model are estimated in order to minimize the reconstruction error (i.e. the error between the data generated by the model and the initial data) or to maximize the likelihood for probabilistic methods such as Hidden Markov Models (HMM) or Gaussian Mixture Models (GMM). The most common one-class modeling method is probably Principal Component Analysis (PCA). Using PCA, a linear orthogonal subspace is learned for minimizing the reconstruction error (that corresponds to the quadratic error between the projections of the data onto the subspace and the initial data). PCA can be used as a dimensionality reduction technique. Let us, as an example, consider a set of 100 images of size 100 by 100. Let us also consider that we succeed to explain 99% of the variance using 10 PCA components. The number of parameters used for describing the initial data was 10^6 and the number of parameters used for describing the projected data (that very closely approximate the initial data) is $10^5 + 10^3$. In that example, the number of parameters has been approximately reduced by a factor 10. One-class modeling methods can also be used for classification purposes by separately modeling the different classes and finally take a classification decision according to the probability of generation (or likelihood for probabilistic models) of the data by the different learned models.

Discriminative methods are specifically designed for classification tasks. They aim at optimally separating data points corresponding to the different classes. A commonly used discriminative method is the Fisher Linear Discriminant Analysis (LDA) [35] that aim at learning a linear subspace that maximizes the interclass variance (separating the different classes) and at the same time minimizes the intraclass variance (gathering same-class samples). Another widely used discriminant method is the binary classification method called Support Vector Machine (SVM). More details about this method are given in subsection 2.2.2. In [110], Weinberger proposed a discriminant metric learning method called Large Margin Nearest Neighbors (LMNN) that we present in subsection 2.2.3.

The third type of methods, regression methods, aim at learning a function whose inputs are the features and whose output is the label intensity value. The most common one is probably Linear Regression, that we introduce in subsection 2.2.2. Recently, Support Vector Regression (SVR) have been extensively used. We discuss SVR in subsection 2.2.2.

2.1.3 Understanding overfitting

In this subsection, we take an example of polynomial regression in order to introduce the notion of overfitting. We discuss overfitting and its relationship to the model complexity and to the training data size before explaining how overfitting is commonly estimated.

Overfitting and model complexity

For simplicity, let us consider a feature f and a label l corresponding to scalar values. Let us also consider that the label and the feature are linked by a polynomial relationship of order 2 (a , b and c being real-valued coefficients), according to:

$$l = a.f^2 + b.f + c$$

We consider a sampling of this function plus some white noise and we fit polynomials of different degrees to it. The obtained results are represented on figure 2.1. Solid green curves represent the initial polynomial. Blue points represent our sampled polynomial. Red dashed curves correspond to the estimated polynomials.

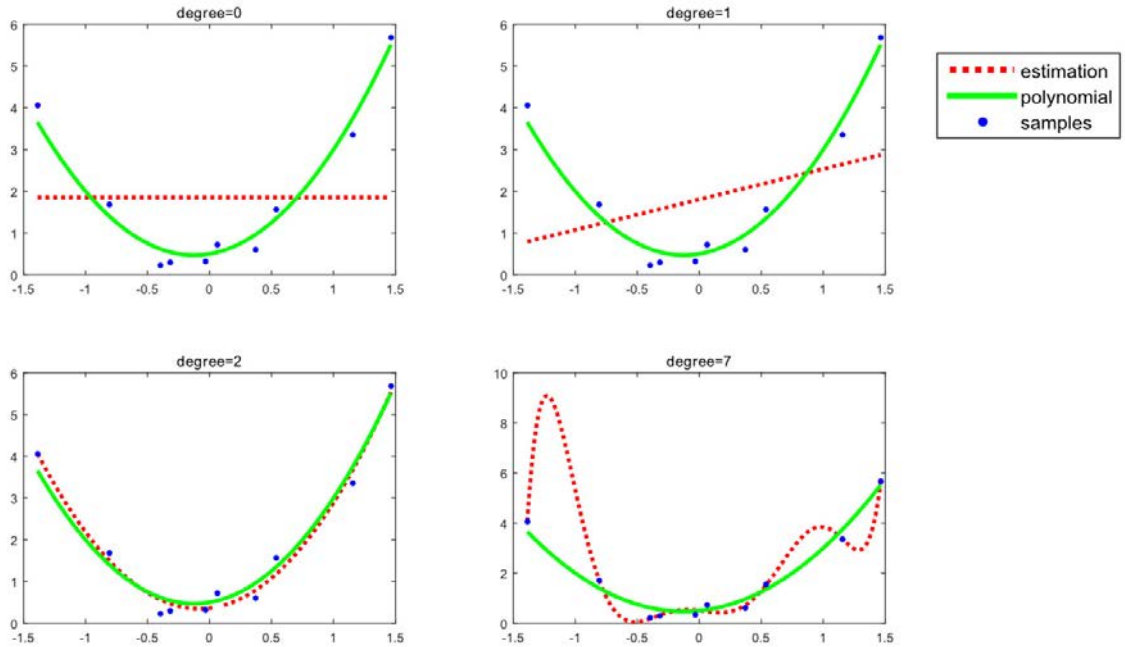


Fig. 2.1 Polynomial regressions of different degrees on sampled data points

Let us define two different errors. The training error corresponds to the error between the

estimated polynomial evaluated on the sampling points and the sampled polynomial. The true error corresponds to the error between the estimated polynomial and the initial polynomial. We can notice that the quality of the regression highly depends on the complexity of the prediction models. When the models are insufficiently complex to be able to represent the data (as degree zero and degree one for fitting a second order polynomial), we obtain what is called underfitting. The true error is large, as well as the training error. On the other hand, a too much high model complexity can result in what is called overfitting. The training error is small, but the true error is large. We can notice that the adequate complexity here corresponds to degree two, with both errors being small.

Overfitting and dataset size

The intensity of overfitting highly depends on the size of the dataset used for fitting the functions. To illustrate that, we plotted on figure 2.2 the true error and the training error of a degree nine polynomial regression for different number of data points. We can notice that with a sufficiently large amount of data, the overfitting issue can vanish even when the model complexity is too high relative to the prediction task.

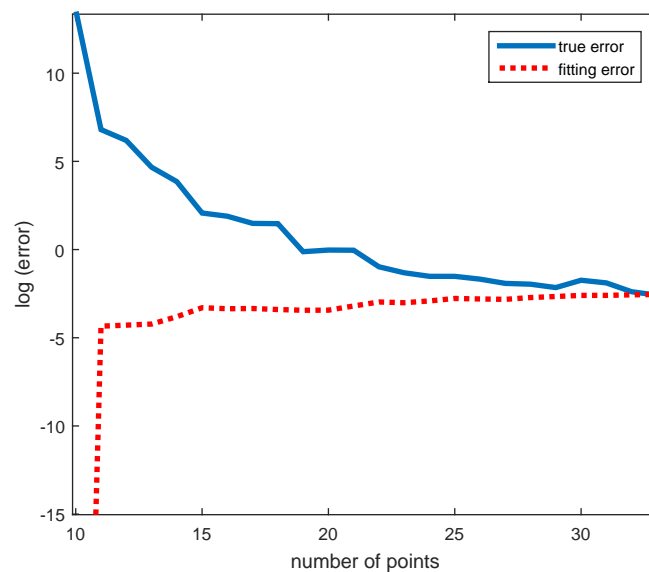


Fig. 2.2 Influence of the number of data points on the intensity of overfitting

We can also notice that with very few samples, the 9th order polynomial is very precisely fitted on the training samples, which results in a very small training error. However, as in the illustration of underfitting on figure 2.1, the areas between the sampling points are very poorly estimated, which results in a large true error. When the number of training samples

increases, the ten estimated parameters lead to a less accurate fitting on the sampling points, resulting in an increased training error. However, it leads to more relevant predictions as the estimated polynomials are closer to the true polynomial (the true error decreases). We presented this example to illustrate the fact that even without knowing the exact complexity of the function that we aim at learning, a complexity that is suited to the training data size can lead to relevant predictions.

How to estimate the overfitting rate?

In the context of machine learning, we only have a set of data samples at our disposal. The equivalent of the initial polynomial in the previous simulations is the true function that we aim to predict, which is by definition unknown when training. In order to be able to evaluate the relevant model complexity (that does not underfit nor overfit), the dataset is spitted into several parts. One part of the dataset is used for training the model (the training set) and another part is used for evaluating the performance of the model (the testing set). The testing set has to be unused during the training process in order for the performance evaluation to make sense. If we set a model complexity and learn the model parameters on the entire training dataset, we cannot prevent potential overfitting. Thus, the model complexity is often estimated in cross-validation on the training set. For estimating the model complexity on an n -fold cross-validation on the training set, we first split the training set into n separate parts. For different model complexities (e.g. for each polynomial degree), we want to evaluate if the model generalizes well on unseen data points (which means that the model results in a low overfitting rate). To do this, for each of the n parts of the training set, we learn the model on the $n - 1$ other parts and we test it on the unused part. The best complexity will be the one which results in a model obtaining the smallest errors on the unused parts. Then, the model is learned on the entire training set and evaluated on the testing set. This process, widely used in machine learning, aims at taking the best possible use of the available data and being confident about the estimated performance and the generalization power of the models.

However, even with a good model complexity estimation process, when the functions that have to be learned are complex and when the dataset size is narrow, the system performance is strongly linked to a relevant model choice and an adequate parameter estimation. In the next paragraph, we present different model types and discuss their advantages and drawbacks relative to overfitting issues.

2.2 Different model types

Different models can be used for supervised learning. We propose to divide them in three categories. In non-parametric models (e.g. k-NN, Nadaraya-Watson), the prediction is performed by using an estimator on the training data points embedded in the initial feature space. More details are given in subsection 2.2.1. In parametric models (e.g. Linear Regression, SVM, SVR), a function is learned for reducing the training error. This function is entirely defined by a set of parameters, and the prediction is performed without needing the training samples. In semi-parametric models (e.g. LMNN, MLKR), a subspace is learned for reducing the training error but the predictions are performed by embedding the training samples in the learned space.

In most parametric or semi-parametric learning models, parameters are estimated by minimizing a cost function on the training samples, also called energy or functional. It is most of the time a combination of two or three elements: a dissimilarity measure, an estimator (or prediction function), and sometimes a regularization term.

The dissimilarity measure quantifies how well the predictions performed by the learned model are close to the training labels. Different types of measures can be used. For instance, the quadratic error is very commonly used. Using that measure, the square of the difference between the prediction and the ground truth is used as a penalty. If l is the label and \hat{l} is the estimated label, the quadratic error dissimilarity measure, denoted e_{sq} , can be written as follows, with n the number of training samples:

$$e_{sq}(\hat{l}, l) = \sum_{i=1}^n (\hat{l}_i - l_i)^2$$

Instead of a dissimilarity measure, one can use a similarity measure (then, the energy should have to be maximized). We can for instance use the Pearson's correlation coefficient, denoted r , defined as (the bar notation denoting the mean):

$$r(\hat{l}, l) = \frac{\sum_{i=1}^n (l(i) - \bar{l})(\hat{l}(i) - \bar{\hat{l}})}{\sqrt{\sum_{i=1}^n (l(i) - \bar{l})^2} \sqrt{\sum_{i=1}^n (\hat{l}(i) - \bar{\hat{l}})^2}}$$

The second element of the cost function is the estimator, that defines the way the samples are predicted by the model. The third term, that is sometimes added to the cost function, is a regularization term. In order to reduce potential overfitting, this term penalizes models

that are too complex. In this section, we present different models belonging to the three previously introduced categories.

2.2.1 Non-parametric models

In non-parametric models, the predictions are performed by applying estimators without making use of a cost function. The k-Nearest-Neighbors estimator (k-NN) and the Nadaraya-Watson estimator are probably the most natural and simple prediction models for classification and regression respectively. Using those estimators, a test label is predicted using the labels of the training samples that lie close to the considered test sample in the feature space.

With the k-NN estimator, a test sample is classified by a majority vote of its neighbors, with the sample being assigned to the most frequent class among its k nearest neighbors.

With the Nadaraya-Watson estimator, a test label is predicted as an average of the training labels weighted by a similarity measure between the considered test sample and training samples. Considering n_s training samples $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{n_s}\}$ and their corresponding labels $\{y_1, y_2, \dots, y_{n_s}\}$, the label associated to a feature vector \mathbf{x}_t is estimated using:

$$\hat{y}_t = \frac{\sum_{i=1}^{n_s} y_i k_{i,t}}{\sum_{i=1}^{n_s} k_{i,t}} \quad (2.1)$$

The kernel $k_{i,t} = k(\mathbf{x}_i, \mathbf{x}_t)$ corresponds a similarity metric between i and t samples.

The most commonly used kernel is the Gaussian kernel, defined as follows:

$$k_{i,j} = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{d_{i,j}^2}{\sigma^2}} \quad (2.2)$$

with σ the Gaussian spread and $d_{i,j} = d(\mathbf{x}_i, \mathbf{x}_j)$ the euclidean distance between i and j samples. Using a Nadaraya-Watson estimator with a Gaussian kernel, the closest samples have more impact on the prediction because the Gaussian kernel is a similarity measure (the closer the samples are, the higher the Gaussian kernel is).

Despite the name, when using those algorithms, one parameter must be estimated in cross-validation (the spread of the Gaussian or the number k of samples respectively). Because those methods only need to optimize one parameter, they have a very small chance of overfitting. Moreover, those estimators are able to predict non-linear relationships between features and labels as they are in a sense local predictors. However, for them to work in a precise manner, the feature space needs to be relevant to the considered task. Indeed,

the distances between samples in the feature space are the central elements that define the prediction. In a regression context, Metric Learning for Kernel Regression (MLKR) was proposed to solve this issue. The algorithm aims at learning a relevant subspace specifically for the Nadaraya-Watson estimator. Our regression frameworks are based on this algorithm that will be presented in section 2.3.

2.2.2 Parametric models

In parametric models, a set of parameters is learned by minimizing a cost function in order to define a function linking features and labels. Those parameters are then sufficient to predict a test sample without needing the training samples. In this subsection, we present three widespread parametric algorithms: Linear Regression, Support Vector Machine (SVM) for binary classification and Support Vector Regression (SVR).

Linear Regression

Linear Regression aims at learning a linear prediction function for linking the features to the label. In order to learn the parameters of this linear estimator, the dissimilarity measure used in Linear Regression is the quadratic error. Let $\{x_t^i, i \in \llbracket 1; n_f \rrbracket\}$ be the coordinates of a test point in a feature space of dimension n_f . With Linear Regression, its label is predicted as (with \mathbf{x}_t a line vector) : $\hat{y}^t = \mathbf{x}_t \beta = \sum_{i=1}^{n_f} x_t^i \beta_i$ with β the column vector of parameters of the Linear Regression. Those parameters β are estimated using the training set. Let \mathbf{X} be a matrix containing training data samples (each line represents one sample) and \mathbf{y} be the column vector corresponding to the associated labels. Ordinary Least Squares (OLS) minimizes the quadratic error L defined as:

$$L = \sum_j (\mathbf{x}_j \beta - y^j)^2$$

using the following estimator:

$$\hat{\beta} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$$

Previously, we noticed that overfitting was linked to the training data size and the number of estimated parameters. To be more precise, it is linked to the training data size and the complexity induced by the parameters. For example, when fitting a 8^{th} order polynomial on a set of points, there is nine parameters to optimize but the parameter corresponding to the 8^{th} order monomial induces a complexity that is way more important than the complexity induced by the constant parameter, or by the parameters of a Linear Regression. Thus, when

the space dimension is low compared to the number of training samples, Linear Regression has a very small chance of overfitting because of the simplicity of its prediction function. However, overfitting can still occur in very high dimensional spaces. In order to be able to learn a relevant estimator in this case, a regularization term can be added to the cost function. This term aims at penalizing models that are too complex. For instance, it is common to use a quadratic penalization of the parameters, also known as the Tikhonov regularization (or the L2-regularization). Adding this term to a Linear Regression is called ridge regression. In ridge regression, a linear function is learned in order to precisely predict the training samples and at the same time to have a parameter vector with the smallest L2-norm. Another common regularization makes use of the L1-norm and is called the Lasso regularization. Those constraints induced by regularization terms encourage the model parameters to be as small as possible which results in more smooth models and can lead to reducing overfitting. More details are given in subsection 2.4.3, along with our proposed Lasso extension of MLKR.

Support Vector Machine

Support Vector Machine (SVM) is a binary classification method, i.e. it is used for separating samples that belongs to two different classes. SVM aims at linearly separating the data samples in some space. The parameters that have to be optimized define a hyperplane that optimally separate the samples of the two classes. The criteria used for estimating the quality of the separation is the maximization of the margin. If the data is separable, i.e. it exists a hyperplane that results in a good classification of all the samples, the margin is defined as the distance between the closest sample and the hyperplane. Maximizing the margin leads to maximizing the gap between both classes. The algorithm input is a matrix, called kernel matrix, that contains the similarities between the samples. If, for instance, the euclidean scalar product is used for defining the similarities contained in the kernel matrix, SVM defines a hyperplane in the initial feature space and leads to a linear classification. However, other kernel functions can be used for estimating the similarities (as polynomial kernels or the mainly used Gaussian kernel). Using those functions leads to classification performed according to a non-linear separation surface.

Even if SVM is a binary classification method (designed for a two classes problems), several methods have been proposed in order to use it for multi-class learning problems. Two of the main proposed methods are the 'one VS all' and the 'one VS one' methods. In the 'one VS all' method, several SVM are learned for separating each class from a virtual class gathering all other classes. In the 'one VS one' method, one SVM is learned for separating each couple of classes before combining the results for taking the final classification decision.

An extension of SVM for regression has been proposed, called Support Vector Regression (SVR) that we introduce in subsection 2.2.2.

In this paragraph, we explain how the optimization works for linear classification with what is called a hard margin. The hyperplane is estimated for leaving the largest possible fraction of points of the same class on the same side, while maximizing the distance of either class from the hyperplane. Let us consider an euclidean space of dimension n . An hyperplane of parameters \mathbf{w} and b is defined as:

$$\{\mathbf{x} \in \mathcal{R}^n | \mathbf{w} \cdot \mathbf{x} + b = 0\}$$

Let $\{\mathbf{x}_i, y_i \in \{-1; 1\}, i \in \llbracket 1; N \rrbracket\}$ be the set of N points and corresponding binary labels. When all data points are correctly classified, the following equation is verified:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) > 0, \forall i \in \llbracket 1; N \rrbracket$$

By an appropriate rescaling of the hyperplane parameters, we obtain:

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) > 1, \forall i \in \llbracket 1; N \rrbracket$$

Under this assumption on \mathbf{w} and b , the distance between the hyperplane and the closest point is $\frac{1}{\|\mathbf{w}\|}$. The SVM algorithm aims at maximizing this distance.

The SVM optimization can be formulated as follows:

$$\min_{y_i(\mathbf{w} \cdot \mathbf{x}_i + b) > 1, \forall i \in \llbracket 1; N \rrbracket} \|\mathbf{w}\|^2$$

This problem of quadratic minimization under linear constraints is solved using Lagrange multipliers. More details can be found in [21].

Using SVM, the estimator for a test sample \mathbf{x}_t is:

$$\hat{y}_t = \text{sign}(\mathbf{w} \cdot \mathbf{x}_t + b)$$

The SVM constraint satisfaction induces the fitting between the prediction and the ground truth. We can notice that the maximization of the margin corresponds to a penalization of the hyperplane parameters, which play the role of a quadratic regularization term, leading in a sense to select the simplest model that satisfies the constraints. However, this hard margin optimization problem is suited for a configuration that is linearly separable. In practice, it is very unlikely. Thus, the soft margin optimization problem has been proposed. Using a soft margin, the constraints are relaxed using slack variables ξ . The new optimization problem

becomes:

$$\min_{y_i(\mathbf{w} \cdot \mathbf{x}_i + b) > 1 - \xi_i, \xi_i > 0, \forall i \in \llbracket 1; N \rrbracket} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i$$

with C a strictly positive constant.

The constant C plays the role of a cursor between the acceptable model complexity and the constraint satisfaction. This parameter can be optimized in cross-validation in order to select the model that leads to the minimal overfitting rate.

Because of the possibility of easy model selection and the possibility of non-linear prediction, SVM have been widely and successfully used in numerous applications.

Support Vector Regression

Support Vector Machines for classification have been extended for regression tasks in [26]. In Support Vector Regression, the label of a test sample \mathbf{x}_t is estimated as:

$$\hat{y}_t = (\mathbf{w} \cdot \mathbf{x}_t + b)$$

There exist several variants of SVR. In the mainly used ε – SVR variant, the minimization problem is the following:

$$\begin{aligned} \min_{\mathbf{w}, b} \quad & \|\mathbf{w}\|^2 \\ \text{subject to} \quad & (\mathbf{w} \cdot \mathbf{x}_i + b) - y_i < \varepsilon \\ & y_i - (\mathbf{w} \cdot \mathbf{x}_i + b) < \varepsilon \end{aligned}$$

This problem corresponds to the 'hard margin' version, which means that the constraints aim at predicting all training samples with an error lesser than a fixed constant ε . As for SVM, a 'soft margin' version with constraints that are relaxed by slack variables has also been proposed for SVR.

2.2.3 Semi-parametric models

Semi-parametric models make use of a cost function for optimizing a set of parameters. However, on the contrary to parametric models, these parameters alone are not sufficient for predicting test samples. The estimator makes use of the training data samples in addition to the learned parameters.

In this subsection, we present a metric learning algorithm called Large Margin Nearest

Neighbors (LMNN) that has been proposed by Weinberger et al. [110]. The core idea of the algorithm is to estimate an Mahalanobis metric suited for a k-NN classifier. This algorithm translates the maximum margin learning principle behind SVM to k-nearest neighbors classification. It is based on a convex optimization of a cost function which is the sum of two weighted terms:

$$\varepsilon^1(\mathbf{M}) = \sum_{ij} \eta_{ij} d_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j)^2$$

and

$$\varepsilon^2(\mathbf{M}) = \sum_{ijl} \eta_{ij}(1 - y_{il}) \{1 + d_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j)^2 - d_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_l)^2\}_+$$

Notations are the followings: η is a binary variable representing whether two samples are neighbors or not (this way the first term penalizes large distances between each sample and its neighbors), y is a binary variable indicating whether two samples are same-class samples or not and $\{a\}_+ = \max(a, 0)$ (this way the second term penalizes small distances between each sample and all other samples that are from different classes).

The cost function is defined as follows:

$$\varepsilon(\mathbf{M}) = \varepsilon^1(\mathbf{M}) + \varepsilon^2(\mathbf{M})$$

Minimizing this cost function, LMNN learns a matrix \mathbf{M} defining a distance function (which implicitly defines a linear subspace) that is used for k-NN estimation. On the contrary to many multi-class discriminative algorithms, such as Fisher Linear Discriminant Analysis (LDA), LMNN is local. Indeed, the constraints are only defined between neighbors, which makes the learned subspace suited for a k-NN estimator. Because the notion of neighborhood depends on the space in which the samples lie, neighbors are updated according to the current distance matrix during the optimization process.

In a similar way, Weinberger and Tesauro [111] proposed a metric learning algorithm for regression tasks estimating a subspace suited for a Nadaraya-Watson estimator (MLKR). More details are given in the next section.

We believe that those semi-parametric models lead to an interesting compromise between the complexity of the prediction function and potential overfitting. Indeed, the learned parameters only induce a linear distortion of the initial feature space but the prediction is non-linear. This explains our choice to use MLKR in the frameworks that we designed in this PhD.

2.3 Metric Learning for Kernel Regression

In this section, we present and discuss the MLKR method on which our proposed regression methods are built upon. We first introduce the algorithm. Afterwards, we discuss issues caused by its non-convexity. Then, we discuss its training time and memory complexities and its robustness to overfitting. Finally, we discuss the extrapolation capabilities of its prediction function.

2.3.1 The MLKR method

Using the MLKR method, a test label is predicted with the Nadaraya-Watson estimator [70]. As we previously discussed, the space in which the samples lie has an important impact on the prediction quality, making dimensionality reduction a relevant initial step. The goal of the MLKR method is to estimate the optimal linear subspace for minimizing the Nadaraya-Watson squared error on the training set with the commonly used Gaussian kernel. Considering an initial space of dimension n_d and an reduced space of dimension n_r , The MLKR method estimates the projection matrix $\mathbf{A} \in \mathcal{M}_{n_r, n_d}(\mathbb{R})$ that minimizes the following error:

$$\mathcal{L}(\mathbf{A}) = \sum_{i=1}^{n_s} (\hat{y}_i - y_i)^2 \quad (2.3)$$

where

$$\hat{y}_i = \frac{\sum_{j \neq i} y_j k_{j,i}(\mathbf{A})}{\sum_{j \neq i} k_{j,i}(\mathbf{A})}$$

with

$$k_{i,j}(\mathbf{A}) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{d_{i,j}(\mathbf{A})^2}{\sigma^2}}$$

$$d_{i,j}(\mathbf{A})^2 = \|\mathbf{A}(\mathbf{x}_i - \mathbf{x}_j)\|^2 = (\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{A}^\top \mathbf{A} (\mathbf{x}_i - \mathbf{x}_j)$$

being the squared distance in the reduced subspace of dimension n_r . The minimization of this error is performed using a gradient descent, with:

$$\frac{\partial \mathcal{L}(\mathbf{A})}{\partial \mathbf{A}} = 4\mathbf{A} \sum_i \frac{(\hat{y}_i - y_i)}{\sum_{j \neq i} k_{ij}} \sum_j (\hat{y}_i - y_j) k_{ij}(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top \quad (2.4)$$

When using the MLKR method, two hyper-parameters have to be estimated. First, the parameter n_r defining the size of the reduced subspace. Second, the parameter σ defining the spread of the Gaussian kernel. We discuss those hyper-parameters effects in next subsections.

During test, the projection matrix \mathbf{A} is used for embedding the training and the test samples in the reduced subspace. Afterwards, the predictions of the test samples are performed using the Nadaraya-Watson estimator. Thus, this method is a semi-parametric model as it needs both the learned parameters and the training samples for computing test predictions.

In the next subsection, we present simulations performed on synthetic data that aim at evaluating the MLKR method regarding its convexity issues.

2.3.2 About MLKR convexity

On the contrary to SVM, the MLKR optimization is non-convex. However, depending on the shape of the cost function, non-convexity may be a more or less important issue. For instance, if the cost function has a relatively small number of local minima with same energy values, the non-convexity may not even be an issue. However, if the cost function is highly non-convex with dense local minima having various energy values, the non-convexity may lead to a very inconvenient optimization process. In this subsection, we present simulations for evaluating the difficulties caused by MLKR non-convexity and discuss the parameters that impact the gradient descent.

We defined a function linking label and features in a non-linear manner with a sufficient amount of non-linearity to induce issues in the optimization process. Let $\{f_i, i \in \llbracket 1; n_f \rrbracket\}$ be a set of n_f features. The label is defined according to:

$$label = k \cdot \cos(a_1 \cdot f_1 + b_1) \cdot \cos(a_2 \cdot f_2 + b_2) + \cos(a_3 \cdot f_3 + b_3)$$

with $k, a_1, a_2, a_3, b_1, b_2, b_3$ all being real-valued parameters. The label and three of the features are linked by a non-linear relationship. In order to have an idea on the induced non-linearity, we represented on figure 2.3 the label (corresponding to point colors) with respect to the three related features using 600 randomly generated data points. We can notice that the small intensity labels are located in three different areas of the space. We designed three different tests for evaluating the impact of the kernel variance, the amount of noise and the number of data points.

The first test aims at evaluating the influence of the kernel variance on the convexity of MLKR. We use 600 randomly generated data points on a five-dimensional space. We run the MLKR algorithm using 20 random initializations for different kernel variances and stored the mean squared errors. We represented on figure 2.4 the mean and variance of the obtained errors. In this test, the best kernel variance is $\frac{1}{10}$ because it leads to the smallest mean error as well as the smallest variance. The large differences that exist between the different kernel variances show that the kernel variance has to be adapted to the data in order for the

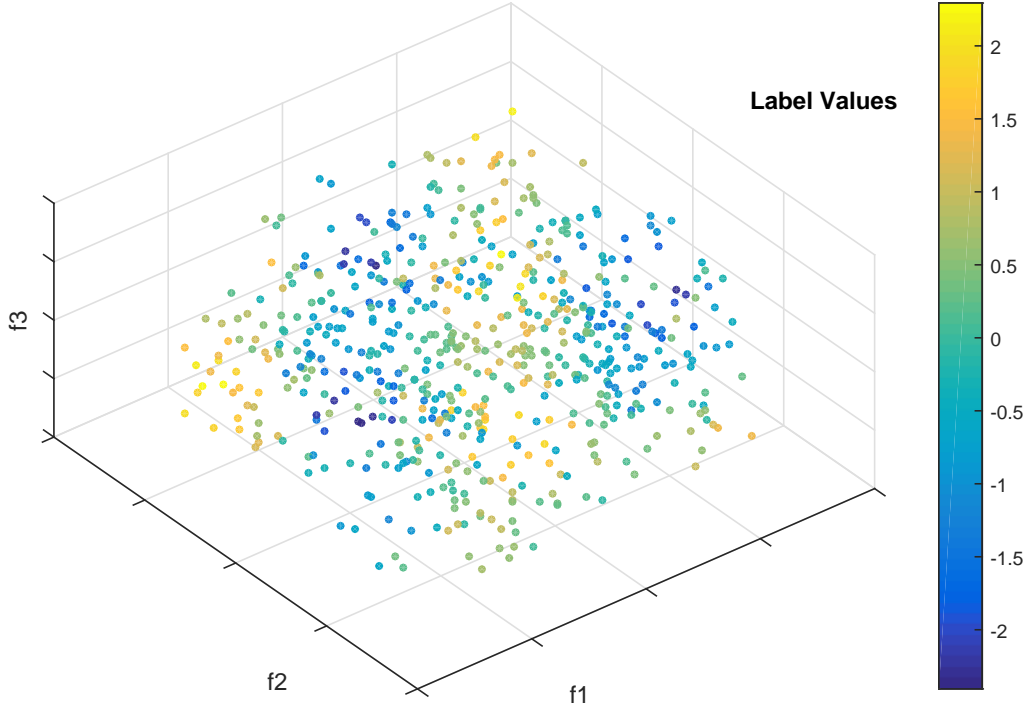


Fig. 2.3 Illustration of our toy example

optimization process to be efficient. At first sight, this seems illogical as multiplying the parameter matrix \mathbf{A} by a real-valued coefficient should make up for the kernel variance. This only remains true within some acceptable range for the kernel variance. However, a too small or a too large kernel variance can cause numerical issues in the optimization process, because of unsuitable initializations. When the kernel variance is too small, each point seems to have (numerically) only one neighbor. Indeed, we have:

$$\forall \{(a, b) \in \mathbb{R}^2 | b > a > 0\}, \lim_{\sigma \rightarrow 0^+} \frac{e^{-\frac{a}{\sigma}}}{e^{-\frac{b}{\sigma}}} = +\infty$$

This means that the dissimilarity measure between a point and its closest neighbor is infinitely higher than the dissimilarity between the point and its second closest neighbor when the kernel variance tends to zero. This results, when the kernel variance is too small, in a gradient estimation at each step that is not impacted by a sufficient number of neighbors for each point and prevents an adequate optimization. A similar issue happens when the kernel variance is

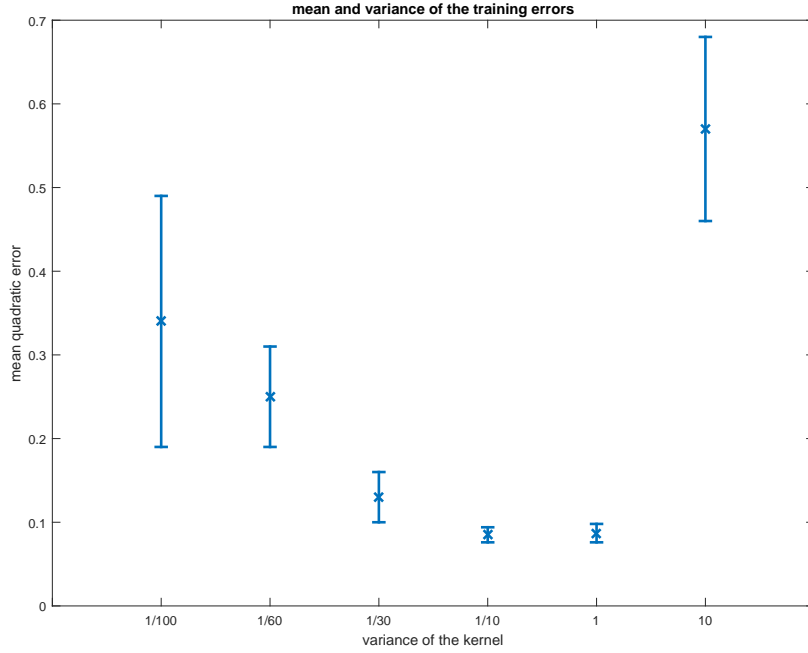


Fig. 2.4 Mean and variance of the training errors for different kernel variances

too high because all points seem (numerically) equally distanced because we have:

$$\forall \{(a, b) \in \mathbb{R}^2 | b > a > 0\}, \lim_{\sigma \rightarrow +\infty} \frac{e^{-\frac{a}{\sigma}}}{e^{-\frac{b}{\sigma}}} = 1$$

This results in a gradient estimation that stays constant at each iteration which also prevents an adequate optimization. Those effects occur frequently in practice, which makes an estimation of the kernel variance suitable range mandatory for an efficient training. This range can be approximated in cross-validation.

Our second test aims at evaluating the impact of noisy features on the convexity of MLKR. In our first test, there were two noisy features because the label was related to three of the five randomly generated features. In this second test, we evaluate the convergence rate for different number of noisy features. We still use 600 data points and 20 random initializations. We consider that a convergence is successful when the obtained mean squared training error is inferior to 0.1. In order to have an idea of corresponding predictions for our label range, we represented on figure 2.5 a prediction with that 0.1 mean squared error. The best kernel variance has been selected for each number of noisy features as the squared distances between points change with the space dimension. We use the following vector for optimizing the kernel variance in cross-validation: $[\frac{1}{100}, \frac{1}{60}, \frac{1}{30}, \frac{1}{10}, 1, 10]$. We represented the obtained

results on figure 2.6. We can notice that the convergence rate is reduced when adding noisy features because an important number of noisy samples can result in inconsistent gradient estimations.

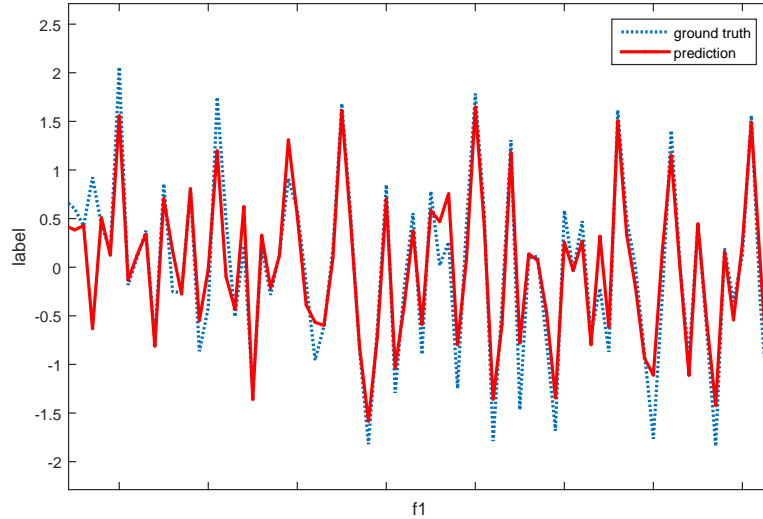


Fig. 2.5 Prediction of the label with a 0.1 mean squared error

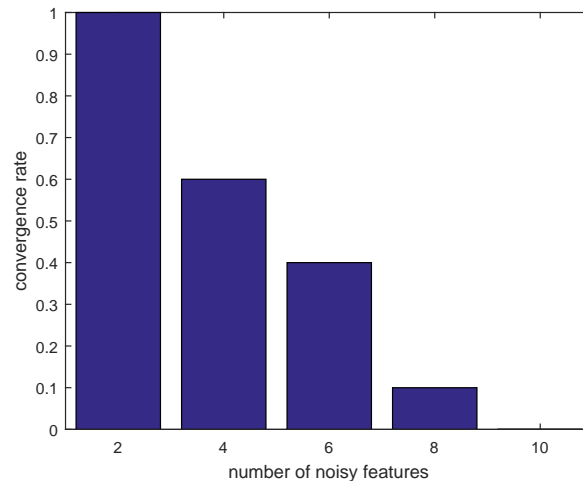


Fig. 2.6 Convergence rate for 600 samples and different number of noisy features

Our third test aims at evaluating the impact of the number of data points on the convergence rate. We consider 10 noisy features. Using 600 points, the algorithm did not succeed to converge even one time upon the 20 random initializations. In this test, we evaluated the

convergence rate for different number of sampling data points. We represented the results on figure 2.7. We can notice that with a sufficient number of data points, MLKR seems to be able to perform efficiently even with important amount of noise. Indeed, using a larger amount of samples, the gradient estimation is somehow smoothed.

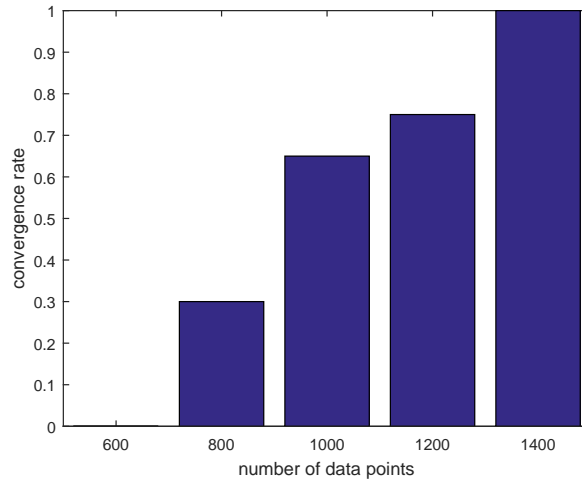


Fig. 2.7 Convergence rate for different number of data points and 10 noisy features

Those tests show that the non-convexity of MLKR can be an issue in some cases. However, with a sufficient number of data samples, a controlled amount of noisy features and an optimization of the kernel variation performed in cross-validation, the impact of those optimization issues can be reduced, at least for the amount of non-linearity of our toy example.

2.3.3 About MLKR complexity

In this subsection, we present simulations for evaluating both the memory and time complexities of MLKR.

The limiting memory factor of MLKR is the size of the kernel matrix. The storage of a 15.000 samples kernel uses approximatively 6 GB of RAM.

In MLKR, the gradient computation for a projection of n_s samples into a space of dimension n_d has a complexity in $O(n_s^2 \cdot n_d^2)$ making it difficult to use in high dimensional spaces. On table 2.1, we represented the time until convergence for different number of features and data points. We used a Matlab implementation on an Intel i7-3770, 3.4 GHz for computing the results. Those tests show that, as many kernel methods like SVR, MLKR does not handle important number of samples, both because of its memory and time complexities. Moreover,

Table 2.1 Time until MLKR convergence for different number of features and data points

| Nb of points | Nb of features | Time (s) |
|--------------|----------------|----------|
| 100 | 3 | 0.2 |
| 200 | 5 | 0.7 |
| 600 | 7 | 4.6 |
| 1600 | 13 | 65 |
| 3000 | 30 | 975 |

those tests also show that MLKR is neither designed for working with a high number of features. We discuss this issue along with our extensions in section 2.4.

2.3.4 About overfitting

In this subsection, we evaluate the impact of the parameter n_r (defining the dimension of the reduced space) on overfitting. This parameter is linearly linked to the number of parameters that have to be optimized as matrix \mathbf{A} is of size $n_d \cdot n_r$. As pointed in subsection 2.2.3, we believe that MLKR, as a semi-parametric method, is less prone to overfitting than fully parametric methods. However, when the number of parameters that have to be learned is too high relative to the number of data samples, overfitting can still occur. In order to evaluate this effect, we performed a test on the standard *pumadyn-32nh* regression dataset. It consists of a realistic simulation of the dynamics of a 'Puma 560' robot arm. The task is to predict the angular acceleration of one of the robot arm links. The inputs include angular positions, velocities and torques of the robot arm. It contains 8192 samples with 32 inputs, non-linearities and a high amount of noise. We learned on two third of the dataset and tested on the other third. We learned reduced spaces of different dimensions and plotted the training and test errors on figure 2.8. The optimal reduced subspace dimension appears to be two as the test error is minimal. However, we notice that for higher dimensions, the training error can be smaller. This illustrates that overfitting can occur using MLKR when the number of parameters that have to be learned is too high. Here, a space of dimension one leads to underfitting (both training and test errors are large). In order to select the appropriate model, this hyper-parameter n_r has to be estimated in cross-validation.

2.3.5 About Nadaraya-Watson extrapolation capabilities

In this subsection, we discuss the extrapolation capabilities of the Nadaraya-Watson estimator. The Nadaraya-Watson prediction function differs a lot from the prediction function of the

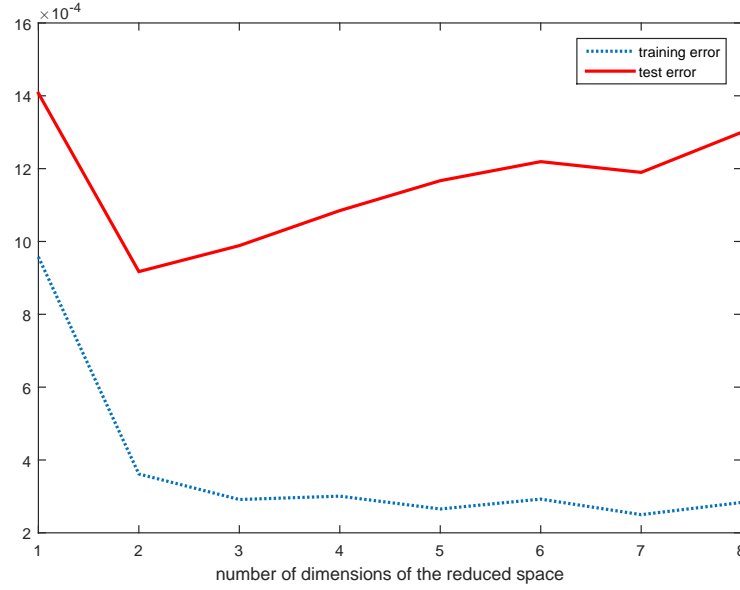


Fig. 2.8 Training and test errors for different reduced space dimensions

mainly used Gaussian kernel SVR regarding extrapolation capabilities. In Gaussian kernel SVR, the prediction function is the following:

$$\hat{y}_t = \alpha_0 + \sum_{i=1}^{n_{sv}} \alpha_i K(x_i, x_t)$$

with \hat{y}_t the estimated label of the test sample x_t , $\{\alpha_i, i \in \llbracket 1; n_{sv} \rrbracket\}$ the coefficients corresponding to the n_{sv} support vectors $\{x_i, i \in \llbracket 1; N \rrbracket\}$, $K(x_i, x_t)$ the similarity between x_i and x_t and α_0 a constant.

We recall that the prediction function of the Nadaraya-Watson estimator is:

$$\hat{y}_t = \frac{\sum_{i=1}^{n_s} y_i K(x_i, x_t)}{\sum_{i=1}^{n_s} K(x_i, x_t)}$$

with $\{x_i, i \in \llbracket 1; n_s \rrbracket\}$ the n_s training samples and $\{y_i, i \in \llbracket 1; n_s \rrbracket\}$ the corresponding labels. The main difference between those two prediction functions is caused by the normalization by $\sum_{i=1}^{n_s} K(x_i, x_t)$ that is included in the Nadaraya-Watson estimator. Gaussian kernel SVR models the areas where training samples are located and the farther a test sample is from those areas, the more the prediction tends to the constant α_0 . Indeed, all support vectors are far from the test sample, and their similarity measures with it tend to zeros. Using the

Nadaraya-Watson estimator, a test sample located far away from the training sample areas tends to be predicted as the label of the closest point. Indeed, we have:

$$\forall \{b > 0\}, \lim_{a \rightarrow +\infty} \frac{e^{-\frac{a}{\sigma}}}{e^{-\frac{a+b}{\sigma}}} = +\infty$$

Thus, the predictions performed by the Nadaraya-Watson estimator and by the SVR prediction function will particularly differ space areas that are sparsely populated by the training points. As an illustration, we designed a toy example in a two-dimensional space with training samples and labels distributed as plotted in figure 2.9.

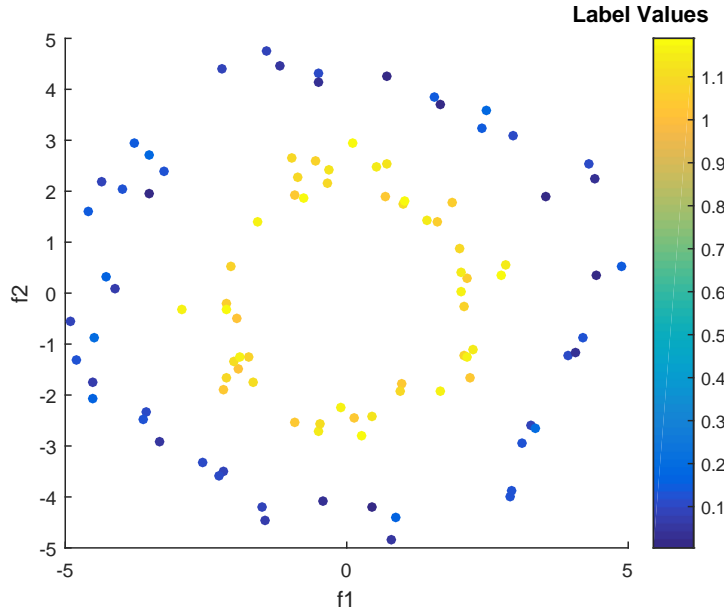


Fig. 2.9 Distribution of our toy example training points and corresponding labels represented as colors

In figure 2.10, we represented the obtained predictions for the two different prediction functions.

We can notice the expected extrapolations. Using the SVR prediction function, the test samples that are located far from the training points tend to the constant α_0 and using the Nadaraya-Watson estimator, the test samples that are located far from the training points tend to be estimated as the closest sample label. Those differences have motivated our choice of using the MLKR method in our frameworks. We think this extrapolation is more natural for our considered applications. Moreover, it frequently occurs, in face-related machine learning application, that a test subject samples lie in a relatively sparsely populated area.

In this section, we discussed the MLKR method and its characteristics relative to convexity,

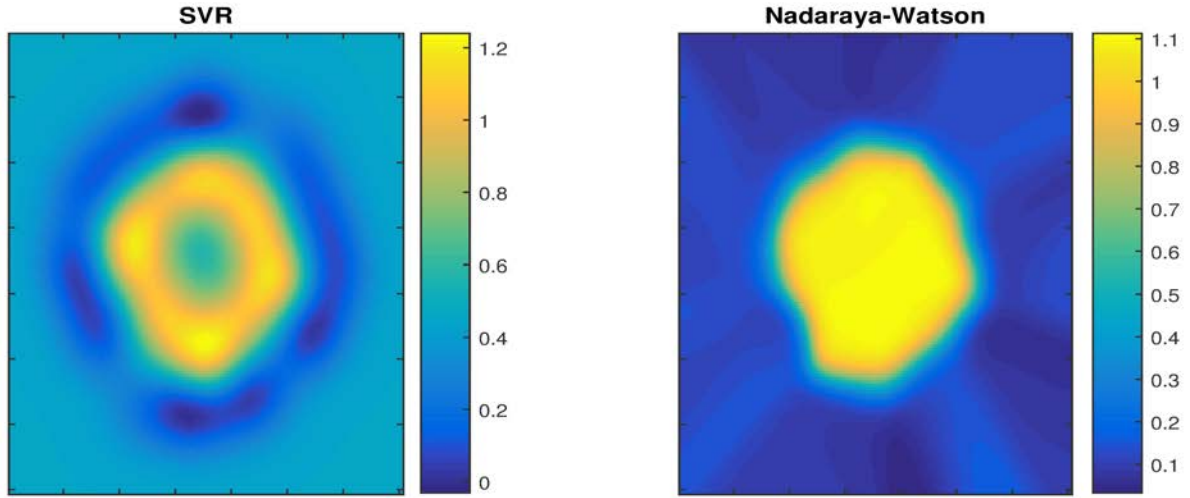


Fig. 2.10 Label estimation using the SVR prediction function and the Nadaraya-Watson estimator

complexity, overfitting and extrapolation. We also discussed some of its limitations regarding acceptable number of features and samples. The MLKR method, as a semi-parametric subspace learning method, has an interesting property regarding the possibility of embedding samples that have not been used for training the model in order to make the predictions. We made use of that property for designing our landmark detection framework (chapter 2).

In the next section, we present the extensions that we designed in order to make use of the MLKR method in an efficient manner. We evaluate some of those proposed modifications on the previously introduced *pumadyn-32nh* dataset.

2.4 Our extensions

As discussed in the previous section, the MLKR method has several drawbacks as its important complexity, its non-convexity that appears to be worsen in the presence of noisy features, and its potential overfitting. In this section, we present our extensions for letting MLKR be used efficiently in the context of facial-related machine learning regression tasks. We first introduce a non-linear feature selection step and a stochastic gradient descent. Then, we present a Lasso-regularization of MLKR. Finally, we present three different multi-dimensional label extensions of MLKR. The first one is a multi-dimensional label version called Common-Space MLKR (CS-MLKR). The second one is a standard multi-task version called Multi-Task MLKR (MT-MLKR), and the last one is a modification of the standard multi-task regularization that allows us to learn similar complexity models with a

lesser number of parameters (and thus reduces overfitting), called Hard Multi-Task MLKR (H-MT-MLKR).

2.4.1 Feature selection

In many facial-related machine learning tasks, we deal with a large number of features. Because of the complexity of MLKR in $O(n_f^2 \cdot n_s^2)$ with n_s the number of training samples and n_f the number of features, using all the features with a large number of training samples can quickly become unpractical. We also noticed that having noisy features may increase the number of local minima which makes the non-convexity of MLKR an issue for optimization (see subsection 2.3.2). Because of those reasons, we think that a pre-processing step of feature selection seems relevant when using MLKR.

The purpose of supervised feature selection is to identify features that contain relevant information for predicting a label. Numerous methods for feature selection have been proposed in the literature. Among them, filter methods select variables by ranking them using similarity measures between features and labels. They characterize the relevance of the features independently of the predictor choice, often one by one, for predicting the label. In other words, it means to compute a similarity (or dissimilarity) measure between each feature and the label and to select the highest ones (or the smallest ones respectively). Other methods evaluate and select subsets of features (sometimes iteratively). Among them, wrapper methods assess subsets of features directly according to a given predictor. The purpose of this paragraph is not to give a complete view of feature selection methods but to introduce two similarity measures and propose a scheme for feature selection that may be used with MLKR. An introduction to feature selection can be found in [40]. We chose to use a filter approach because of the learning complexity of our method that would make the use of wrapper-like approaches too much time-consuming.

Different prior assumptions can be made on the functional relationship existing between features and label. The simplest prior assumption that can be made between features and label is linear dependency. The similarity measure associated with that dependency is Pearson's correlation coefficient, defined as follows between a feature f and a label l :

$$r(f, l) = \frac{\sum_{i=1}^n (f_i - \bar{f})(l_i - \bar{l})}{\sqrt{\sum_{i=1}^n (f_i - \bar{f})^2} \sqrt{\sum_{i=1}^n (l_i - \bar{l})^2}}$$

That prior of linear dependency is not suited for MLKR because of the non-linearity of the Nadaraya-Watson estimator. Several similarity measures exist for quantifying non-linear

functional relationships. We decided to use conditional entropy as dissimilarity measure. Conditional entropy is defined as follows (for a label l given a feature f):

$$H(l|f) = - \sum_{x \in \mathcal{F}} p(x) \sum_{y \in \mathcal{L}} p(y|x) \log[p(y|x)]$$

with \mathcal{F} and \mathcal{L} the sample spaces in which the feature and label are respectively defined. Because fine estimations of conditional probabilities can be time-consuming with a high number of samples, a common approximation is computed with quantizations of the features and labels.

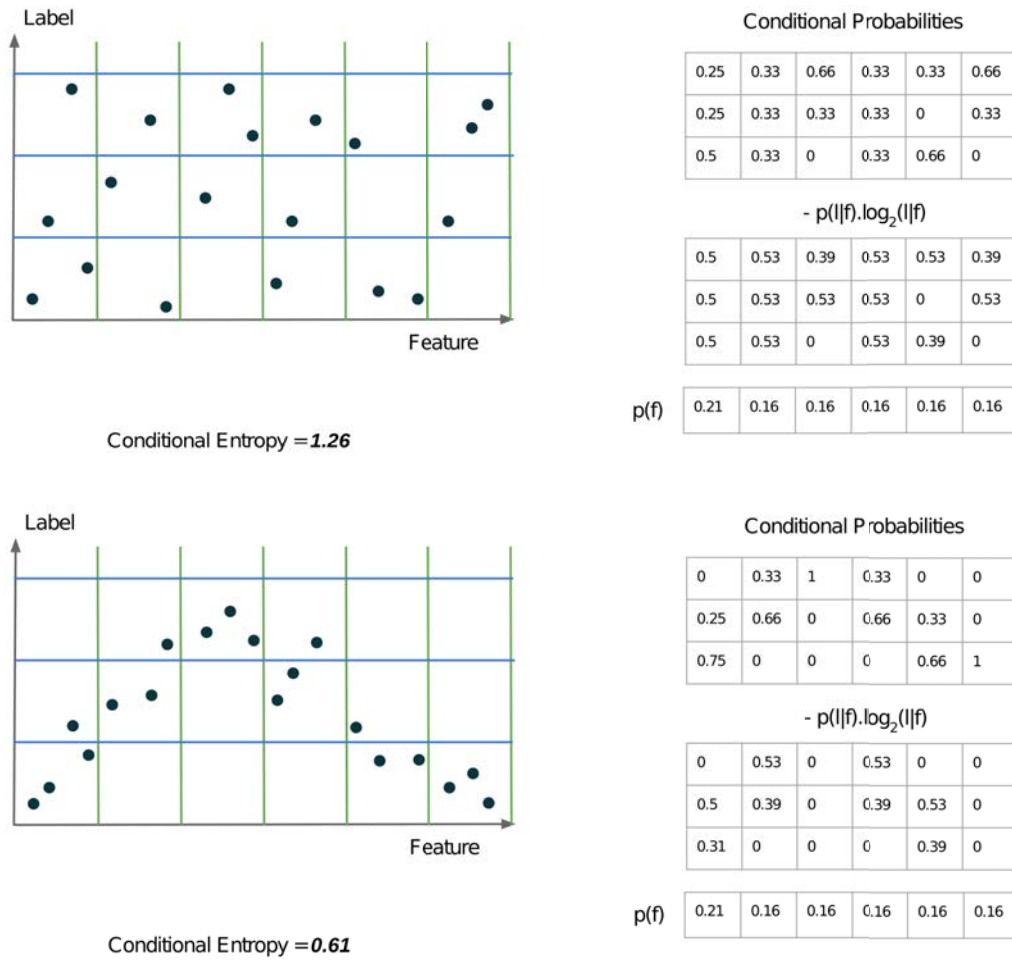


Fig. 2.11 Toy examples for understanding conditional entropy

In order to understand the meaning of conditional entropy, we represented on figure 2.11

two toy examples with different conditional entropies and the conditional probabilities corresponding to each bin. In this example, we have chosen a three bins quantization of the label and a six bins quantization of the features. We can notice that conditional entropy gives information on the possibility of predicting a label given a feature, which is particularly suited for feature selection in machine learning. We notice that the non-linear relationship between feature and label on the second plot is captured using conditional entropy, which is very small compared to the noisy configuration of the first plot (as it is a dissimilarity measure). The numbers of bins in the quantizations define the minimal scale under which non-linearities will not be captured. These hyper-parameters can be optimized in cross-validation. We use this non-linear feature selection as a first step before MLKR in order to reduce the number of features and make the training process time-acceptable. This lets us, at the same time, reduce the amount of noisy features for smoothing the optimization process.

2.4.2 Stochastic gradient descent

The computational cost of the computation of the gradient in MLKR is quadratic with respect to the number of samples. Modifications of standard gradient descent have been proposed in the literature [5]. Among them, the batch stochastic gradient descent proposes to compute the descent direction at each step by only using a random subset of samples. We decided to use this modification. In order to compare the computation time and the results, we used the standard *pumadyn-32nh* regression dataset (see subsection 2.3.4). We reduce the initial space into a five-dimensional space. In this test, each batch is composed of 60% of all training samples. In table 2.2, we present the results that we obtain on the training and testing sets for both the standard gradient descent and the stochastic gradient descent along with the computation time. We can notice that the stochastic gradient descent performs better and faster than the standard one. The improvement is due to a reduction of overfitting induced by the random selection of samples at each step of the descent.

Table 2.2 Comparison between stochastic and standard gradient descent in MLKR on the *pumadyn* regression dataset

| | training error ($\times 10^{-4}$) | test error ($\times 10^{-4}$) | time (in s) |
|------------|-------------------------------------|---------------------------------|-------------|
| standard | 3.5 | 11 | 109 |
| stochastic | 4.3 | 7.2 | 76 |

2.4.3 Lasso-regularization

The initial MLKR method minimizes the training reconstruction error with respect to the coefficients of the projection matrix \mathbf{A} . We propose to regularize the initial MLKR cost function using a Lasso penalty, meaning that we add a term to this cost function corresponding to the L_1 -norm of the matrix \mathbf{A} (which is the sum of the absolute values of its coefficients). This penalty has been proven to induce sparsity in the estimated parameter vector, reducing the risk of overfitting [96]. Some of the coefficients are shrunk all the way to zeros and corresponding solutions, with multiple values that are identically zeros, are said to be sparse. The new cost function becomes:

$$\mathcal{L}(\mathbf{A}) = \sum_{i=1}^{n_s} (\hat{y}_i - y_i)^2 + \lambda \cdot L_1(\mathbf{A}) \quad (2.5)$$

where λ controls the regularization rate. The associate gradient becomes:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{A}} = 4\mathbf{A} \sum_{i=1}^{n_s} (\hat{y}_i - y_i) \sum_{j=1}^{n_s} (\hat{y}_j - y_j) k_{ij} (\mathbf{x}_i - \mathbf{x}_j) (\mathbf{x}_i - \mathbf{x}_j)^\top + \lambda \cdot \text{sign}(\mathbf{A})$$

The regularization rate λ can be optimized in cross-validation.

We tested Lasso-MLKR with a stochastic gradient descent on the standard *pumadyn-32nh* dataset obtaining a test error of 5.1×10^{-4} compared to 7.2×10^{-4} without regularization, showing how forcing sparsity on the parameters can lead to a significant improvement. In order to have an idea of the induced sparsity, we represented on figure 2.12 both obtained parameter matrices with and without L_1 -regularization.

With the regularization, seven cells have absolute values superior to 10% of the maximal absolute value (and thus we can estimate that seven parameters will have important impact on the subspace definition). Without the regularization, we have more than 18 coefficients with absolute values superior to 10% of the maximal absolute value. The proposed Lasso regularization has been sparsifying the projection matrix, which resulted in a reduction of overfitting.

2.4.4 Multi-dimensional label extensions

In this subsection, we consider that we have a multi-dimensional label to predict, which is equivalent to having several labels to predict on the same data samples. We consider the predictions of the different labels as different 'tasks'.

As we discussed, training a model without a sufficient amount of labeled data can result in

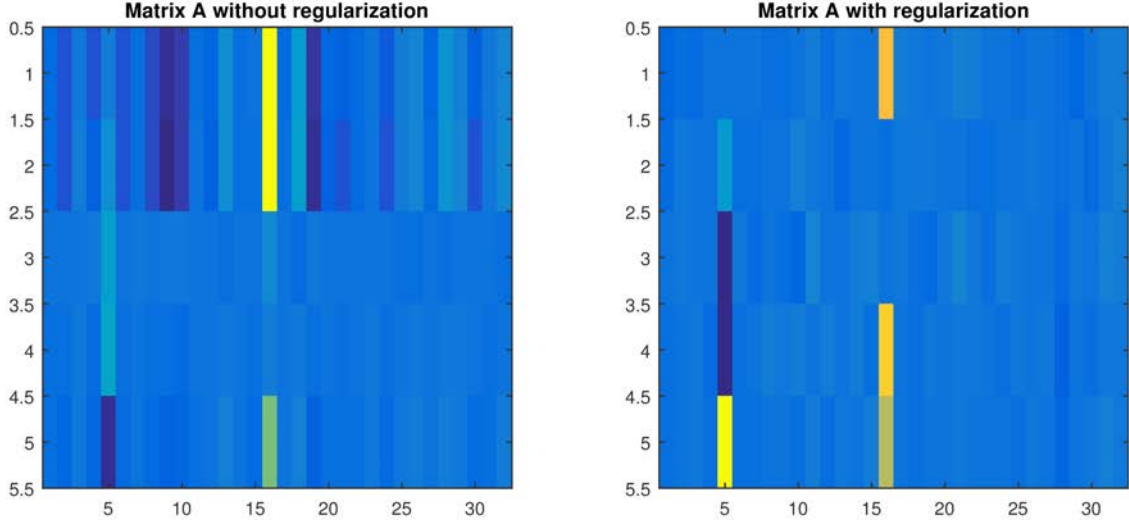


Fig. 2.12 Comparison between MLKR and Lasso-MLKR parameter matrices on the *pumadyn* dataset

overfitting issues. Reducing the number of learned parameters or including prior knowledge can reduce this effect. Thus, including the prior knowledge that the different tasks have a great probability of sharing a common representation because they are related together can result in a more efficient training. In most databases designed for face-related machine learning problems, there are several labels. For instance, an image can be labeled with different AU intensities. For simplicity, let us consider that the tasks are correlated in an important manner. In order to make a relevant use of those different labels in the databases, multi-dimensional label models can be trained. Let us consider that only one model (one projection matrix) is trained for predicting five different labels in a joint manner. The model is trained using $5.n_s$ *data-label* couples with n_s the number of samples. By training five separate models (one for each task) we only have n_s *data-label* couples for training each model. This explains why using multi-dimensional label models can virtually increase the number of information used for training the models and, as a consequence, potentially reduce overfitting.

In this subsection, we first introduce Common-Space MLKR that is a multi-dimensional label extension of MLKR. Then, we present the multi-task regularization that has been proposed in [34], before presenting an extension of MLKR that uses this regularization called Multi-Task MLKR. Finally, we introduce a new regularization called Hard Multi-Task regularization.

Common-Space MLKR

We propose to extend MLKR for multi-dimensional labels with the so-called Common-Space MLKR. In this extension, we learn one unique space in which the training samples are projected for predicting all the different labels. We recall that the matrix \mathbf{A} is the matrix used for projecting samples in the initial space on the reduced space. If \mathbf{X} is a matrix containing the data points in the initial space, data points in the reduced space become:

$$\mathbf{X}_r = \mathbf{A}\mathbf{X}$$

Let us consider T tasks. Let \mathcal{L}_t be the error corresponding to dimension t . The cost function of CS-MLKR is defined as follows:

$$\mathcal{L}_{CS} = \sum_{t=1}^T \mathcal{L}_t(\mathbf{A}) + \gamma \|\mathbf{A}\|^2$$

with $\|\cdot\|$ the Frobenius norm. To simplify the derivative formula, we introduce:

$$\mathbf{D}_t = \sum_{i=1}^{n_s} \frac{(\hat{y}_i - y_i)}{\sum_{j \neq i} k_{ij}} \sum_{j=1}^{n_s} (\hat{y}_j - y_j) k_{ij} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top$$

with task t labels and estimators. We obtain:

$$\frac{\partial \mathcal{L}_{CS}}{\partial \mathbf{A}} = 4 \sum_{t=1}^T \mathbf{A} \mathbf{D}_t + 2\gamma \mathbf{A}$$

The use of CS-MLKR corresponds to making a prior assumption of very strong correlations between the tasks. However, when the labels are correlated in a less important manner, it can be hard to capture the information that are specific to each task. We present in the next paragraph the multi-task regularization that aims at capturing both common and specific information.

Note: we chose to use an L2-norm regularization in our multi-dimensional extensions of MLKR as in the initial multi-task regularization presented in next paragraph. In our experiments (for our considered applications), we found that both L2 and L1-norm regularizations led to important parameter vector sparsification and similar results.

The multi-task regularization for SVM

In [34], Evgeniou and Pontil proposed an extension of SVM for multi-task learning. The goal is to discover the common representation between different related tasks while being able to learn their specificities. Considering T tasks, the algorithm aims at learning T classifiers $\{\mathbf{w}_t, t \in \llbracket 1; T \rrbracket\}$, one for each task. A common representation is introduced between the tasks by splitting each hyperplane into:

$$\mathbf{w}_t = \mathbf{v}_0 + \mathbf{v}_t$$

A test label y_i for task t corresponding to a feature vector \mathbf{x}_i is then classified using:

$$\hat{y}_i = \text{sign}(\mathbf{x}_i^\top (\mathbf{v}_0 + \mathbf{v}_t))$$

A penalty term is then added to the initial SVM cost function in order to encourage \mathbf{v}_0 to contain a representation that is shared by the different tasks and each vector \mathbf{v}_t to contain a specific representation for task t . The minimization problem becomes:

$$\min_{\mathbf{v}_0, \dots, \mathbf{v}_T} \sum_{t=0}^T \gamma_t \|\mathbf{v}_t\|^2 + \sum_{t=1}^T \sum_i [1 - y_i (\mathbf{v}_0 + \mathbf{v}_t)^\top \mathbf{x}_i]_+$$

with $[a]_+ = \max(a, 0)$ and γ_t controlling multi-task penalties. We propose in the next paragraph an adaptation of this regularization for MLKR.

Multi-task MLKR

This multi-task extension aims at learning T projection spaces \mathbf{A}_t , one for each task. We introduce common representation between the tasks by splitting the spaces into:

$$\mathbf{A}_t = \mathbf{B}_0 + \mathbf{B}_t$$

Let \mathcal{L}_t be the error corresponding to task t . The cost function of our MT-MLKR is defined as follows:

$$\mathcal{L}_{MT} = \sum_{t=1}^T \mathcal{L}_t(\mathbf{B}_0 + \mathbf{B}_t) + \gamma \sum_{t=0}^T \|\mathbf{B}_t\|^2$$

with $\|\cdot\|$ the Frobenius norm. To simplify the derivative formula, let

$$\mathbf{D}_t = \sum_i \frac{(\hat{y}_i - y_i)}{\sum_{j \neq i} k_{ij}} \sum_j (\hat{y}_i - y_j) k_{ij} (\mathbf{x}_i - \mathbf{x}_j) (\mathbf{x}_i - \mathbf{x}_j)^\top$$

corresponding with task t labels and estimators. We obtain:

$$\frac{\partial \mathcal{L}_{MT}}{\partial \mathbf{B}_0} = 4 \sum_{t=1}^T (\mathbf{B}_0 + \mathbf{B}_t) \mathbf{D}_t + 2\gamma \mathbf{B}_0$$

and

$$\frac{\partial \mathcal{L}_{MT}}{\partial \mathbf{B}_t} = 4(\mathbf{B}_0 + \mathbf{B}_t) \mathbf{D}_t + 2\gamma \mathbf{B}_t$$

By splitting each task parameters as a sum of common and specific ones and by adding an adequate penalty, this multi-task regularization lets us find an underlying common space and at the same time capture the task specificities.

Hard Multi-Task MLKR

We propose in this paragraph a more constrained multi-task regularization of MLKR replacing the sum by a concatenation in the subspaces split, namely H-MT-MLKR. Thus, we force a fixed number of axis to be shared by the different spaces. Let n_c be the number of common axis and n_r be the projection space dimension. The matrices \mathbf{A}_t can be defined as follows:

$$\mathbf{A}_t = \begin{bmatrix} \mathbf{B}_0 \\ \mathbf{B}_t \end{bmatrix}$$

with $\mathbf{B}_0 \in \mathcal{M}_{n_c, n_d}(\mathbb{R})$ and $\mathbf{B}_t \in \mathcal{M}_{n_r - n_c, n_d}(\mathbb{R})$

The derivatives become:

$$\frac{\partial \mathcal{L}_{HMT}}{\partial \mathbf{B}_0} = 4\mathbf{B}_0 \sum_{t=1}^T \mathbf{D}_t + 2\gamma \mathbf{B}_0$$

and

$$\frac{\partial \mathcal{L}_{HMT}}{\partial \mathbf{B}_t} = 4\mathbf{B}_t \mathbf{D}_t + 2\gamma \mathbf{B}_t$$

The proposed hard multi-task regularization lets us reduce overfitting by learning fewer parameters for same dimension projection spaces. Using MT-MLKR, the number of learned parameters is:

$$n_{par}^{MT} = n_d \cdot n_r \cdot (T + 1)$$

while using H-MT-MLKR, it is:

$$n_{par}^{HMT} = n_d \cdot (n_c + T \cdot (n_r - n_c))$$

As an example, let us consider a number of common axis $n_c = 3$ with subspace dimension $n_r = 5$ and $n_d = 80$ features in the initial space, we obtain, for 5 tasks, $n_{par}^{MT} = 2400$ and

$n_{par}^{HMT} = 1040$ parameters for MT-MLKR and H-MT-MLKR respectively. This illustrates the significance of the parameter number reduction. This extension lets us learn similar complexity predictors with less parameters, potentially resulting in overfitting reduction. We evaluate those proposed extensions on real data in chapters 3 and 4 (for landmark detection and AU intensity prediction respectively).

2.5 Conclusion

In this chapter, we introduced basic principles of machine learning. Most methods aim at learning a set of parameters for minimizing a pre-defined cost function. For instance, PCA leads to minimizing the quadratic reconstruction error of a projection in a linear orthogonal subspace. Linear Regression leads to minimizing the quadratic reconstruction error of a linear prediction function. The MLKR method leads to minimizing the quadratic error of a Nadaraya-Watson estimator in a linear subspace. We classified learning models into three different types, namely non-parametric models, parametric models and semi-parametric models. In semi-parametric methods as MLKR, the prediction step requires both the model parameters and the training samples. We believe semi-parametric models may lead to an interesting compromise between the complexity of the functions that can be learned and the rate of potential overfitting.

We analyzed several characteristics of the MLKR method using synthetic data. First, we discussed issues related to its non-convexity. Then, we discussed both its time and memory complexities. Finally, we discussed its power of generalization (robustness to overfitting and extrapolation capabilities of its prediction function). This analysis let us identify the advantages and drawbacks of the MLKR method.

We introduced different extensions of MLKR, designed for reducing the effects of its drawbacks. First, we proposed to use a non-linear feature selection step based on conditional entropy. Then, we proposed to perform a stochastic gradient descent that lets us increase performance while reducing the computation time.

Face-related prediction tasks can be particularly complex and the data required for training the models is hard to collect and label. When a large number of parameters have to be learned with limited data, the probability of overfitting rises. For that reason, we proposed different multi-dimensional label extensions of the MLKR method focusing on overfitting reduction. Our three proposed extensions are Common-Space MLKR (CS-MLKR), Multi-Task MLKR (MT-MLKR) and Hard Multi-Task-MLKR (H-MT-MLKR). The CS-MLKR extension lets us learn one unique space for predicting all labels at the same time. The MT-MLKR extension makes use of the multi-task extension initially proposed for SVM and lets us learn a common

representation between the labels while capturing their specificities at the same time. The H-MT-MLK extension, based on our proposed hard multi-task regularization, lets us models that have similar properties to those obtained by MT-MLKR but requires a reduced number of parameters to optimize. We believe this reduction of the degrees of freedom of the model may result in overfitting reduction.

In chapters 3 and 4 of this dissertation, we present two complete frameworks for facial landmark detection and AU intensity prediction that all both based on our proposed Hard Multi-Task-MLKR extension.

Chapter 3

Facial landmark detection

3.1 Introduction

Landmark detection is one of the most important steps in many face-involving machine learning systems. As an example, it is a key step in automatic face recognition, whose applications exist in numerous domains as for automatic tagging of pictures in social networks or for airport security. It is also of primary importance for automatic mental state prediction. For all those applications, a precise tracking of facial landmarks is an extremely valuable information, that highly impacts the performance of a wide range of the prediction systems. As we noticed in chapter 1, most landmark detection methods are iterative ones. A mean shape is first initialized in the image, and then refined step by step to converge towards the true facial shape. We presented in chapter 1 different kinds of iterative landmark detection methods. Cascaded-regression methods have recently emerged and let to impressive results. At each step of those methods, a regressor is learned. The input of each regressor is a set of features characterizing the appearance of the face around the set of previously located landmarks. The outputs are the displacements of the landmarks. Among those methods, Supervised Descent Method (SDM [117]) has been rapidly and widely used for its impressive results. In this method, the regressor that is used at each iteration is a linear regressor.

As it is tedious to gather various face images and precisely label them with an important number of fiducial points, datasets often contain a limited number of images (for instance, for the 300W challenge, the participants had access to around 3,000 images for learning). Moreover, as in many face-related machine learning systems, there is a need for extracting a high number of features in order to characterize extensively the facial appearance. As we discussed in chapter 2, in this context, linear regression may be prone to overfitting. Linear regression is a parametric model whose number of parameters is equal to the number of features. In order to avoid overfitting, a Principal Component Analysis (PCA) is performed

in [117]. However, features resulting from PCA projections contain global information. At the last iterations of a landmark detection method, the points are close to their true location and require small refinements. Those steps are very local steps. Features characterizing locally the appearance around each landmark may contain particularly relevant information for those steps. Our proposed method lets us make use of local features.

In chapter 2, we proposed several extensions of the Metric Learning for Regression (MLKR) method. In MLKR, the regression is performed in a non-linear manner. We believe that this non-linearity of the prediction is relevant because the relationship between appearance features and landmark displacements has no reason to be linear. Moreover, some proposed extensions (CS-MLKR and H-MT-MLKR) let us reduce overfitting by making the prior assumption that some features can be useful for predicting different labels. During the first iterations of a landmark detection method alignment process, when the task is to roughly estimate the shape using features extracted on a far from ground truth model, we believe some features can be relevant for predicting several landmark displacements. For instance, if a landmark supposedly being the center of the inferior lip appears to have a like chin appearance, it is probably relevant to pull up the set of landmarks defining the whole inferior lip. We decided to define five groups of landmarks (gathering the points defining the mouth together, those defining the nose, the eyes, the eyebrows and the face contour). Our framework lets us learn five different spaces that are each used for predicting all the landmarks within a group. Each space is defined as the concatenation of a common space and a specific space in order to include a constraint over the locations of the groups relative to each others. Our framework uses our H-MT-MLKR regression method (details are given in section 3.4.2).

We present and discuss some of the most commonly used features in facial analysis machine learning systems in section 3.2. In section 3.3, we present the 300W challenge database, that is used for evaluating our system. We detail our framework in section 3.4. The obtained results and the comparison to state-of-the-art methods are presented in section 3.5 before concluding in section 3.6.

3.2 Commonly used appearance features

In face-related machine learning systems, different methods have been proposed for describing the appearance of an image patch. A direct use of raw data (corresponding to the intensity values of the pixels defining the patch) is not always the most relevant choice depending on the application. Appearance features are transformations of raw data into different information, to should be easier to use or contain relevant information for a specific task.

In this section, we detail three commonly used methods for face-related machine learning systems, namely Histograms of Oriented Gradients (HOG), Local Binary Patterns (LBP) and Local Gabor Binary Patterns (LGPB). Then, we discuss some advantages and drawbacks of those methods as well as the importance of histogram normalization.

Note: many other methods have been proposed in the literature (e.g. Haar-like features or Local Phase Quantization). This section aims at giving a brief review of different kinds of commonly used features in order to explain and highlight the feature choice we made for our landmark detection framework as well as for our AU prediction system presented in next chapter.

Histograms of Oriented Gradients (HOG)

HOG aim at characterizing the appearance of a patch by describing the contours of its inside shapes and their orientations (e.g. horizontal, vertical, diagonal). They were proposed by Dalal and Triggs [22] and have been widely used since. For computing the oriented gradients of an image, a convolution with different filters is performed. For instance, the filters presented in figure 3.1 may be used in order to calculate the derivatives of an image in four different directions. We can notice here that the third filter highlights horizontal contours.



Fig. 3.1 Convolution of an image by four oriented gradient filters

In order to describe the information contained in the obtained oriented gradient maps, histograms are used. For each map, the sum of the intensities of positive valued pixels is computed. It is then stored in the corresponding histogram bin. For instance, for computing a eight orientations HOG, the four previously presented filters can be used to obtain four maps. Those maps are sufficient to compute the eight bins histogram (by making use of

the symmetry in a regular eight directions division). More directions can be used for HOG computation using other sets of filters.

Those HOG globally describe an image patch by characterizing its gradients. In figure 3.2, we represented three patches and their corresponding HOG histogram. We can notice that the three different images are here described by feature vectors of distinctive shapes.

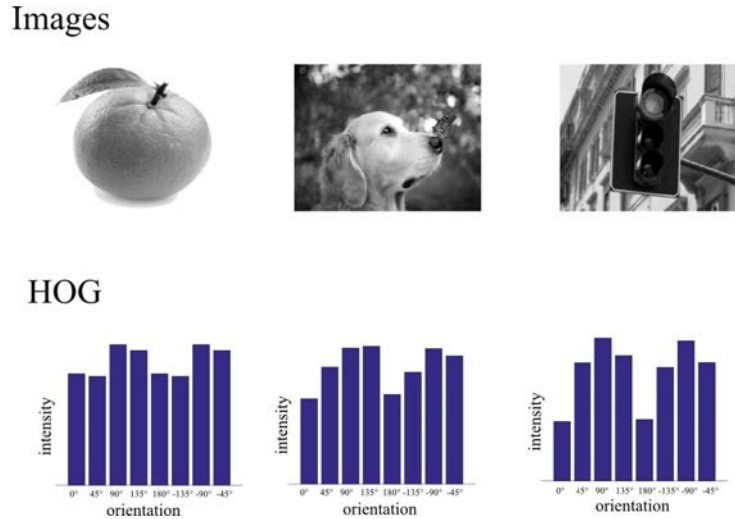


Fig. 3.2 Examples of images and corresponding HOG histogram

Local Binary Patterns (LBP)

LBP are another method for describing the appearance of a patch. They have been proposed by Zhao and Pietikainen [129]. Pixels are characterized one by one relative to their neighbors. They are classified according to the following method. Let us consider the array of eight neighbors surrounding a pixel (e.g. beginning at the top right corner, turning clockwise, and ending at the top). Whether those pixel values are higher or lower than the central pixel value, a one or a zero is stored in a cell, building an eight elements binary array. Each pixel is associated to a class whose number is the $base_{10}$ conversion of this binary array. On the contrary to HOG, that contain information about gradient intensities, LBP histogram only contains the number of pixels belonging to the different classes. This information makes LBP invariant to lighting changes. There are $2^8 - 1$ different pixel classes. Then, LBP histogram is of size 255.

We illustrate the computation of the class number in figure 3.3. Examples of patches and their corresponding LBP histogram are represented in figure 3.4.

LBP are particularly efficient for characterizing a patch of sufficient size. However, for small

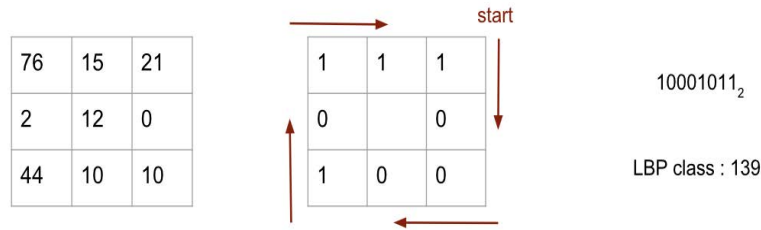


Fig. 3.3 Illustration of LBP class computation

Images



LBP

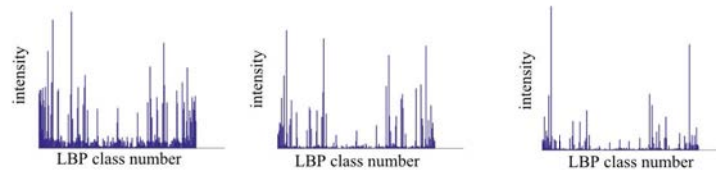


Fig. 3.4 Examples of images and corresponding LBP histograms

patches, LBP histograms can be very sparse and this information has to be taken into account when designing prediction systems.

Local Gabor Binary Patterns (LGBP)

LBP describe each pixel at a very small scale (using a 3x3 patch centered on the pixel). In order to add information at larger scales and stress the differences of gradient orientations, Zhang et al. [126] proposed LGBP. The idea is to use LBP for describing maps resulting of the convolution of an image by a set of Gabor filters, that are used for highlighting oriented gradients at different scales. Figure 3.5 illustrates a bank of Gabor filters along with the results of the convolution of an image with that filter bank. The dimension of an LGBP histogram can be high. Using the previously presented filter bank (with seven orientations and three different scales, we obtain $21 \times 3 \times 255 = 16,065$ features for one patch). With an important number of patches having to be characterized, LGBP leads to high dimensionality feature spaces.

There are several differences between the three kinds of features we previously introduced.

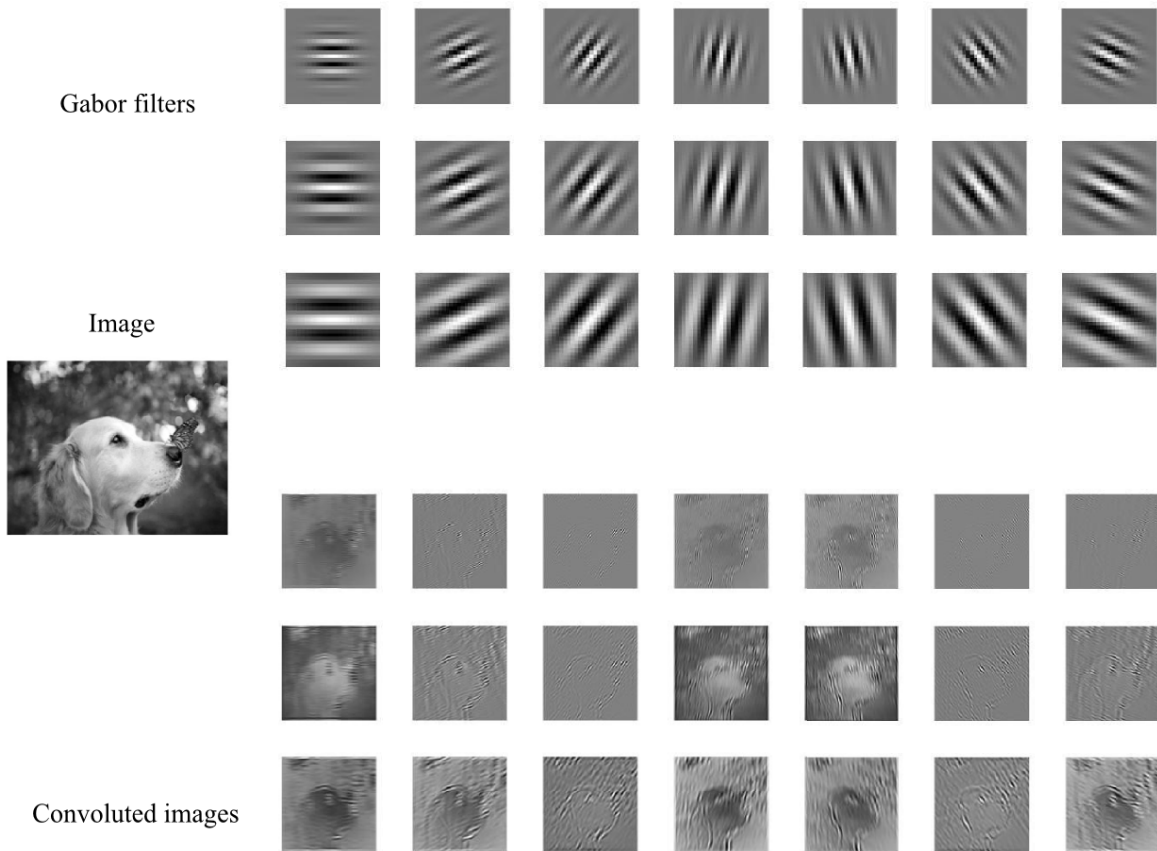


Fig. 3.5 Convolution of an image by a Gabor filter bank

They describe the appearance of patches more or less precisely. They are said to be more or less discriminant because they can make a difference between more or less similar patches. LBP and LGBP are more discriminant than HOG features. On the other hand, their dimensions are higher as well as their computation costs. According to the number of available training samples, the discriminant power of features has to be carefully chosen. For landmark detection, the computation cost is highly important because the task often runs in background and is combined with other tasks in a system. We decided to use HOG features as in [117] because of the very high performance reached by their method as well their low computation cost.

About histogram normalization

When using HOG, histogram normalization is of primary importance. Let us consider that we want to describe images using HOG histograms. Let us also consider that there are large luminosity variations in the database. As HOG are computed by summing gradient

intensities, two identical images with different dynamics will be described as two completely different feature vectors. As an example, if the task is object classification, for example, this description will not be optimal. In that case, it can be useful to normalize the histograms. There exist different ways of normalizing a histogram. It is possible to divide by the maximum of the different bins or simply scale the norm to the value one. The obtained histogram describes the amount of gradients in each direction relative to the total amount of gradients. This leads to classifying objects without the disturbance of a luminosity change.

Normalizations can be performed using more or less local areas. First, a simple normalization can be performed using the same patch for normalizing and for computing the histogram. We illustrate this normalization on the left column of figure 3.6 in which the considered patch (surrounded by a blue line) is also the one used for normalizing (filled in blue). Notations are the followings: $\{b_{i,j}, j \in \llbracket 1; n_{or} \rrbracket\}$ is the initial histogram (i being the index of the considered patch and n_{or} the number of different orientations), $\{\hat{b}_{i,j}, j \in \llbracket 1; n_{or} \rrbracket\}$ is the normalized histogram.

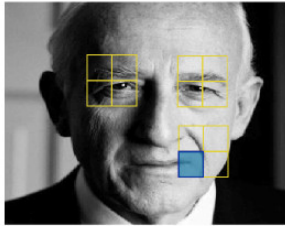
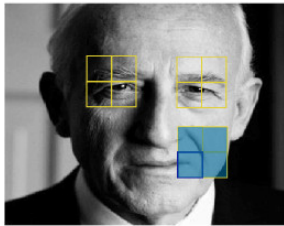
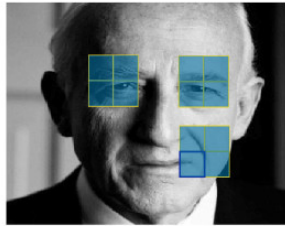
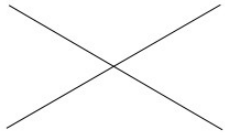
| | Unitary | Local | Global |
|------------------|---|--|---|
| Set |  same patch |  $\mathcal{J} = \text{neighbors}$ |  $\mathcal{J} = \text{all patches}$ |
| same orientation |  | $\hat{b}_{i,j} = \frac{b_{i,j}}{\sqrt{\sum_{l \in \mathcal{J}} b_{l,j}^2}}$ | $\hat{b}_{i,j} = \frac{b_{i,j}}{\sqrt{\sum_{l \in \mathcal{J}} b_{l,j}^2}}$ |
| all orientations | $\hat{b}_{i,j} = \frac{b_{i,j}}{\sqrt{\sum_{k=1}^{n_{or}} b_{i,k}^2}}$ | $\hat{b}_{i,j} = \frac{b_{i,j}}{\sqrt{\sum_{l \in \mathcal{J}} \sum_{k=1}^{n_{or}} b_{l,k}^2}}$ | $\hat{b}_{i,j} = \frac{b_{i,j}}{\sqrt{\sum_{l \in \mathcal{J}} \sum_{k=1}^{n_{or}} b_{l,k}^2}}$ |

Fig. 3.6 Illustration of different histogram normalizations

The previously described normalization characterizes the amount of gradients oriented to a specific direction relative to the total amount of gradients in the patch. Another commonly

used normalization is the one used in SIFT points description [57]. That normalization characterizes the amount of gradients oriented to a specific direction relative to the amount of gradients but in neighboring areas. When using a large patch divided into 2 by 2 small patches for describing an area, this normalization brings out the gradients of the small patches relative to the large patch gradients. Two variants can be computed, using same orientation bins for normalizing or using all orientations. We illustrate those local normalizations on the central column of figure 3.6 with the considered patch surrounded by a blue line and the four filled patches around it used for normalizing. Let us consider a very local task, for instance during the last iterations of a face alignment process. We can assume at those iterations that the landmarks have already been roughly located. Let us assume that a mouth corner is at a few pixels of its ground truth. Using a 2 by 2 patch centered on the current location of the landmark, the previously presented normalizations can bring out the contour of the mouth relative to the smoother appearance of the cheeks and lips. This information could be very useful for a precise refinement of the mouth corner.

A third normalization uses patches from all over the image in order to normalize the histograms. This way, the extracted information is relative to an estimation of the gradients of the whole facial area. Both orientation variants can be computed for a global normalization. We illustrate this normalization on the right column of figure 3.6 with the considered patch surrounded by a blue line and the normalization being performed using patches from all over the image (filled in blue).

We evaluate two different normalizations (both local and global) for our landmark prediction system in section 3.5. In next section, we detail the 300W challenge database that is used in our evaluations.

3.3 The 300W database

Databases designed for learning landmark detection systems contain images of faces with labeled landmarks. A set of points defining the contours of the different elements of the face are located in the images (abscissa and ordinate of the points). Several characteristics are important and make differences between databases. First there is the question of the number of labeled landmarks. The more landmarks there are, the more precise description of the faces movements can be analyzed afterwards. Some databases are only labeled with 5 points (centers of the eyes, top of the nose and corners of the mouth). Other databases are labeled with more than a hundred points (describing precisely the different contours). With the recent improvements of landmark detection, widely-used detectors currently predict around 70 landmarks. Another important characteristic is the variability of the data. The number of

different subjects and their differences of age, facial morphology and skin color is important. Various head poses and image qualities (resolution, noise, occlusions) are also needed for learning systems designed for an in-the-wild use. For a long time, the BioID database has been extensively used. It is labeled with 20 landmarks and contain near-frontal images. The evaluations presented in this chapter are performed on the 300W database. This database is created from existing datasets, including LFPW [4], AFW [132], Helen [52], XM2VTS¹ and the IBUG dataset. It contains around 4,000 images labeled with 68 landmarks. The IBUG dataset is extremely challenging as its images have large variations in face poses, expressions and illuminations. On figure 3.7, we show four labeled images extracted from AFW, Helen, LFPW and IBUG included in the 300W challenge database.



Fig. 3.7 Images and landmarks extracted from the 300W challenge database

3.4 Our facial landmark prediction framework

Our framework is based on an iterative regression method. First, a mean shape is initialized in the image and then refined step after step. At each step, appearance features are extracted using the information of the appearance around the current landmark locations and the point displacements are predicted in a joint manner using our proposed Hard Multi-Task MLKR method. First, we present our feature extraction process. Then, we detail our regression method. Finally, we present our experimental setup.

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

3.4.1 Feature extraction

As a first step, a Viola-Jones face detection is performed. Then, we rescale the image into a 180 by 180 pixels image. Afterwards, we extract HOG features on 2 by 2 patches centered on each of the 68 landmarks. In our experiments, we evaluate both local and global normalizations of those features using the variants that use all orientations for normalizing. The patch sizes define the scale of each landmark appearance characterization. In iterative regression methods, landmark estimations are more and more close to the ground truth. Thus, we decided to reduce the patch sizes at each step. We perform 10 steps with the following HOG sizes for our patches: $s = [32, 24, 20, 16, 12, 8, 4, 4, 4, 4]$. In figure 3.8, we illustrate the HOG feature extraction process for a patch of size s (corresponding here to the first step size).

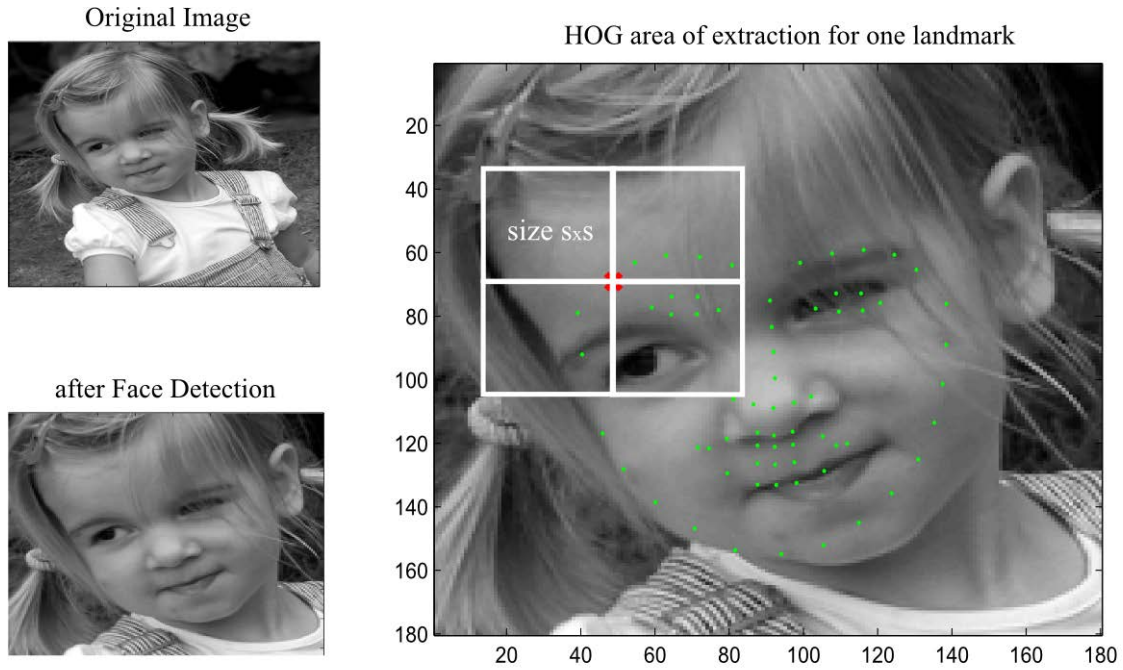


Fig. 3.8 Illustration of our landmark detection framework HOG extraction process on a 2 by 2 patch

We obtain 4352 features (two normalizations, four areas for each landmark, 68 landmarks, eight orientations for the gradient computation). In next subsection, we detail the implementation of our H-MT-MLKR regression method.

3.4.2 Proposed regression method

In our landmark prediction framework, the regression step is based on our proposed Hard Multi-Task MLKR. We define five different groups of landmarks as in figure 3.9. The set of predictions in each group is considered in our method as one task.

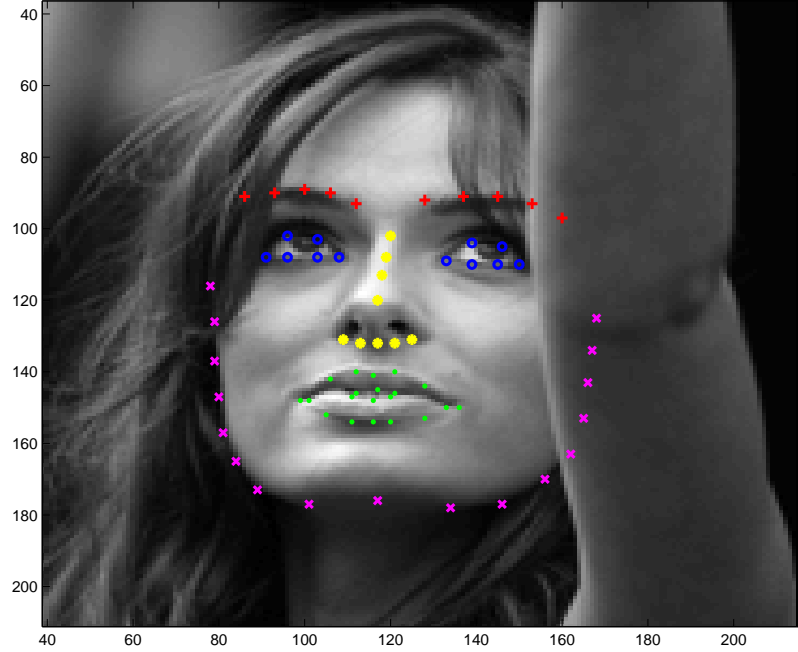


Fig. 3.9 Definition of the five different point sets

We aim at learning five spaces (one for each group of landmarks), that are each defined as the concatenation of common axis (shared by the five spaces) and specific axis. Imposing the landmarks within a group to be predicted all together using the same space (and thus the same set of features), lets us force a strong shape constraint within each group. Indeed, the estimated displacements of each group at each step are weighted means of training displacements (using the Nadaraya-Watson estimator). This strong shape constraint reduces the possibility of unrealistic estimated shapes of each group. The groups have been defined by gathering closely located landmarks in order for a group to potentially share several relevant common features. We gathered frequently occurring symmetric movements (e.g. both eyes). We gathered shape modifications induced by head rotation (e.g. nose or limbs). We separated frequently independent movements (e.g. eyebrows from mouth). Most iterative regression methods include a global shape constraint in an implicit manner by predicting all landmarks at the same time and with the same set of features. Using one space for each

landmark group without enforcing common axis between them would not let us force the groups relative to each other and would not prevent the unrealistic location of a group with respect to another. By imposing at each step the five spaces to share common axis, we aim at keeping the global consistency of the groups relative to each other.

Let $\{\mathbf{A}_t, t \in \llbracket 1; 5 \rrbracket\}$ be the five projection matrices, defined as follows:

$$\mathbf{A}_t = \begin{bmatrix} \mathbf{B}_0 \\ \mathbf{B}_t \end{bmatrix}$$

with \mathbf{B}_0 the common part and \mathbf{B}_t the specific parts. Let \mathcal{L}_t be the error associated to landmark group t . Let $\{\mathcal{S}_t, t \in \llbracket 1; 5 \rrbracket\}$ be the sets of labels that have to be predicted in each group (e.g. ordinate and abscissa of the landmarks defining the groups). We have, for n_s training samples:

$$\mathcal{L}_t(\mathbf{A}_t) = \sum_{i \in \mathcal{S}_t} \sum_{j=1}^{n_s} (\hat{y}_{i,j} - y_{i,j})^2$$

with the estimation \hat{y} being computed in the space defined by \mathbf{A}_t (see chapter 2 for more details about H-MT-MLKR). The global cost error of our model can be written as follows:

$$\mathcal{L}_{HMT} = \sum_{t=1}^5 \mathcal{L}_t(\mathbf{A}_t) + \gamma \sum_{t=0}^5 \|\mathbf{B}_t\|^2$$

with $\|\cdot\|$ the Frobenius norm.

We select features using conditional entropy (see chapter 2 for more details). There are several ways of selecting features using conditional entropy for multi-dimensional label problems. A fixed number of features could be selected for each of the labels (for each of the 68 by 2 landmark coordinates). However, in our method, each learned space is used for predicting a whole set of labels. Thus, we characterize the relevance of each feature for predicting the different global sets of labels. For doing that, we sum the conditional entropies. For each feature and each of the five tasks, we compute:

$$H(\mathcal{S}|f) = \sum_{i \in \mathcal{S}} H(l_i|f)$$

Afterwards, we rank the features according to H and select a fixed number of features for each of the five groups.

3.4.3 Experimental setup

Our experiments have been performed on the 300W challenge dataset following the training and testing division used in [80]. The training set consists of AFW, the training parts of LFPW and Helen, with 3148 images in total. The testing set consists of IBUG, the testing parts of LFPW and Helen, with 689 images in total. As in [80], the images from XM2VTS have not been used. Each learned subspace is of dimension 12 (6 common axis and 6 specific axis). We selected 500 features at each step (100 for each task). In order to induce more variability during training, as in [117], some deformed mean shapes are computed. We computed two deformed shapes for each image using a random rotation between -15° and 15° plus a random translation from -10 to 10 pixels and a random rescaling by a factor between 0.9 and 1.1. Those shapes are denoted $MShape_{:,1}$ in algorithm 1. Using those shapes, we extract both features (HOG histograms) and labels (displacements towards ground truth). The obtained data $F_{:,1}$ is used for learning the spaces. In order to reduce overfitting, we decided to perform the predictions using other shape initializations than those used for learning our spaces. We compute four shapes for each image ($MShape_{:,2}$) for obtaining data $F_{:,2}$ that is embedded in our spaces for predicting the displacements. The training process is detailed in algorithm 1. We begin by learning projection spaces for step one using data points (features and labels) extracted on a set of deformed mean shapes (line 1 to line 3). Then, we extract new data points on another set of mean shapes that will be used (combined with the learned spaces) for predicting the displacements of the landmarks during first step (line 4 to line 5). Then, for each step s superior to one, we perform the regression until step $s - 1$ (line 7 to line 10). We extract data (features and labels) on corresponding locations and we learn the corresponding projection spaces for step s (line 11 to 12). Then, in order to perform the regression for step s using other data points than those used for training the spaces, we initialize new shapes, perform the regression until step $s - 1$ and extract data points on the obtained locations (line 13 to line 17). This framework combines new shape initializations for learning each step parameters and other data points than those used during training for performing the regressions which aims at reducing overfitting as much as possible. Our model parameters are learned in a way that tends to be similar to the test conditions.

3.5 Results on the 300W dataset

In this section, we evaluate the different elements of our landmark detection framework and compare it to state-of-the-art methods on the 300W dataset. First, we evaluate the impact of the different HOG normalizations on the prediction results. Second, we compare our H-MT-MLKR method to a CS-MLKR method for evaluating the impact of the constraint

Algorithm 1 Training algorithm for our landmark detection framework

```

1: initialize deformed mean shapes:  $MShape_{1,1}$ 
2: extract data:  $F_{1,1}$ 
3: learn H-MT-MLKR spaces for step 1 using  $F_{1,1}$ :  $Spaces_1$ 
4: initialize deformed mean shapes:  $MShape_{1,2}$ 
5: extract data:  $F_{1,2}$ 
6: for step  $s=2$  to 10 do
7:   initialize deformed mean shapes  $MShape_{s,1}$ 
8:   for  $i=1$  to  $(s-1)$  do
9:     perform regression using  $F_{i,2}$  and  $Spaces_i$ 
10:  end for
11:  extract data  $F_{s,1}$ 
12:  learn H-MT-MLKR spaces for step  $s$  using  $F_{s,1}$ :  $Spaces_s$ 
13:  initialize deformed mean shapes:  $MShape_{s,2}$ 
14:  for  $i=1$  to  $(s-1)$  do
15:    perform regression using  $F_{i,2}$  and  $Spaces_i$ 
16:  end for
17:  extract data  $F_{s,2}$ 
18: end for

```

brought by the inclusion of shared axis. Then, we show how semi-parametric methods can be advantageous for overfitting reduction by evaluating the impact of the embedding of new training data points in the learned subspaces for performing the predictions. Afterwards, we compare our method to the initial approach proposed in [117] that uses a combination of PCA and Linear Regression. Finally, we compare to recent state-of-the-art methods.

3.5.1 HOG normalizations

In section 4.3.1, we discussed different feature normalizations. We computed features using both local and global normalizations. In order to evaluate the relevance of each normalization, we learned two H-MT-MLKR landmark prediction methods with both obtained feature sets. We present the results in figure 3.10 where the normalized mean error is plotted with respect to the iteration steps. The error corresponds to a mean error of all landmarks normalized by the interocular distance (defined in the 300W challenge as the distance between exterior corners of the eyes, that we denoted p^{re} and p^{le}). It is defined as follows:

$$e_n = \frac{\frac{1}{n_p} \sum_{i=1}^{n_p} \sqrt{(\hat{p}_x^i - p_x^i)^2 + (\hat{p}_y^i - p_y^i)^2}}{\sqrt{(p_x^{re} - p_x^{le})^2 + (p_y^{re} - p_y^{le})^2}}$$

with n_p the number of landmarks.

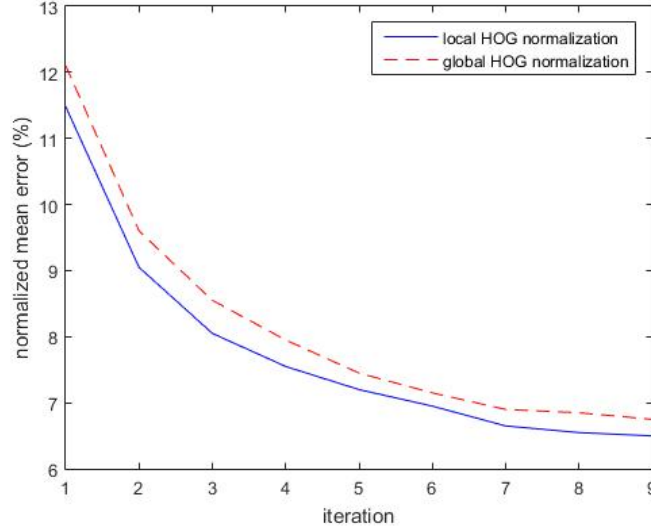


Fig. 3.10 Comparison between local and global HOG normalizations for landmark prediction on 300W dataset

We can notice that the local normalization performs better than the global normalization from the first iteration. The first step of the alignment process aims at roughly locating the facial shape. Its role is to estimate if the initial mean shape was too much left, or too much down for instance. A few landmarks of the initial mean model are often located outside the face, in the background. Thus, depending on the gradients of the background, a global normalization could lead to different description for the description of the same appearance patch around the same estimated landmark. This could explain why the local normalization outperforms the global one from the first step. We can notice that the normalized error for the local normalization at the third step is 8.0% compared to 8.6% for the global one.

3.5.2 Comparison to CS-MLKR

In this subsection, we evaluate the relevance of our H-MT-MLKR approach compared to a CS-MLKR approach (both systems are learned with local normalizations). When using CS-MLKR, we learn five independent spaces in which the different groups of landmarks are predicted. In H-MT-MLKR, the spaces share half of their axis. We present the obtained results in figure 3.11 where the normalized mean error is plotted with respect to the iterations. We can notice that H-MT-MLKR greatly outperforms CS-MLKR at every step of the alignment process. This is explained by the lack of geometric constraints between the different groups

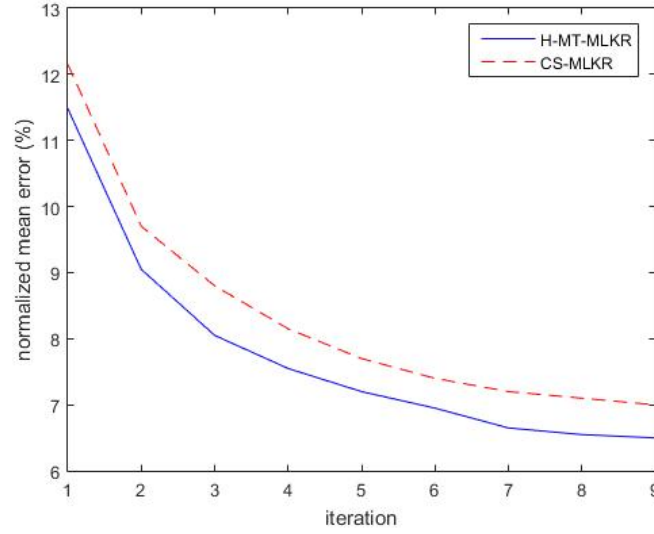


Fig. 3.11 Comparison between Hard MT-MLKR and CS-MLKR for landmark prediction on the 300W dataset

of landmarks when using the CS-MLKR approach. We can notice that the normalized error for H-MT-MLKR at the third step is 8.0% compared to 8.8% for CS-MLKR.

3.5.3 Embedding more training data samples

A limiting factor of kernel methods is the number of samples that can be used for training because of the quadratic memory complexity of kernel computation. One of the advantages of semi-parametric method as MLKR is that more data samples can be embedded in the learned spaces for performing the Nadaraya-Watson predictions, which can lead to higher performance. In this subsection, we evaluate the impact of the number of shape initializations used for computing the data samples that are used for the regressions. The different Nadaraya-Watson predictions are performed using two or four deformed shapes for each training image. The number of samples used for learning the spaces stays the same for both tests (computed using two deformed shapes per image). On figure 3.12, we present the obtained results.

We can notice that embedding more learning samples for the regressions lets to an improvement of the results. The normalized error using four deformed shapes per image at the third step is 8.0% compared to 8.2% only using two deformed shapes. However, the more images are embedded, the longer the Nadaraya-Watson prediction takes at each step. Using four deformed shapes per image, our system works at 15 frames per second (Matlab implementation on an Intel Core i7-3770 at 3.4 GHz). This frame rate makes the current

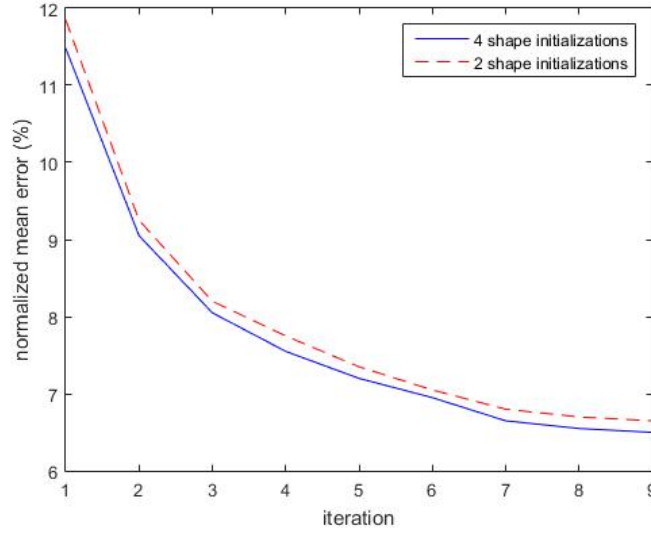


Fig. 3.12 Impact of the number of shape initializations used for performing the Nadaraya-Watson regressions

system difficult to use in real-time applications. We discuss a few ideas for reducing the computation time of our landmark detection system in chapter 5.

3.5.4 Comparison to global PCA and Linear Regression

In this subsection, we compare our H-MT-MLKR method to the initial method of [117] combining a PCA and a Linear Regression. We present the results on figure 3.13.

We can notice how our proposed regression outperforms PCA+LR at every step of the process. This can be explained by the fact that our method lets us use more local features. The first iterations being very global alignments, similar performances are obtained by PCA+LR and by our method. The more local refinements of the latter steps are better predicted using local features. We can notice that the normalized error for H-MT-MLKR at the third step is 8.0% compared to 9.0% for PCA+LR.

3.5.5 Comparison to state-of-the-art methods

In this subsection, we compare our landmark detection framework to state-of-the-art methods on the 300W dataset. In table 3.1, we compare our results to Explicit Shape Regression (ESR [6]), Supervised Descent Method (SDM [117]) and Local Binary Features (LBF [80]). We can notice that our method outperforms both SDM and ESR methods on the 300W dataset and performs similarly to LBF which also makes use of local features. We represented

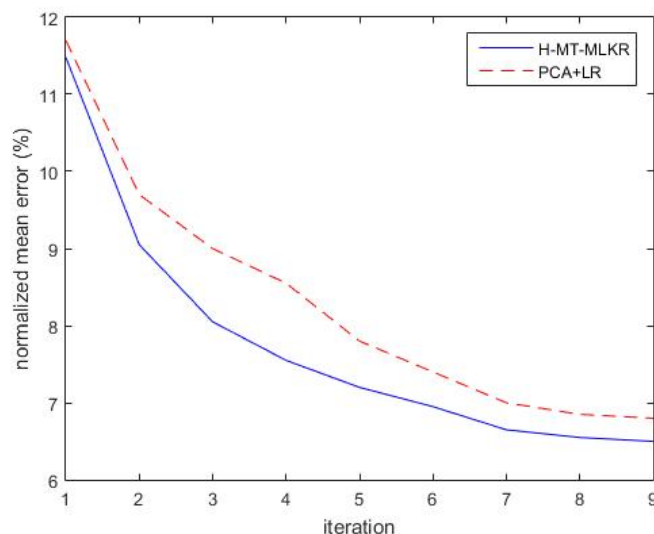


Fig. 3.13 Comparison between H-MT-MLKR and PCA+LR for landmark prediction on the 300W dataset

Table 3.1 Comparison between H-MT-MLKR and three state-of-art methods on the 300W dataset

| Method | Mean normalized error (%) |
|------------------------------|---------------------------|
| ESR [6] (reported in [80]) | 7.58 |
| SDM [117] (reported in [80]) | 7.52 |
| LBF [80] | 6.32 |
| H-MT-MLKR | 6.50 |

the cumulative error distribution of our system along with a few examples of images with corresponding normalized errors on figure 3.14.

We can notice that landmarks of images with common facial expressions and no self occlusions are very precisely predicted (as the frontal example with the 3.2% normalized error or the mid-angle head-pose example with the 6% normalized error). Error levels under 6% are obtained for 65% of the images. We can notice that we obtain a 8% normalized error for an image with self occlusions caused by a beard that led the contour of the face to be predicted less accurately. Error levels under 8% are obtained for more that 80% of the images. We obtain a normalized error of 15.3% for an image with highly asymmetric self occlusions and facial expressions. However, we can notice that the global locations of the different facial parts are still well localized with this error rate. More than 96% of the images are predicted under this error level.

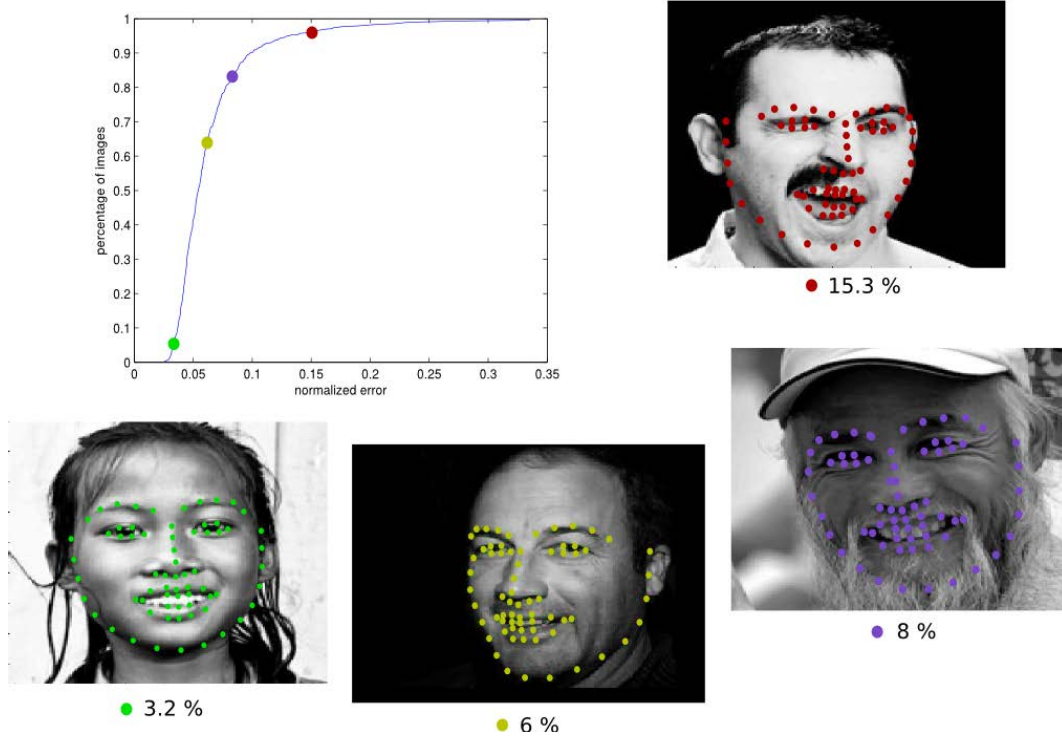


Fig. 3.14 Cumulative error distribution and examples of images and located landmarks with corresponding errors

3.6 Conclusion

In this chapter, we presented a complete framework for landmark detection. Our method is an iterative regression method based on our proposed H-MT-MLKR method. We define five groups of landmarks and learn five subspaces for predicting the landmark displacements at each step. Each subspace is used for predicting a whole set of landmarks, which lets us keep an implicit strong constraint within the different groups. In order to keep a global shape consistency, we force a set of axis to be shared by the different spaces. This lets us use local features while enforcing a global shape constraint.

In this chapter, we also discussed commonly used features as well as the importance of feature normalization (that can be performed on more or less global areas surrounding the feature extraction patch). We discussed how the use of a semi-parametric regression method can help training without overfitting by embedding new training data points that have not been used for learning the subspaces.

We evaluated our framework that combines a non-linear feature selection and a multi-task regression on the challenging 300W dataset (obtaining a 6.50% mean normalized error) and compared very favorably to the initial method of [117] composed of a PCA plus a Linear

Regression (7.52% mean normalized error) and to Explicit Shape Regression [6] (7.58% mean normalized error). We also compared our method to the very recent approach of Ren et al. [80] (6.32% mean normalized error) that also makes use of local features.

The fact that a particular focus on overfitting reduction seems to increase landmark detection systems performance raises the question of the necessary amount of data for designing a precise in-the-wild system. We discuss an idea about database design in chapter 5. In next chapter, we present our facial Action Unit prediction framework.

Chapter 4

Action Unit prediction

4.1 Introduction

The Facial Action Coding System (FACS) is a system used for describing facial expressions. Action Units (AU) characterize activations of facial muscles and aim at coding in a unique way every facial expression potentially reachable by humans. The FACS has been widely used because it is compact and leads to a precise description of facial expressions (see chapter 1). Being able to predict AU in an automatic manner is of primary importance for numerous applications (e.g. for letting robots gather high-level information for interacting with humans in a natural manner).

In order to learn machine learning systems for predicting AU, labeled data is needed. Designing AU databases is complex for several reasons. First, because collecting useful data is a hard task. Databases need to contain numerous and various facial expressions for leading to precise systems. Second, labeling AU is time-consuming and difficult. Most people cannot precisely perceive which facial muscles are activated or not in a facial expression. Thus, the labeling requires experts that are specifically trained for the task. For several years, AU prediction has been seen as a classification problem because of those difficulties in AU database design. A facial expression was then described as a set of activated muscles. Along with the need for systems to predict AU in a more precise manner, recent databases contain the intensities of each AU. The problem can then be viewed as a multi-dimensional label regression task. We present in this chapter the application of our proposed Hard Multi-Task Metric Learning for Kernel Regression (H-MT-MLKR) method to AU intensity prediction. Several reasons can explain why we applied our method to this specific task. In order to predict AU, the first step is to detect a set of landmarks on faces. Afterwards, features are extracted in order to describe the face and then used for training the models. Commonly used features for this task are of two different kinds: geometric and appearance features.

Geometric features aim at characterizing the landmark shape independently of the appearance of the face. Appearance features aim at describing the texture of the different areas of the face, e.g. in order to gather information about expressive-related wrinkles. Those wrinkles are particularly relevant for AU prediction. For instance, the emergence of nasolabial folds is a strong indicator of the activation of AU12 (Lip Corner Puller). Histograms of Oriented Gradients (HOG) (detailed in chapter 3) are often used as appearance features in AU prediction systems. We believe that the relationship between those features and the intensity of AU has no reason to be linear. Let us consider a feature that is potentially relevant for characterizing wrinkles linked to eyebrow frowning (AU4), e.g. the amount of vertical gradients between the eyebrows. The relationship between this feature and the corresponding AU4 intensity is probably monotonically increasing. The more vertical gradients there are in the area, the more the eyebrow frowning intensity should be important. However, the relationship between them has no reason to be linear. When starting non-activated to reach a mid-level activation, the amount of horizontal gradients may for instance increase slowly, and when going from a mid-level activation towards a high-level activation, the emergence of a deeper or larger wrinkle may lead to a very rapid increase of the amount of vertical gradients. This explains why we believe that the non-linearity of the Nadaraya-Watson estimator used in our proposed regression method can lead to a precise prediction of AU intensities.

The second reason why we decided to use our proposed H-MT-MLKR method for this task is because we believe that it is of primary importance to focus on overfitting when designing AU prediction systems. Indeed, the difficulty of data labeling leads to a number of available activations that may be very limited for some AU. Some AU of very rarely activated in natural behaviors. Even when collecting and labeling a large amount of video data, the resulting number of activations may be small. Limited data combined with the difficulty of the task due to large morphological differences among humans makes AU prediction prone to overfitting.

Our method lets us learn by making the prior assumption that some features may be useful for predicting several AU. It seems unlikely for a feature to be useful for the prediction of the intensities of both the smile and the eyebrow raising. However, features such as the angle between the mouth corner, the center of the upper lip and the eye corner may potentially be useful for both AU12 (Lip Corner Puller) and AU10 (Upper Lip Raiser) for instance.

In this chapter, we present the application of the H-MT-MLKR method to AU intensity prediction. We first present the BP4D dataset (that is used for our experiments) in section 4.2. Then, we detail our framework in section 4.3. Our results can be found in section 4.4. Finally, we conclude in section 4.5.

4.2 The BP4D dataset

The Binghamton-Pittsburgh (BP4D) dataset [127] includes videos of 41 participants (56% female, 49% white, ages from 18 to 29). To elicit relevant facial expressions, different tasks were asked to the participants by a professional actor. The procedures were designed to elicit a range of emotions and facial expressions including happiness, sadness, surprise, embarrassment, fear, physical pain, anger and disgust. Several experts labeled frame-by-frame the intensities of five AU: AU6 (Cheek Raiser), AU10 (Upper Lip Raiser), AU12 (Lip Corner Puller), AU14 (Dimpler) and AU17 (Chin Raiser). Figure 4.1 contains image samples extracted from the BP4D dataset.



Fig. 4.1 Examples of images extracted from the BP4D dataset

In figure 4.2, we represented two images with corresponding intensities for activated AU. On the left side, the subject is raising its cheeks and its upper lip while smiling. On the right side, the subject is raising its chin and activating dimpler in addition to previous actions. This illustrates how coding the intensities of those five AU leads to a precise description of the lower area of the face.

We used this dataset for evaluating our framework in the context of our participation to the FERA'15 challenge. This competition has been challenging participants to automatically analyze facial expressions. One sub-challenge tackled the issue of fully automatic AU intensity estimation. For that sub-challenge, the BP4D dataset has been divided into three subsets: a training set, a development set, as well as a test set that was not available for the participants in order for the evaluation to be fair.

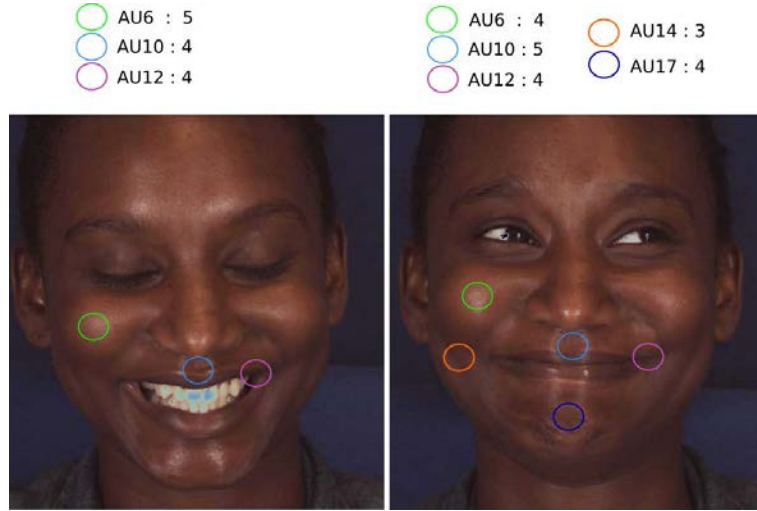


Fig. 4.2 Two images extracted from the BP4D dataset with corresponding activated AU intensities

4.3 AU prediction framework

In this section, we detail our framework for AU intensity prediction. First, we present the features that we extracted. Then, we discuss how we modified our proposed H-MT-MLKR method for learning efficiently in the context of data samples that come from videos.

4.3.1 Feature extraction

Note: in this framework, we use the Intraface tracker [117] that localizes 49 facial landmarks in real-time. This work has been realized before our work on landmark localization (chapter 3), which explains our tracker choice.

In this framework, we use both geometric and appearance features. We recall that geometric features are information relative to the locations of key landmarks in faces (eyes, nose, eyebrows and mouth contours), and appearance features aim at describing the image texture (globally or locally). For AU prediction, landmark locations contain particularly relevant information because some AU activations directly induce important key point movements (as when rising the eyebrows or smiling). However, it is important to combine geometric features with appearance ones for at least two reasons. First, geometric features cannot encode some crucial information for AU prediction as expression-relative wrinkle characterization. Second, current trackers may suffer from a lack of precision or robustness in challenging conditions and appearance can make up for those potential errors in landmark localization.

Geometric features

A few works on facial expression analysis use geometric features obtained after projection onto a manifold learned by Principal Component Analysis (PCA) [69] [74]. However, we believe that it would not be relevant here because those features encode the information in a global manner. AU prediction being a local task and in order to be insensitive to scaling and rotation in the image plane, we extracted geometric features relative to point triplets (as in [74]). For each point triplet $\mathbf{t}_{k_1k_2k_3} = (\mathbf{p}_{k_1}, \mathbf{p}_{k_2}, \mathbf{p}_{k_3})$, we computed the ratio of both vectors

$$\mathbf{v}_{k_2k_3} = \mathbf{p}_{k_3} - \mathbf{p}_{k_2} = (\mathbf{p}_{k_3}^x - \mathbf{p}_{k_2}^x) + i \cdot (\mathbf{p}_{k_3}^y - \mathbf{p}_{k_2}^y)$$

and

$$\mathbf{v}_{k_2k_1} = \mathbf{p}_{k_1} - \mathbf{p}_{k_2} = (\mathbf{p}_{k_1}^x - \mathbf{p}_{k_2}^x) + i \cdot (\mathbf{p}_{k_1}^y - \mathbf{p}_{k_2}^y)$$

to form

$$f(\mathbf{t}_{k_1k_2k_3}) = \frac{\mathbf{v}_{k_2k_1}}{\mathbf{v}_{k_2k_3}} = \frac{\|\mathbf{v}_{k_2k_1}\|}{\|\mathbf{v}_{k_2k_3}\|} \cdot e^{i(\widehat{\mathbf{v}_{k_2k_3}, \mathbf{v}_{k_2k_1}})}$$

that indicates the location of \mathbf{p}_{k_1} relative to \mathbf{p}_{k_2} and \mathbf{p}_{k_3} . Then, we use the norm and the angle of $f(\mathbf{t}_{k_1k_2k_3})$ as features.

Appearance features

Before extracting appearance features, we canceled the rotation in the image plane and normalized the image using the estimations of the eye centers (that are often the most reliable points in landmark tracking methods). We illustrate the pre-processing step of our framework in figure 4.3. After the rotation, we crop the image with a width of 2.2 inter-ocular distances and an height defined for letting 1 inter-ocular distance upon the center of the eyes and 0.6 inter-ocular distance under the center of the lower contour of the inferior lip. After that, we resize the image into a 150 by 150 pixels image.

Then, we extracted HOG on different image patches. Some of our chosen patches are centered on and around the landmarks (for describing local texture and be able to capture expression-related wrinkles), and others are obtained by an 8 by 8 division of the image (see figure 4.4), letting us the possibility to catch up for potential point tracking errors. The centers of the patches defined using the landmarks are presented in figure 4.4. The size of those patches is the same as those obtained by the 8 by 8 regular division of the image.

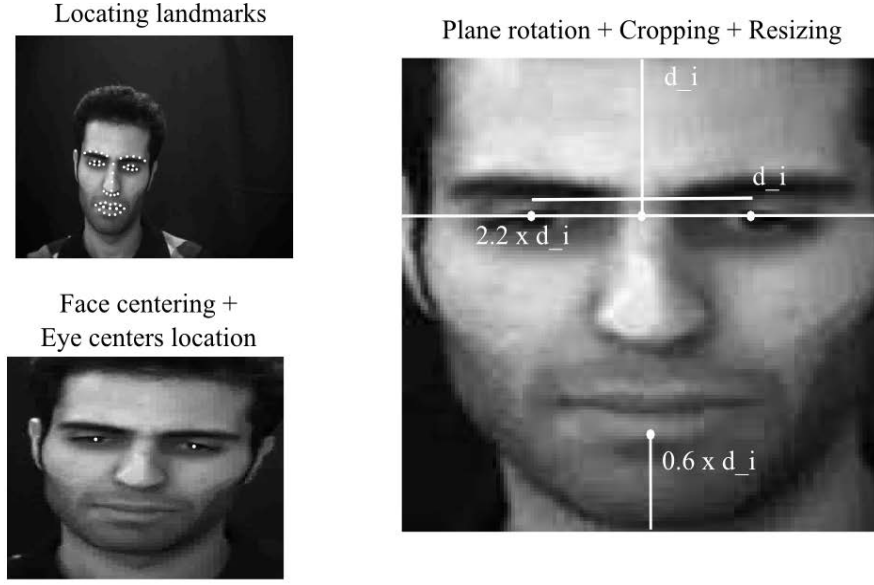


Fig. 4.3 Image pre-processing step of our AU prediction framework

We normalize the features globally using the following equation:

$$\hat{b}_{i,j} = \frac{b_{i,j}}{\sqrt{\sum_{l \in \mathcal{I}} b_{l,j}^2}}$$

with $\{b_{i,j}, i \in \mathcal{I}, j \in \llbracket 1; 8 \rrbracket\}$ the initial histograms and \mathcal{I} the complete set of patches. Details about different normalizations have been presented in chapter 3. We chose to use a global normalization in order to bring out the areas containing important gradients in comparison to the mean rate of gradients on the subject skin. This way, we aim at being insensitive to the age differences between subjects because age-related wrinkles often spread out on larger areas of skin than expression-related ones.

We used our proposed H-MT-MLKR method considering the prediction of each AU as one task. We refer to chapter 2 for details about the regression method. In next subsection, we explain how we modified the proposed H-MT-MLKR method for a use in the context of video data.

4.3.2 About learning with video data

In numerous face-involving machine learning applications, databases may contain several samples corresponding to the same subjects. In order to design a database of labeled images for AU prediction, it is common to record videos of different subjects performing a task of

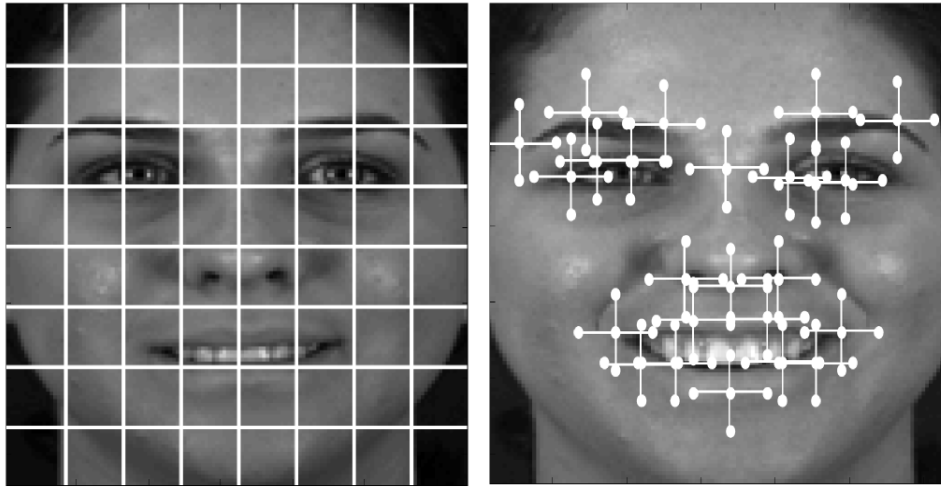


Fig. 4.4 On the left: patches defined without the landmarks. On the right: centers of the patches defined using the landmarks.

a few minutes. Afterward, videos are coded in terms of AU frame by frame by a pool of experts. It results from that protocol that many images come from the same subjects in most AU databases.

For evaluating a system, it is thus important to separate the data into different parts (for training and testing) that do not contain common subjects. The evaluation is said to be subject-independent. This results in a more objective estimation of the system performance in a real-life use, when the system has to deal with previously unseen subjects.

Moreover, depending on the learning methods, the information that many samples come from the same subjects may be important. For example, when using the MLKR method (and a fortiori when using the different extensions that we introduced), we learn a space in which the training samples are projected. This space is estimated by minimizing a cost function computed by predicting samples using the Nadaraya-Watson estimator. This non-parametric estimator predicts a sample label as a mean of the labels of every other samples weighted by some similarity measure. When the estimator makes use of all other samples in the training set for predicting some sample label, it also makes use of the other samples of the same subject, which can be an important issue. Indeed, if the images are extracted from video sequences, the continuity of movements implies that the samples that lie close to some considered sample in the space will probably correspond to neighbor images in the video, that most likely have the same AU labels as the image that is to be predicted (the recording frame rate being often highly superior to most facial action speed). As a consequence, any projection space may lead to a low cost function. However, this will obviously not indicate

the relevance of the learned space. This can be an issue in any metric learning method using a non-parametric estimator when the training set contains same-subject samples.

For solving this issue, we considered a Nadaraya-Watson estimator that predicts a sample only making use of the samples corresponding to other subjects. This amounts to place zeros in the kernel matrix for every couple of samples belonging to the same subject. We represented in figure 4.5 two examples of kernel matrices, one corresponding to an estimator using all the other samples, and one corresponding to an estimator only using the other subjects samples for prediction. Using this technique, the training samples are estimated

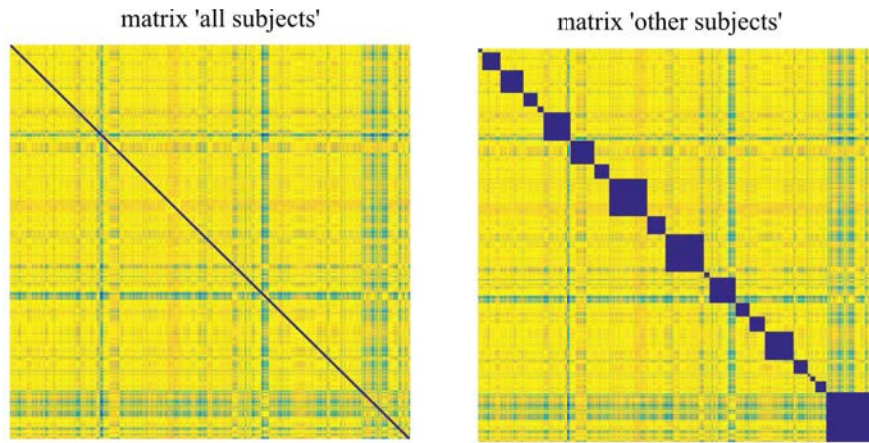


Fig. 4.5 Two kernel matrices corresponding to different estimators

the same way as samples will be estimated during test or in a real-life use, which lets us learn relevant spaces. In the next subsection, we present our experimental setup and give the values of the hyper-parameters that have been used for learning the different systems whose results are presented in subsection 4.4.

4.3.3 Experimental setup

The optimization of the hyper-parameters has been performed on a subject independent four-folds cross-validation on the concatenation of the training and the development datasets. HOG features have been computed in eight directions. We extracted a total of 2768 features. We selected $n_d = 80$ features for each experiment using conditional entropy. For our multi-task extensions, feature selection was performed using the sum of conditional entropies over the different tasks. We randomly selected 10.000 images for reducing the memory cost of the kernel computations. The dimension of the projection spaces that we learned was $n_r = 5$. The number of common axis for our H-MT-MLKR method was $n_c = 3$. The optimal hyper-parameter that we found for regularization was $\gamma = 0.9$.

4.4 Results on the BP4D dataset

In this section, we first present an analysis of the most impacting features that define our different learned spaces in order to stress the relevance of our multi-task approach. Then, we present an evaluation of our method as well as the challenge results.

4.4.1 Analysis of feature impact

In this subsection, we discuss the impact of features on the different AU by analyzing our obtained learned subspaces. In figure 4.6, we represented the main features that form the axis of the common subspace, as well as the five specific spaces. Let $\mathbf{B} \in \mathcal{M}_{n_r, n_f}(\mathbb{R})$ (with n_r the number of axis and n_f the number of features) be a considered space. We computed:

$$V(i) = \max(\{|\mathbf{B}_{j,i}|, j \in \llbracket 1; n_r \rrbracket\})$$

for each feature i and selected the features corresponding to the four highest values of V . White lines between fiducial points indicate that the angle between corresponding vectors had important impact. Black arrows indicate that HOG extracted in the area along the indicated direction had important impact. We can notice that the angle between the extremity of the right eye, the center of the inferior lip and the right mouth corner is an important feature for all tasks, which can be explained by the fact that this angle varies a lot when AU12 (Lip Corner Puller), AU10 (Upper Lip Raiser) or AU17 (Chin Raiser) are activated. Appearance around nasolabial folds appears to be important for all tasks as well. As for specific spaces, we can notice the importance of the appearance between the right ear and the right cheek for AU6 (Cheek Raiser), or external to the right mouth corner for AU14 (Dimpler). We can also notice that the eyebrows appear to be useful for predicting AU6 (Cheek Raiser), AU10 (Upper Lip Raiser) and AU14 (Dimpler). This can be explained by existing correlations between AU in natural facial expressions (for instance, it is rare to frown eyebrows when raising cheeks). We can be surprised by the fact that no appearance area around the chin appeared to have important impact for AU17 (Chin Raiser). However, we can notice that an appearance area on the inferior lip, which is close to the chin, has important impact on the definition of the common space. We believe that the relevance of the features that define the common space as well as the different specific spaces gives an idea about the behavior of our proposed method. Next subsection contains evaluations and results.

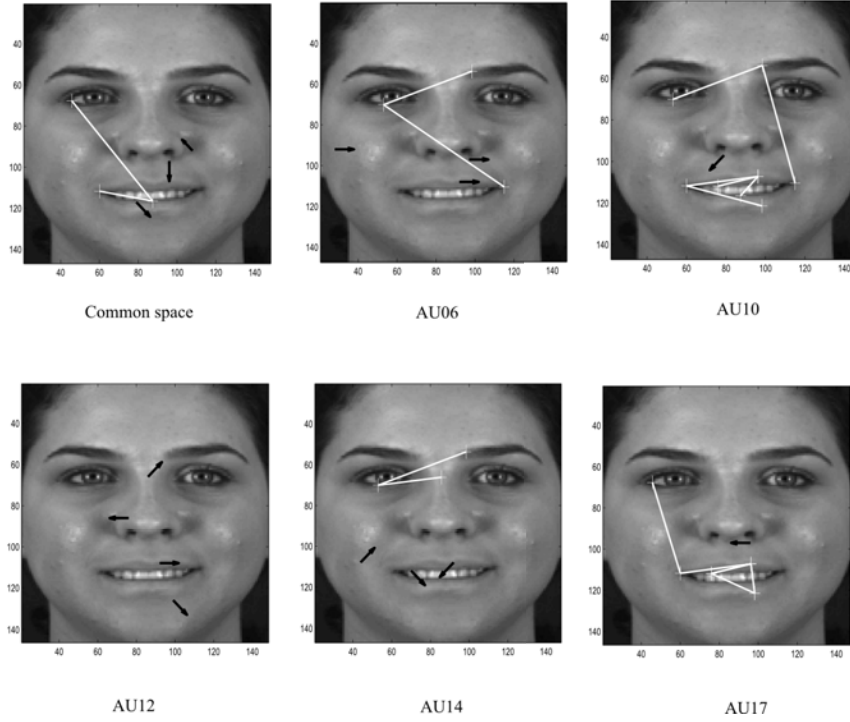


Fig. 4.6 Illustration of the four most impacting features for each learned subspace. White lines indicate point triplet angles and black arrows indicate HOG features.

4.4.2 Evaluations and results on the BP4D dataset

In this subsection, we first evaluate the impact of the proposed MT and H-MT regularizations. Then, we compare our system to the baseline systems on the FERA'15 development and test sets. Finally, we present the FERA'15 challenge results.

4.4.3 Evaluation of regularization impact

In table 4.1, we compare our H-MT-MLKR method to two simpler models (MLKR and MT-MLKR) in a subject independent four-folds cross-validation on the concatenation of the training and the development datasets in terms of Pearson's correlation coefficient. We can notice improvements when using the standard multi-task regularization of MLKR compared to initial MLKR algorithm for four of the five considered AU. For AU12, which is the most accurately predicted one, the results stay similar when using standard multi-task regularization. We can also notice that the more constrained H MT-MLKR method significantly improves the results for four of the five considered AU with a mean improvement of 5% over the initial MLKR algorithm. The highest improvement (of more than 15%) corresponds to AU14, that appears to be the hardest one to predict for our system.

Table 4.1 Comparison between standard MLKR and the two proposed multi-task extensions of MLKR in terms of Pearson’s correlation coefficient (in percentage)

| AU | MLKR | MT-MLKR | H-MT-MLKR |
|------|-------------|---------|-------------|
| 6 | 74.2 | 75.4 | 76.3 |
| 10 | 70.9 | 72.8 | 75.2 |
| 12 | 86.6 | 86.4 | 86.5 |
| 14 | 41.4 | 44.3 | 47.7 |
| 17 | 52.6 | 52.7 | 54.8 |
| Mean | 65.1 | 66.3 | 68.1 |

4.4.4 Comparison to baseline systems on the FERA’15 development set

In table 4.2, we compare the results we obtain to the baseline results of the FERA’15 fully-continuous AU intensity prediction challenge. The geometric and appearance baseline features are detailed in [99]. The prediction method used in the baseline system is a Support Vector Regressor (SVR). The evaluation corresponds to systems learned on the training dataset and tested on the development dataset. Results are given in terms of Pearson’s correlation coefficient. In the baseline paper, the results are presented separately for geometric and appearance features. For being able to compare our system to the baseline, we present results of three H-MT-MLKR systems learned with only geometric (G), only appearance (A), and both geometric and appearance (F) features. We can notice a mean improvement of 14% over the baseline with our complete system. The systems learned using only appearance features or only geometric features are outperformed by the complete system for each AU. We can notice that for AU6 and AU17, appearance features lead to better results than geometric features. We can also notice that, for AU6, AU12, AU14 and AU17, any of our systems greatly outperforms both baseline systems. For AU17, we obtain a correlation coefficient of 60.6% with our complete system while the best baseline system leads to 36.5%.

4.4.5 Comparison to baseline systems on the FERA’15 test set

In table 4.3, we compare the results we obtain on the test partition to the baseline results in terms of Intraclass Correlation Coefficient (ICC [92]) and Mean Squared Error (MSE). The FERA’15 official measure was ICC. We can notice that our system greatly outperforms the baseline system in terms of both ICC and MSE for four AU. We can also notice that the baseline systems for AU17 lead to better results than our system in terms of MSE. However, for that same AU, our system leads to results that are almost three times superior to the baseline results in terms of ICC. This can be explained by the fact that, for an imbalanced

Table 4.2 Comparison between baseline system (B) and proposed H-MT-MLKR (H) in terms of Pearson’s correlation coefficient in percentage for different feature subsets (only geometric (G), only appearance (A) and the fusion of both (F))

| AU | B, G | H, G | B, A | H, A | H, F |
|------|------|-------------|------|-------------|-------------|
| 6 | 69.9 | 74 | 72 | 78.2 | 78.2 |
| 10 | 71.5 | 72.3 | 68.3 | 70.4 | 75.4 |
| 12 | 70.6 | 85.2 | 69.5 | 84.8 | 85.6 |
| 14 | 47.2 | 50 | 39.6 | 47.4 | 54.2 |
| 17 | 36.5 | 56.2 | 30.3 | 58 | 60.6 |
| Mean | 59.2 | 67.5 | 55.9 | 67.8 | 70.8 |

AU (with a very small activation rate), taking small risks, meaning predicting all samples very close to zeros, may lead to a low MSE, even when the prediction is weakly correlated with ground truth.

Table 4.3 Comparison between geometric and appearance baseline systems (G and A) and proposed H-MT-MLKR (H) in terms of ICC (I) in percentage and MSE (M)

| AU | G, I | A, I | H, I | G, M | A, M | H, M |
|------|------|------|-------------|--------------|-------|--------------|
| 6 | 67 | 62.2 | 78.7 | 1.004 | 1.366 | 0.829 |
| 10 | 73.2 | 65.6 | 80.1 | 0.897 | 1.209 | 0.801 |
| 12 | 78 | 76.7 | 86 | 0.738 | 1.092 | 0.622 |
| 14 | 58.6 | 38.9 | 71 | 1.227 | 1.526 | 1.14 |
| 17 | 14.4 | 16.8 | 44.3 | 0.806 | 0.819 | 0.844 |
| Mean | 58.2 | 52 | 72 | 0.934 | 1.202 | 0.847 |

4.4.6 Comparison to other participants on the FERA’15 test set

In table 4.4, we compare the results we obtain on the test partition to the results obtained by the other participants in terms of ICC. Our system corresponds to the ISIR team.

Our proposed H-MT-MLKR method let us win the fully continuous FERA’15 challenge. We can notice that we greatly outperformed all teams for each AU. We obtain a mean ICC over the five AU corresponding to a 13% increase over the second best team (from Cambridge university). We can notice that the AU with the greater performance gap compared to the other participants is AU14. We obtain a 71.1% ICC compared to 54.9% for the second best team for that AU (the VicarVision company). We believe that our proposed regularization has been an important element in this gap. Indeed, the most important improvement of our

Table 4.4 Results of the FERA'2015 fully continuous challenge in terms of ICC.

| AU | ISIR [73] | Rainbow Group [2] | KIT | VicarVision [39] | LaBRI [68] |
|------|-------------|-------------------|------|------------------|------------|
| 6 | 78.7 | 71.9 | 67.8 | 66.4 | 72 |
| 10 | 80.2 | 71.8 | 73.1 | 73.4 | 72 |
| 12 | 86.1 | 82.8 | 82.6 | 78.8 | 78.4 |
| 14 | 71.1 | 54.6 | 53.3 | 54.9 | 27.7 |
| 17 | 44.3 | 37.7 | 30.8 | 32.9 | 26.8 |
| Mean | 72.1 | 63.8 | 61.5 | 61.3 | 55.4 |

regularization compared to the initial MLKR method was obtained for AU14. For this AU, it is possible that the available dataset was not containing enough positive and various samples for learning a precise model without overfitting, which could explain the obtained results.

4.5 Conclusion

In this chapter, we presented a complete framework for AU intensity prediction. It is based on our proposed multi-dimensional label extension of MLKR, namely Hard Multi-Task MLKR. This method lets us identify common features, that impacts several AU, and features that are specific to each AU at the same time. It also leads to non-linear estimations of the relationship between features and labels, which we believe is particularly important when using appearance features. We evaluated the impact of the proposed method over both standard multi-task extension and the initial MLKR method showing how our proposed regularization greatly improves AU intensity prediction. We obtained a 5% improvement on the BP4D database using H-MT-MLKR compared to the initial MLKR method in terms of Pearson's correlation coefficient.

We also evaluated our system in the context of the FERA'15 challenge. Our H-MT-MLKR framework let us win the first place of the FERA'15 fully continuous challenge. We obtained a mean result of 72% in terms of Intraclass Correlation Coefficient compared to 64% for the second team (Cambridge university) and to 58% for the baseline system on the test set.

Note: a previous system that we designed for AU intensity prediction was based on a Lasso extension of the MLKR regression method. The current version of the paper can be found in appendix A.

Chapter 5

Conclusion and future works

5.1 Conclusion

During this PhD, we designed a novel regression method, the so-called Hard Multi-Task Metric Learning for Kernel Regression (H-MT-MLKR) method. We applied it with success to two different computer vision applications, namely automatic landmark localization and facial muscle activation prediction. The method is based on the Nadaraya-Watson regressor, which is both non-linear and non-parametric. Using this regressor, the label of an unknown data sample is predicted as a weighted mean of training labels. The weights are computed using a Gaussian kernel in order for closest samples to have more impact on the prediction than farther ones. This non-parametric local regressor is in a sense a continuous version of the mainly used k-nearest-neighbors algorithm. We chose to use this regressor in our method because it is a very natural prediction approach to consider that close data points should have close labels. Because the whole prediction depends entirely on the distances between data samples, the space in which the samples lie is a key element. The relevance of the prediction highly depends on this space. The MLKR algorithm proposed by Weinberger and Tesauro [111] aims at learning a subspace for minimizing the quadratic error of a Nadaraya-Watson regressor on the training samples. Our work includes a precise analysis of the MLKR algorithm along with several modifications and extensions for predicting multi-dimensional label with a particular focus on overfitting reduction.

In the first chapter of this dissertation, automatic facial information extraction was introduced along with state-of-the-art methods relative to three sub-domains, namely landmark localization, facial Action Units (AU) intensity prediction and emotion prediction in a continuous space. Collecting and labeling relevant datasets for learning to automatically extract information from faces is a very complex task. Some information may be hard to detect even for humans (e.g. emotional states), and thus even harder to label in a precise manner. In many

cases, a robust learning requires a very large amount of data making the database design very time-consuming. In order for the learned models to be relevant for real-life applications, there is a need for the recorded human behaviors to be both natural and various. Moreover, some actions are very rarely occurring in natural behaviors, reducing the rate of relevant data samples. Because of all those difficulties, the learning models have to be designed by trying to make the best use of all information available in the databases. This was the main focus of our work. It explains that our proposed model aims at predicting data with multi-dimensional labels while minimizing as much as possible its degrees of freedom in order to reduce potential overfitting.

Basic principles of machine learning were introduced in the second chapter of this dissertation along with a precise analysis of the MLKR model relative to its complexity, its training ease, its robustness to noise and the extrapolation capabilities of its prediction function. This analysis let us explain the choice of our proposed modifications and extensions. Among them, a filter-based feature selection step based on conditional entropy, the use of a batch stochastic gradient descent and several regularizations designed for multi-dimensional labels prediction. The core idea of our main extension, Hard Multi-Task regularization, is to learn several spaces for the different dimensions enforcing some of the axis to be shared among all dimensions while other axis are left free for capturing each dimension particularities.

In the third chapter, the application of our method for landmark localization was presented. Recently, works of Cao et al. [6] and Xiong and De la Torre [117] led to the emergence of Cascaded-Regression approaches for facial landmark localization. A shape is initialized in the image and then refined step by step using regression methods. Information about the appearance of the image relative to the current shape is used as input for predicting the displacements of the landmarks towards ground truth. The framework we designed lets us learn local features while including a global shape constraint using the common axis of our learned subspaces. Results show that our method greatly outperforms the standard method of [117] and leads to similar results as the very recent approach of Ren et al. [80]. We also highlighted in this chapter an advantage of subspace learning methods for overfitting reduction performing the regression by embedding data samples that have not been used for learning the spaces.

In the fourth chapter of this dissertation, we presented an application of our method for AU intensity prediction. We evaluated in this chapter the gain brought by our proposed H-MT-MLKR framework relative to the initial MLKR approach. We evaluated our results by competing in the Facial Expression Recognition and Analysis challenge (FERA'15). We obtained the best results for fully-continuous AU intensity prediction. The obtained results were highly outperforming the other participants, that included the commercial company Vi-

carVision¹, as well as high-level teams such as Cambridge University. We believe that those results highlight the relevance of multi-dimensional labels subspace learning approaches based on the Nadaraya-Watson regressor for automatic information extraction from faces as well as the need of focusing on overfitting reduction.

Subspace learning methods have numerous strengths. The first one is the possibility to easily analyze the learned model, as it corresponds to a linear distortion of the initial feature space. This can be very useful while designing the systems because the links between the obtained prediction and the initial data are very straightforward. A second strength is the possibility to easily modify the cost function for adapting to a specific problem as a simple gradient computation is sufficient for testing the new model. As an example, we modified the cost function of H-MT-MLKR for learning efficiently with video data when different samples come from the same subjects. The simplicity of design and use of subspace learning methods (as well as their efficiency) opens a wide range of prospects.

5.2 Future works

Our works on automatic information extraction from face-centered data let us identify a few ideas that could be relevant to investigate. Because data is, according to us, the element that has the most impact on the quality of face-related prediction systems, our first idea is related to database design. We present this idea in subsection 5.2.1. Further investigations about the advantages of being able to embed new data samples in the spaces for performing the regression (samples that have not been used for learning the spaces) could be conducted. We discuss it in subsection 5.2.2. Our proposed method is hard to use without modifications or prior steps for large and high dimensional datasets. We present a few ideas relative to handling big data sets in subsection 5.2.3.

5.2.1 Towards coupling database design and model training

The question of the relationship between database design and model training may be crucial to investigate. The quality of the database (including its size, the variability of its data samples and the precision of its labeling) has a huge impact on the obtained system performance in automatic information extraction from faces. Thus, database design and model training that are at our knowledge always separated in our domain, may need more interaction. We think that new data acquisition protocols fusing database design and model training may lead to a performance increase for the systems as well as a speed up for the database design. As an

¹<http://www.vicarvision.nl/>

example, we could define parameters for the database, as (for face-centered data), the number of different subjects and the number of samples for each subject. Afterwards, different iterative acquisitions may be done for building the database step after step. A model training step could be performed using the current database and a gradient of the system performance relative to the database parameters may be computed for each label. Those gradients may be used for optimizing the amount of data to acquire for each task and knowing which samples to label. This way, the cost of the acquisition could be reduced by knowing that each step of the database design will lead to an increment of the learned system performance for a specific task.

5.2.2 Towards smart system adaptation

Our proposed regression method is what we called a semi-parametric method. The predictions are performed using both the model parameters (defining the spaces) and the training data. This property may be particularly interesting for adapting the system to a specific subject (sex, age, facial morphology) or context (lighting, camera parameters, head pose, occlusions). As an example, let us consider that the subject is able to take time for a calibration phase. Before the calibration phase, the subspaces have been learned using the training data in order to identify which feature combinations seemed to define relevant axis for the considered prediction task. Performing the predictions using the training data lead to predictions that have been globally optimized for the different subjects and contexts included in the training database. Embedding and appropriately weighting a few data points that are specific to the subject and context (collected during the calibration phase) is straightforward, does not need a re-training phase, does not add significant computation complexity, lead to a smart calibration of the system and, as a consequence, to a potential performance increase.

5.2.3 Towards handling big data sets

Our proposed method has been designed in the context of landmark localization and AU prediction for research purposes. For those applications, we worked on mid-sized datasets (number of samples and features both inferior to 100k) and designed systems working in close-to-real-time speed (15 fps). In this context, the important memory complexity, the important training time complexity, and the important computation time complexity of the method could be handled by modifications we presented in this PhD (as the stochastic gradient descent or the feature selection step). Those modifications also let us reduce the overfitting issue. In our applications, the optimal number of selected features as well as the optimal batch size for the gradient descent were both possible to handle with one standard

computer (Intel i7-3770, 3.4 Ghz). However, those modifications may not be sufficient to reduce the (training time, memory and computation time) complexities of the proposed method when dealing with industrial big data sets.

Reducing the computation time complexity while keeping the subspace-estimator couple that is optimized using our method is not straightforward. Indeed, the Nadaraya-Watson estimator makes use of all the distances from the test sample towards training data samples. However, the estimation makes use of a Gaussian kernel and is, as a consequence, local in some sense, as the similarities vanish with large distances. Because of that property, the computation time may be reduced with an acceptable precision loss by dividing the spaces into smaller areas. The mapping between a sample and its corresponding area could be performed using a binary tree. However, this may not be efficient for some systems which performance highly relies on the estimation of uncertainty of the predictions (which may need all samples for being precisely estimated).

Reducing the training time or memory complexities of the method is far from being straightforward. An idea could be to learn random forests of H-MT-MLKR spaces (randomizing both features and samples), for easily parallelizing the training step. However, the loss of performance induced by this modification is an unknown parameter that has to be estimated on the considered dataset. This raises the open question of the potential relevance of subspace learning regression methods for big data applications.

References

- [1] Baltrusaitis, T., Banda, N., and Robinson, P. (2013). Dimensional affect recognition using continuous conditional random fields. In *Automatic Face and Gesture Recognition (FG 2013), 10th IEEE International Conference and Workshops on*, pages 1–8. IEEE.
- [2] Baltrusaitis, T., Mahmoud, M., and Robinson, P. (2015). Cross-dataset learning and person-specific normalisation for automatic action unit detection. In *Automatic Face and Gesture Recognition (FG 2015), 11th IEEE International Conference and Workshops on*. IEEE.
- [3] Batur, A. U. and Hayes, M. H. (2005). Adaptive active appearance models. *Image Processing, IEEE Transactions on*, 14(11):1707–1721.
- [4] Belhumeur, P. N., Jacobs, D. W., Kriegman, D. J., and Kumar, N. (2013). Localizing parts of faces using a consensus of exemplars. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(12):2930–2940.
- [5] Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In *COMPSTAT'2010, Proceedings of*, pages 177–186. Springer.
- [6] Cao, X., Wei, Y., Wen, F., and Sun, J. (2014). Face alignment by explicit shape regression. *International Journal of Computer Vision*, 107(2):177–190.
- [7] Castellano, G., Kessous, L., and Caridakis, G. (2008). Emotion recognition through multiple modalities: face, body gesture, speech. In *Affect and Emotion in Human-Computer Interaction*, pages 92–103. Springer.
- [8] Chang, Y., Hu, C., Feris, R., and Turk, M. (2006). Manifold based analysis of facial expression. *Image and Vision Computing*, 24(6):605–614.
- [9] Chang, Y., Vieira, M., Turk, M., and Velho, L. (2005). Automatic 3d facial expression analysis in videos. In *Analysis and Modelling of Faces and Gestures (AMFG 2005)*, pages 293–307. Springer.
- [10] Chu, W.-S., De la Torre, F., and Cohn, J. F. (2013). Selective transfer machine for personalized facial action unit detection. In *Computer Vision and Pattern Recognition (CVPR 2013), IEEE Conference on*, pages 3515–3522. IEEE.
- [11] Chuang, C.-F. and Shih, F. Y. (2006). Recognizing facial action units using independent component analysis and support vector machine. *Pattern Recognition*, 39(9):1795–1798.

- [12] Cohen, I., Sebe, N., Garg, A., Chen, L. S., and Huang, T. S. (2003). Facial expression recognition from video sequences: temporal and static modeling. *Computer Vision and Image Understanding*, 91(1):160–187.
- [13] Cohn, J. F. (2006). Foundations of human computing: facial expression and emotion. In *Multimodal Interfaces, Proceedings of the 8th International Conference on*, pages 233–238. ACM.
- [14] Cohn, J. F. (2010). Advances in behavioral science using automated facial image analysis and synthesis. *Signal Processing Magazine*, 27(6):128–133.
- [15] Cootes, T. F., Edwards, G., and Taylor, C. (1999). Comparing active shape models with active appearance models. In *British Machine Vision Conference (BMVC 1999)*, pages 173–182. Springer.
- [16] Cootes, T. F., Edwards, G. J., and Taylor, C. J. (1998). Active appearance models. In *European Conference on Computer Vision (ECCV 1998)*, pages 484–498. Springer.
- [17] Cootes, T. F., Edwards, G. J., and Taylor, C. J. (2001). Active appearance models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(6):681–685.
- [18] Cootes, T. F., Ionita, M. C., Lindner, C., and Sauer, P. (2012). Robust and accurate shape model fitting using random forest regression voting. In *European Conference on Computer Vision (ECCV 2012)*, pages 278–291. Springer.
- [19] Cootes, T. F. and Taylor, C. J. (1992). Active shape models ‘smart snakes’. In *British Machine Vision Conference (BMVC 1992)*, pages 266–275. Springer.
- [20] Cootes, T. F. and Taylor, C. J. (1993). Active shape model search using local grey-level models: A quantitative evaluation. In *British Machine Vision Conference (BMVC 1993)*, volume 93, pages 639–648. Springer.
- [21] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3):273–297.
- [22] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition (CVPR 2005), IEEE Conference on*, volume 1, pages 886–893. IEEE.
- [23] Darwin, C. (2002). *The expression of the emotions in man and animals*. Oxford University Press.
- [24] Dhall, A., Goecke, R., Joshi, J., Sikka, K., and Gedeon, T. (2014). Emotion recognition in the wild challenge 2014: Baseline, data and protocol. In *International Conference on Multimodal Interaction (ICMI 2014), Proceedings of the 16th on*, pages 461–466. ACM.
- [25] Dhall, A., Goecke, R., Joshi, J., Wagner, M., and Gedeon, T. (2013). Emotion recognition in the wild challenge 2013. In *International Conference on Multimodal Interaction (ICMI 2013), Proceedings of the 15th on*, pages 509–516. ACM.
- [26] Drucker, H., Burges, C. J., Kaufman, L., Smola, A., Vapnik, V., et al. (1997). Support vector regression machines. *Advances in Neural Information Processing Systems*, 9:155–161.

- [27] Ekman, P. (2003). Darwin, deception, and facial expression. *Annals of the New York Academy of Sciences*, 1000(1):205–221.
- [28] Ekman, P., Friesen, W., and Hager, J. (2002). Emotional facial action coding system. *Manual and investigators guide. CD-ROM*.
- [29] Ekman, P. and Friesen, W. V. (1971). Constants across cultures in the face and emotion. *Journal of personality and social psychology*, 17(2):124.
- [30] Ekman, P. and Friesen, W. V. (1976). Measuring facial movement. *Environmental psychology and nonverbal behavior*, 1(1):56–75.
- [31] Ekman, P. and Friesen, W. V. (1981). The repertoire of nonverbal behavior: Categories, origins, usage, and coding. *Nonverbal communication, interaction, and gesture*, pages 57–106.
- [32] Ekman, P. and Friesen, W. V. (1986). A new pan-cultural facial expression of emotion. *Motivation and emotion*, 10(2):159–168.
- [33] El Kaliouby, R. and Robinson, P. (2005). Real-time inference of complex mental states from facial expressions and head gestures. In *Real-time vision for human-computer interaction*, pages 181–200. Springer.
- [34] Evgeniou, T. and Pontil, M. (2004). Regularized multi-task learning. In *International conference on Knowledge Discovery and Data mining (SIGKDD 2004), Proceedings of the 10th ACM on*, pages 109–117. ACM.
- [35] Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188.
- [36] Fontaine, J. R., Scherer, K. R., Roesch, E. B., and Ellsworth, P. C. (2007). The world of emotions is not two-dimensional. *Psychological science*, 18(12):1050–1057.
- [37] Freund, Y. and Schapire, R. E. (1995). A decision-theoretic generalization of on-line learning and an application to boosting. In *Computational learning theory*, pages 23–37. Springer.
- [38] Girard, J. M. (2014). *Automatic detection and intensity estimation of spontaneous smiles*. PhD thesis, University of Pittsburgh.
- [39] Gudi, A., Tasli, H. E., den Uyl, T. M., and Maroulis, A. (2015). Deep learning based faces action unit occurrence and intensity estimation. In *Automatic Face and Gesture Recognition (FG 2015), the 11th IEEE International Conference and Workshops on*.
- [40] Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182.
- [41] Harms, M. B., Martin, A., and Wallace, G. L. (2010). Facial emotion recognition in autism spectrum disorders: a review of behavioral and neuroimaging studies. *Neuropsychology review*, 20(3):290–322.

- [42] He, L., Jiang, D., Yang, L., Pei, E., Wu, P., and Sahli, H. (2015). Multimodal affective dimension prediction using deep bidirectional long short-term memory recurrent neural networks. In *Proceedings of the 5th International Workshop on Audio/Visual Emotion Challenge*, pages 73–80. ACM.
- [43] Hjortsjö, C.-H. (1969). *Man's face and mimic language*.
- [44] Hu, S. and Zheng, G. (2009). Driver drowsiness detection with eyelid related parameters by support vector machine. *Expert Systems with Applications*, 36(4):7651–7658.
- [45] Jeni, L. A., Girard, J. M., Cohn, J. F., and De La Torre, F. (2013). Continuous au intensity estimation using localized, sparse facial feature space. In *Automatic Face and Gesture Recognition (FG 2013), the 10th IEEE International Conference and Workshops on*, pages 1–7. IEEE.
- [46] Kächele, M., Schels, M., and Schwenker, F. (2014). Inferring depression and affect from application dependent meta knowledge. In *Proceedings of the 4th International Workshop on Audio/Visual Emotion Challenge*, pages 41–48. ACM.
- [47] Kahou, S. E., Pal, C., Bouthillier, X., Froumenty, P., Gülcehre, C., Memisevic, R., Vincent, P., Courville, A., Bengio, Y., Ferrari, R. C., et al. (2013). Combining modality specific deep neural networks for emotion recognition in video. In *International Conference on Multimodal Interaction (ICMI 2013), Proceedings of the 15th ACM on*, pages 543–550. ACM.
- [48] Kanade, T., Cohn, J., and Tian, Y. (2000). Comprehensive database for facial expression analysis. In *Automatic Face and Gesture Recognition (FG 2000). Proceedings of the 4th IEEE International Conference on*, pages 46–53.
- [49] Kanluan, I., Grimm, M., and Kroschel, K. (2008). Audio-visual emotion recognition using an emotion space concept. In *16th European Signal Processing Conference*.
- [50] Kapoor, A., Burleson, W., and Picard, R. W. (2007). Automatic prediction of frustration. *International journal of human-computer studies*, 65(8):724–736.
- [51] Kohler, C. G., Turner, T. H., Bilker, W. B., Brensinger, C. M., Siegel, S. J., Kanes, S. J., Gur, R. E., and Gur, R. C. (2003). Facial emotion recognition in schizophrenia: intensity effects and error pattern. *American Journal of Psychiatry*.
- [52] Le, V., Brandt, J., Lin, Z., Bourdev, L., and Huang, T. S. (2012). Interactive facial feature localization. In *European Conference on Computer Vision (ECCV 2012)*, pages 679–692. Springer.
- [53] Li, Y., Chen, J., Zhao, Y., and Ji, Q. (2013). Data-free prior model for facial action unit recognition. *Affective Computing, IEEE Transactions on*, pages 127–141.
- [54] Littlewort, G., Bartlett, M. S., Fasel, I., Susskind, J., and Movellan, J. (2006). Dynamics of facial expression extracted automatically from video. *Image and Vision Computing*, 24(6):615–625.

- [55] Littlewort, G. C., Bartlett, M. S., and Lee, K. (2007). Faces of pain: automated measurement of spontaneous facial expressions of genuine and posed pain. In *International conference on Multimodal interfaces, Proceedings of the 9th on*, pages 15–21. ACM.
- [56] Liu, M., Wang, R., Li, S., Shan, S., Huang, Z., and Chen, X. (2014). Combining multiple kernel methods on riemannian manifold for emotion recognition in the wild. In *International Conference on Multimodal Interaction (ICMI 2014), Proceedings of the 16th*, pages 494–501. ACM.
- [57] Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *International Conference on Computer Vision (ICCV 1999), Proceedings of the 7th IEEE*, volume 2, pages 1150–1157. IEEE.
- [58] Lucey, P., Cohn, J., Kanade, T., Saragih, J., Ambadar, Z., and Matthews, I. (2010). The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *Computer Vision and Pattern Recognition Workshops (CVPRW 2010), IEEE Conference on*, pages 94–101.
- [59] Lucey, P., Cohn, J. F., Prkachin, K. M., Solomon, P. E., and Matthews, I. (2011). Painful data: The unbc-mcmaster shoulder pain expression archive database. In *Automatic Face and Gesture Recognition (FG 2011), 9th IEEE International Conference and Workshops on*, pages 57–64. IEEE.
- [60] Matthews, I. and Baker, S. (2004). Active appearance models revisited. *International Journal of Computer Vision*, 60(2):135–164.
- [61] Mavadati, S. M., Mahoor, M. H., Bartlett, K., Trinh, P., and Cohn, J. F. (2013). Disfa: A spontaneous facial action intensity database. *Affective Computing, IEEE Transaction on*, 4(2):151–160.
- [62] McDuff, D., El Kaliouby, R., Senechal, T., Amr, M., Cohn, J. F., and Picard, R. (2013). Affectiva-mit facial expression dataset (am-fed): Naturalistic and spontaneous facial expressions collected "in-the-wild". In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*, pages 881–888. IEEE.
- [63] Mehrabian, A. (1977). *Nonverbal communication*. Transaction Publishers.
- [64] Meng, H. and Bianchi-Berthouze, N. (2011). Naturalistic affective expression classification by a multi-stage approach based on hidden markov models. In *Affective Computing and Intelligent Interaction (ACII 2011)*, pages 378–387. Springer.
- [65] Meng, H. and Bianchi-Berthouze, N. (2014). Affective state level recognition in naturalistic facial and vocal expressions. *Cybernetics, IEEE Transactions on*, 44(3):315–328.
- [66] Meng, H., Huang, D., Wang, H., Yang, H., Al-Shuraifi, M., and Wang, Y. (2013). Depression recognition based on dynamic facial and vocal expression features using partial least square regression. In *Proceedings of the 3rd ACM international workshop on Audio/Visual Emotion Challenge*, pages 21–30. ACM.

- [67] Milborrow, S. and Nicolls, F. (2008). Locating facial features with an extended active shape model. In *European Conference on Computer Vision (ECCV 2008)*, pages 504–513. Springer.
- [68] Ming, Z., Bugeau, A., Rouas, J.-L., and Shochi, T. (2015). Facial action units intensity estimation by the fusion of features with multi-kernel support vector machine. In *11th IEEE International Conference on Automatic Face and Gesture Recognition Conference and Workshops*.
- [69] Murthy, G. and Jadon, R. (2009). Effectiveness of eigenspaces for facial expressions recognition. *International Journal of Computer Theory and Engineering*, 1(5):1793–8201.
- [70] Nadaraya, E. A. (1964). On estimating regression. *Theory of Probability and Its Applications*, 9(1):141–142.
- [71] Nicolaou, M. A., Gunes, H., and Pantic, M. (2012). Output-associative rvm regression for dimensional and continuous emotion prediction. *Image and Vision Computing*, 30(3):186–196.
- [72] Nicolaou, M. A., Zafeiriou, S., and Pantic, M. (2013). Correlated-spaces regression for learning continuous emotion dimensions. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 773–776. ACM.
- [73] Nicolle, J., Bailly, K., and Chetouani, M. (2015). Facial action unit intensity prediction via hard multi-task metric learning for kernel regression. In *Automatic Face and Gesture Recognitions (FG 2015), FERA 2015 Challenge, 11th IEEE International Conference on*.
- [74] Nicolle, J., Bailly, K., Rapp, V., and Chetouani, M. (2013). Locating facial landmarks with binary map cross-correlations. In *International Conference on Image Processing (ICIP 2013)*, pages 2978–2982.
- [75] Nicolle, J., Rapp, V., Bailly, K., Prevost, L., and Chetouani, M. (2012). Robust continuous prediction of human emotions using multiscale dynamic cues. In *International Conference on Multimodal Interaction (ICMI 2012), Proceedings of the 14th ACM*, pages 501–508. ACM.
- [76] Ojala, T., Pietikainen, M., and Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):971–987.
- [77] Pantic, M., Patras, I., and Valstar, M. (2005). Learning spatio-temporal models of facial expressions. In *Proceedings of International Conference on Measuring Behaviour*.
- [78] Prkachin, K. M. and Solomon, P. E. (2008). The structure, reliability and validity of pain expression: Evidence from patients with shoulder pain. *Pain*, 139(2):267–274.
- [79] Ramirez, G. A., Baltrusaitis, T., and Morency, L.-P. (2011). Modeling latent discriminative dynamic of multi-dimensional affective signals. In *Affective Computing and Intelligent Interaction (ACII 2011)*, pages 396–406. Springer.

- [80] Ren, S., Cao, X., Wei, Y., and Sun, J. (2014). Face alignment at 3000 fps via regressing local binary features. In *Computer Vision and Pattern Recognition (CVPR 2014), IEEE Conference on*, pages 1685–1692. IEEE.
- [81] Ringeval, F., Schuller, B., Valstar, M., Jaiswal, S., Marchi, E., Lalanne, D., Cowie, R., and Pantic, M. (2015). Avec 2015 - the first affect recognition challenge bridging across audio, video, and physiological data. In *Proceedings of the 22st ACM international conference on Multimedia*. ACM.
- [82] Rudovic, O., Pavlovic, V., and Pantic, M. (2012). Kernel conditional ordinal random fields for temporal segmentation of facial action units. In *European Conference on Computer Vision (ECCV 2012), Workshops and Demonstrations*, pages 260–269. Springer.
- [83] Sagonas, C., Tzimiropoulos, G., Zafeiriou, S., and Pantic, M. (2013). 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *International Conference on Computer Vision Workshops (ICCVW 2013)*, pages 397–403. IEEE.
- [84] Saragih, J. M., Lucey, S., and Cohn, J. F. (2011). Deformable model fitting by regularized landmark mean-shift. *International Journal of Computer Vision*, 91(2):200–215.
- [85] Savran, A., Alyüz, N., Dibeklioglu, H., Celiktutan, O., Gökberk, B., Sankur, B., and Akarun, L. (2008). Bosphorus database for 3d face analysis. In *Biometrics and Identity Management*, pages 47–56. Springer.
- [86] Savran, A., Cao, H., Shah, M., Nenkova, A., and Verma, R. (2012a). Combining video, audio and lexical indicators of affect in spontaneous conversation via particle filtering. In *International Conference on Multimodal Interaction (ICMI 2012), Proceedings of the 14th ACM*, pages 485–492. ACM.
- [87] Savran, A., Sankur, B., and Taha Bilge, M. (2012b). Regression-based intensity estimation of facial action units. *Image and Vision Computing*, 30(10):774–784.
- [88] Schuller, B., Valstar, M., Eyben, F., McKeown, G., Cowie, R., and Pantic, M. (2011). Avec 2011—the first international audio/visual emotion challenge. In *Affective Computing and Intelligent Interaction*, pages 415–424. Springer.
- [89] Schuller, B., Valster, M., Eyben, F., Cowie, R., and Pantic, M. (2012). Avec 2012: the continuous audio/visual emotion challenge. In *International Conference on Multimodal Interaction (ICMI 2012), Proceedings of the 14th ACM*, pages 449–456. ACM.
- [90] Senechal, T., Rapp, V., Salam, H., Segulier, R., Bailly, K., and Prevost, L. (2011). Combining aam coefficients with lgbp histograms in the multi-kernel svm framework to detect facial action units. In *Automatic Face and Gesture Recognition (FG 2011), 9th IEEE International Conference and Workshops on*, pages 860–865. IEEE.
- [91] Shen, J., Zafeiriou, S., Chrysos, G. G., Kossaifi, J., Tzimiropoulos, G., and Pantic, M. (2015). The first facial landmark tracking in-the-wild challenge: Benchmark and results. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 50–58.
- [92] Shrout, P. E. and Fleiss, J. L. (1979). Intraclass correlations: uses in assessing rater reliability. *Psychological bulletin*, 86(2):420.

- [93] Sun, Y., Reale, M., and Yin, L. (2008). Recognizing partial facial action units based on 3d dynamic range data for facial expression recognition. In *Automatic Face and Gesture Recognition (FG 2008), the 8th IEEE International Conference on*, pages 1–8. IEEE.
- [94] Sun, Y., Wang, X., and Tang, X. (2013). Deep convolutional network cascade for facial point detection. In *Computer Vision and Pattern Recognition (CVPR 2013), IEEE Conference on*, pages 3476–3483. IEEE.
- [95] Tian, Y.-l., Kanade, T., and Cohn, J. F. (2001). Recognizing action units for facial expression analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(2):97–115.
- [96] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, pages 267–288.
- [97] Tong, Y., Liao, W., and Ji, Q. (2007). Facial action unit recognition by exploiting their dynamic and semantic relationships. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(10):1683–1699.
- [98] Tzimiropoulos, G., Alabort-i Medina, J., Zafeiriou, S., and Pantic, M. (2013). Generic active appearance models revisited. In *Asian Conference on Computer Vision (ACCV 2013)*, pages 650–663. Springer.
- [99] Valstar, M., Girard, J., Almaev, T., McKeown, G., Mehu, M., Yin, L., Pantic, M., and Cohn, J. (2015). Fera 2015 - second facial expression recognition and analysis challenge. In *Automatic Face and Gesture Recognition (FG 2015), 11th IEEE International Conference and Workshops on*. IEEE.
- [100] Valstar, M., Martinez, B., Binefa, X., and Pantic, M. (2010). Facial point detection using boosted regression and graph models. In *Computer Vision and Pattern Recognition (CVPR 2010), IEEE Conference on*, pages 2729–2736. IEEE.
- [101] Valstar, M., Schuller, B., Smith, K., Almaev, T., Eyben, F., Krajewski, J., Cowie, R., and Pantic, M. (2014). Avec 2014: 3d dimensional affect and depression recognition challenge. In *Proceedings of the 4th International Workshop on Audio/Visual Emotion Challenge*, pages 3–10. ACM.
- [102] Valstar, M., Schuller, B., Smith, K., Eyben, F., Jiang, B., Bilakhia, S., Schnieder, S., Cowie, R., and Pantic, M. (2013). Avec 2013: the continuous audio/visual emotion and depression recognition challenge. In *Proceedings of the 3rd ACM international workshop on Audio/visual emotion challenge*, pages 3–10. ACM.
- [103] Valstar, M. F., Jiang, B., Mehu, M., Pantic, M., and Scherer, K. (2011). The first facial expression recognition and analysis challenge. In *Automatic Face and Gesture Recognition (FG 2011), 9th IEEE International Conference and Workshops on*, pages 921–926. IEEE.
- [104] Valstar, M. F. and Pantic, M. (2012). Fully automatic recognition of the temporal phases of facial actions. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 42(1):28–43.

- [105] Valstar, M. F., Pantic, M., Ambadar, Z., and Cohn, J. F. (2006). Spontaneous vs. posed facial behavior: automatic analysis of brow actions. In *Proceedings of the 8th international conference on Multimodal interfaces*, pages 162–170. ACM.
- [106] Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition (CVPR 2001), Proceedings of the 2001 Conference on*, volume 1. IEEE.
- [107] Wan, S. and Aggarwal, J. (2013). Spontaneous facial expression recognition: A robust metric learning approach. *Pattern Recognition*.
- [108] Wang, Y. and Guan, L. (2005). Recognizing human emotion from audiovisual information. In *Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on*, volume 2. IEEE.
- [109] Wang, Y., Lucey, S., and Cohn, J. F. (2008). Enforcing convexity for improved alignment with constrained local models. In *Computer Vision and Pattern Recognition (CVPR 2008), IEEE Conference on*, pages 1–8. IEEE.
- [110] Weinberger, K. Q., Blitzer, J., and Saul, L. K. (2005). Distance metric learning for large margin nearest neighbor classification. In *Advances in neural information processing systems*, pages 1473–1480.
- [111] Weinberger, K. Q. and Tesauro, G. (2007). Metric learning for kernel regression. In *International Conference on Artificial Intelligence and Statistics*, pages 612–619.
- [112] Williamson, J. R., Quatieri, T. F., Helfer, B. S., Ciccarelli, G., and Mehta, D. D. (2014). Vocal and facial biomarkers of depression based on motor incoordination and timing. In *Proceedings of the 4th International Workshop on Audio/Visual Emotion Challenge*, pages 65–72. ACM.
- [113] Williamson, J. R., Quatieri, T. F., Helfer, B. S., Horwitz, R., Yu, B., and Mehta, D. D. (2013). Vocal biomarkers of depression based on motor incoordination. In *Proceedings of the 3rd International Workshop on Audio/Visual Emotion Challenge*, pages 41–48. ACM.
- [114] Wöllmer, M., Kaiser, M., Eyben, F., Schuller, B., and Rigoll, G. (2013). Lstm-modeling of continuous emotions in an audiovisual affect recognition framework. *Image and Vision Computing*, 31(2):153–163.
- [115] Xiao, R., Li, M.-J., and Zhang, H.-J. (2004). Robust multipose face detection in images. *Circuits and Systems for Video Technology, IEEE Transactions on*, 14(1):31–41.
- [116] Xiao, S., Yan, S., and Kassim, A. (2015). Facial landmark detection via progressive initialization. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 33–40.
- [117] Xiong, X. and De la Torre, F. (2013). Supervised descent method and its applications to face alignment. In *Computer Vision and Pattern Recognition (CVPR 2013), IEEE Conference on*, pages 532–539. IEEE.

- [118] Yan, J., Lei, Z., Yi, D., and Li, S. Z. (2013). Learn to combine multiple hypotheses for accurate face alignment. In *Computer Vision Workshops (ICCVW 2013), IEEE International Conference on*, pages 392–396. IEEE.
- [119] Yang, J., Deng, J., Zhang, K., and Liu, Q. (2015). Facial shape tracking via spatio-temporal cascade shape regression. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 41–49.
- [120] Yang, P., Liu, Q., and Metaxas, D. N. (2007). Boosting coded dynamic features for facial action units and facial expression recognition. In *Computer Vision and Pattern Recognition (CVPR 2007), IEEE Conference on*, pages 1–6. IEEE.
- [121] Yang, S. and Bhanu, B. (2011). Facial expression recognition using emotion avatar image. In *Automatic Face and Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, pages 866–871. IEEE.
- [122] Yik, M. S., Russell, J. A., and Barrett, L. F. (1999). Structure of self-reported current affect: Integration and beyond. *Journal of personality and social psychology*, 77(3):600.
- [123] Yuce, A., Gao, H., and Thiran, J.-P. (2015). Discriminant multi-label manifold embedding for facial action unit detection. In *Automatic Face and Gesture Recognitions (FG 2015), FERA 2015 Challenge, 11th IEEE International Conference on*.
- [124] Zafeiriou, S., Zhang, C., and Zhang, Z. (2015). A survey on face detection in the wild: past, present and future. *Computer Vision and Image Understanding*.
- [125] Zeng, Z., Tu, J., Liu, M., Huang, T. S., Pianfetti, B., Roth, D., and Levinson, S. (2007). Audio-visual affect recognition. *Multimedia, IEEE Transactions on*, 9(2):424–428.
- [126] Zhang, W., Shan, S., Gao, W., Chen, X., and Zhang, H. (2005). Local gabor binary pattern histogram sequence (lgbphs): A novel non-statistical model for face representation and recognition. In *Computer Vision (ICCV 2005), 10th IEEE International Conference on*, volume 1, pages 786–791. IEEE.
- [127] Zhang, X., Yin, L., Cohn, J. F., Canavan, S., Reale, M., Horowitz, A., Liu, P., and Girard, J. M. (2014). Bp4d-spontaneous: a high-resolution spontaneous 3d dynamic facial expression database. *Image and Vision Computing*, 32(10):692–706.
- [128] Zhang, Y. and Ji, Q. (2005). Active and dynamic information fusion for facial expression understanding from image sequences. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(5):699–714.
- [129] Zhao, G. and Pietikainen, M. (2007). Dynamic texture recognition using local binary patterns with an application to facial expressions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(6):915–928.
- [130] Zhao, K., Chu, W.-S., De la Torre, F., Cohn, J. F., and Zhang, H. (2015). Joint patch and multi-label learning for facial action unit detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2207–2216.

-
- [131] Zhou, E., Fan, H., Cao, Z., Jiang, Y., and Yin, Q. (2013). Extensive facial landmark localization with coarse-to-fine convolutional network cascade. In *Computer Vision Workshops (ICCVW 2013), IEEE International Conference on*, pages 386–391. IEEE.
- [132] Zhu, X. and Ramanan, D. (2012). Face detection, pose estimation, and landmark localization in the wild. In *Computer Vision and Pattern Recognition (CVPR 2012), IEEE Conference on*, pages 2879–2886. IEEE.

Appendix A

Iterative Regularized Metric Learning

This appendix contains a work on Action Unit intensity prediction based on a Lasso extension of Metric Learning for Kernel Regression (MLKR).

Real-Time Facial Action Unit Intensity Prediction with Regularized Metric Learning

J          , K          , Mohamed Chetouani

Abstract

Being able to automatically infer emotional states, engagement, depression or pain from non-verbal behaviors has recently become of great interest to lots of research or industrial works. This will bring the emergence of a wide range of applications in robotics, biometrics, marketing or either medicine. The Facial Action Coding System (FACS) proposed by Ekman lets to objective description of facial movements, characterizing activations of facial muscles. Achieving an accurate intensity prediction of those Action Units (AUs) has a significant impact on the prediction quality of more high-level information regarding human behavior (e.g. emotional states). Real-time AUs intensity prediction, as many image-related machine learning tasks, is a high dimensional problem. For solving this task, we propose in this paper to adapt the Metric Learning for Kernel Regression (MLKR) framework focusing on overfitting issues induced in high dimensional spaces. MLKR aims at estimating the optimal linear subspace for reducing the squared error of a Gaussian kernel regressor. We introduce Iterative Regularized Kernel Regression (IRKR), an iterative nonlinear feature selection method combined with a Lasso-regularized version of the original MLKR formulation, improving state-of-the-art results on several AUs databases, from prototypical to natural and wild data.

Keywords: Facial Expression, Action Units, FACS, Metric Learning for Kernel Regression

1. Introduction

Automatic facial expression recognition has recently become a very active and rapidly evolving research domain. In order to precisely describe facial expressions, the Facial Action Coding System (FACS [1]) encodes Action Units (AUs), that correspond to the activations of facial muscles.

Being able to accurately predict AUs intensity has a significant impact for human behavior assessment. During a video, being able to describe at each frame what facial muscles are activated and how much they are gives us a complete description of a subject's facial movements. It contains precious information for mental states [2], depression [3] and pain [4] [5] prediction for instance. Industrial applications that take advantage of AUs predictions are numerous as well. Applications in marketing [6] or either Human-Computer Interaction [7] have recently emerged.

In this paper, we address three main issues: First, AUs automatic prediction has mainly been seen as a classification problem. However, being able to predict muscle activations more precisely is essential.

Very small and short activations of AUs (called micro-expressions) can be of great value for emotion assessment [8]. Moreover, dynamics of AUs has an important impact on the meaning of facial expressions. In [9], the authors worked on classifying two different kinds of smiles (frustrated and delighted) showing the relevance of temporal pattern analysis for this task. For those reasons, multi-levels annotated databases have recently been released (enhanced CK+ [10], DISFA dataset [11], AM-FED dataset [6]), making it possible to build and evaluate new methods suited for regression tasks. The second issue is that the algorithms should be real-time, which is an important constraint for many domains such as personal robotics or car passenger security. This constraint encourages fast-to-compute features and fast regression methods. Finally, some AUs are very rarely activated in natural behaviors as Nose Wrinkler (AU9) or Lip Stretcher (AU20). This makes the number of positive examples small, even when the amount of acquired video data is important. Thereby, a particular focus on the risk of overfitting on the training data must be made.

We propose a regression method based on a Lasso-regularization of MLKR included within an itera-

tive nonlinear feature selection framework. This method lets us project data points into sparse and low dimensional spaces letting us reduce overfitting issues. In Section 2, we present a brief state-of-the-art of AUs prediction methods. Section 3 contains an outline of our framework and the paper contributions. In Section 4, we present MLKR, on which our regression method is built upon and discuss some of its advantages. Section 5 describes our proposed regression method. Its application to AUs intensity prediction and the associated results are presented in Section 6. Finally, we conclude and discuss a few issues and perspectives in Section 7.

2. Related Works

Numerous AUs prediction methods have been proposed during the past decade along with the growing interest for this domain. Detecting AUs is a supervised machine learning problem. Face-centered data is acquired (gray-level, RGB and/or depth-map) and labeled by humans. The labels indicate the different muscles activated by the subject. Then, we need to extract features describing data before learning a prediction model. Because AUs are related to local changes in facial expression, it is common to use a facial landmark detector to localize the different parts of the face (mouth, eyes, nose, eyebrows). The features can afterwards be extracted on different facial areas. Those features characterizing data samples are then used for predicting labels with a supervised machine learning algorithm. Along all the data processing chain, from the acquisition sensors to the prediction method, many questions have been highlighted by past works. First, the availability of affordable 3D-sensors has attracted many researchers to focus on the utility and contribution of depth-related data for facial muscle activation predictions and has made the data type a relevant question. Second, the choice of the areas used for feature extraction has an important impact. Third, including prior human knowledge when designing high-level features relevant to the task can increase performance but leads to less generic methods. In a similar way, including prior knowledge within the models (e.g. about AUs co-occurrences in natural facial expressions) has also raised questions. Finally, the choice of the learning machines used to model the data has also been an active topic in past works. In this section, we will briefly review and discuss some of the main AUs prediction methods recently proposed.

The relevance of using 3-dimensional data for facial expression recognition has been investigated by several researchers. Sun et al. [12] used 3D motion vectors and Hidden Markov Models (HMMs) for predicting AUs

and discrete emotions on Dynamic 3D Facial Expression Database. Savran et al. [13] extracted local 3D shape features (mean and Gaussian curvatures, shape index and curvedness among others) and use an SVM for predicting AUs on Bosphorus database. However, 3D sensors are not yet widely democratized and many applications have a need of 2D data solutions, which explains the numerous recent 2D approaches for AUs prediction [14] [11] [15]. Most of those 2D approaches can be easily extended to 3D approaches, extracting complementary features using depth maps the same way than gray-level or color images.

Before extracting features from images, a common first step in many face-centered machine learning systems is to detect fiducial points, that are some key points in faces (centers and corners of the eyes, contours of the nose, the mouth and the eyebrows). In Jeni et al. [16] and Chu et al. [17], those fiducial points are used to define local patches for feature extraction in order to predict AUs. However, a few methods [18] [19] avoid this part of fiducial point localization extracting features on more or less global regions only defined using the area obtained with the face detector (commonly using Viola and Jones algorithm [20]). Yang et al. [18] directly extracted dynamic Haar-like features after a rescaling of the detected face image and then encoded it with binary patterns before classifying using Adaboost [21]. Chuang and Shih [19] divided the face region in upper and lower parts before using Support Vector Machine (SVM) on Independent Component Analysis (ICA) projections. Other methods only use eye localization for defining feature extraction areas [10] [22]. By definition, AUs are characterized by local movements of face appearance. This is why extracting features on local areas defined from fiducial points lead to relevant information for our task. However, using more global areas defined only using the face region or the centers of the eyes (which are the most accurately located points in most landmark detection methods), can avoid the spread of possible errors in facial point tracking. The recent improvement of facial point localization systems can explain the fact that local areas are more and more used in AUs prediction systems [16] [17] [15].

AUs prediction methods also differ regarding the amount of human knowledge included in the feature choice. Some methods use data-driven features, which often makes the framework more generic, as Chuang and Shih [19] that used Independent Component Analysis (ICA) or Jeni et al. [16] using Non-negative Matrix Factorization (NMF). Even if it introduces a loss of genericity, other methods use handcrafted features, that may lead to relevant invariance and characteriza-

tions. Rudovic et al. [23] used Local Binary Patterns (LBPs) that are invariant to illumination changes. Gabor wavelets are commonly used [10] [22] [13] and have shown to give promising results for AUs prediction as pointed by Littlewort et al. [24]. However, dense computation of those features for different scales and orientations quickly becomes time-consuming and unsuited for real-time algorithms. It can explain the choice of Histograms of Oriented Gradients (HOG) made by McDuff et al. [6] which encode relevant information for expression-relative wrinkle characterization while being less time-consuming to extract.

Prior knowledge can also be included in data modeling. Several researchers focused on learning dynamic relationships and co-occurrences between AUs in order to increase algorithm performance, as Tong et al. [10] and Li et al. [14] that use Dynamic Bayesian Networks (DBNs). These approaches are able to take into account correlations between AUs in natural facial expressions. For instance, eyebrow rising (AU1+AU2) and upper lid rising (AU5) are often activated simultaneously. However, AUs correspond to facial muscles and can be activated in an independent manner, making the prior knowledge about dynamic relations between AUs not adequate in some applications. For instance, in the context of facial reeducation for patients that had a cerebrovascular accident (CVA), different muscles may need to be separately activated by the patient and thus separately recognized. A prior knowledge inclusion in this case could bias the prediction system.

Finally, there is the question of the machine learning algorithms used for building prediction models. In many databases (Cohn-Kanade [25], Carnegie Mellon University PIE database [26], Fera-Gemep [27]) AUs are labeled as activated or not, stating the problem as a classification one. Thus, Support Vector Machines (SVMs) have been widely used in the facial expression domain [22] [28] [6]. A need for a greater precision in AUs recognition systems has motivated the availability of new databases approaching the task as a regression one (Bosphorus [29], CK+ [30], UNBC-McMaster [31], DISFA [11]). However, the choice of optimal machine learning algorithms for AUs intensity prediction stays an open question. For solving this task, Savran et al. [13] adapt a classification learning machine for regression. For doing that, they use SVM and afterwards perform a logistic regression on the non-thresholded SVM outputs. On the contrary, Jeni et al. [16] directly use a regression learning method, Support Vector Regression (SVR), for AUs intensity prediction, obtaining excellent results on Enhanced Cohn-Kanade (CK+) database. Recently, Girard et al. [32] focused on smile

intensity prediction, showing that SVR outperformed multiclass SVM.

Our choices regarding the issues highlighted by this state-of-the-art are presented in the next section in an overview of our regression framework.

3. Overview

In figure 1, we present the architecture of our system. We use gray-level images as raw data type to ensure a wide range of applications. We chose both geometric and appearance features. Our geometric features characterize relationships among triplets of fiducial points in order to be insensitive to rotations and scaling. For appearance features, we use Histograms of Oriented Gradients (HOGs) on local patches because of their relevance for describing emotion-related wrinkles and their low computation time. Some of our patches are centered using the fiducial points. Other patches are located only using the Viola-Jones face detection area in order to be robust in case of a landmark tracking failure. More details about our features can be found in Section 6.1. Those features and the associated labels are then used for learning our prediction system.

Labeling AUs is complex and time-consuming for several reasons. Only experts, with specific training, can precisely identify the activated muscles and their corresponding intensities in an image [33]. Thus, frame by frame annotation of an important number of AUs is difficult (there are more than 45 muscles in a human face). Moreover, in natural behaviors, many AUs are very rarely activated. It explains that, even with several hours of video data, the number of positive activations can be small (*e.g.* there is only 4 activations of maximal intensity for AU2 in DISFA database). The data is said to be imbalanced (the number of unactivated samples is considerably higher than the number of activated ones). Thus, we decided to focus on overfitting when designing our method.

For each AU, we learn a low-dimensional space suited for a non-parametric Gaussian kernel regressor by using a Lasso-regularized version of MLKR within an iterative nonlinear feature selection process. The small number of dimensions in our representation spaces and the regularization aim at reducing a potential overfitting on the training data. Moreover, the imbalanced data distribution induces some issues for regression evaluation when using commonly used metrics as Root Mean Square Error (RMSE) or Correlation Coefficient (CC). We discuss this and introduce a new evaluation metric, *r*-AUC, in Section 6.3.

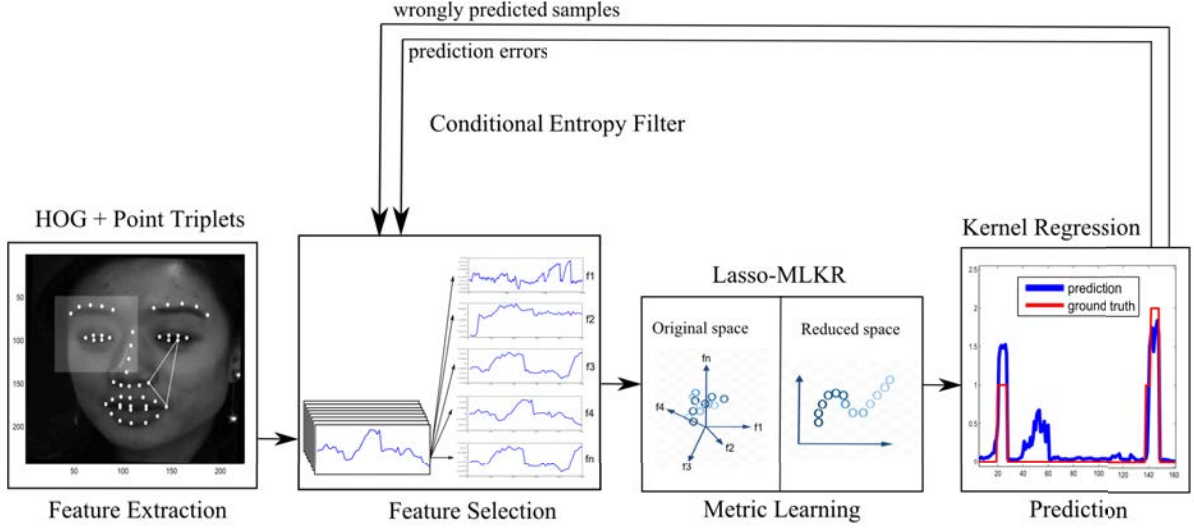


Figure 1: Architecture of the proposed framework

More details about the regression framework we propose can be found in Section 5. The main contributions of this paper are the following:

- A complete framework for real-time AUs intensity prediction improving state-of-the-art results on prototypical and natural databases.
- A Lasso-regularized version of Metric Learning for Kernel Regression (Lasso-MLKR).
- A new evaluation metric (r-AUC), suited for regression tasks on imbalanced data, extending Area Under ROC Curve for regression, that we present in Section 6.3.

Our method is built upon Metric Learning for Kernel Regression (MLKR) that we introduce in the next section.

4. Metric Learning for Kernel Regression

Kernel regression has proven to be efficient in a wide range of applications (from image deblurring [34] or segmentation [35] to automatic human emotion prediction [36]). However, performance of the regressors highly depends on the relevance of the space in which the samples lie, making appropriate dimensionality reduction a needed initial step. Weinberger and Tesauro [37] proposed MLKR (Metric Learning for Kernel Regression), that aims at finding the optimal linear projection to minimize the kernel regression squared error on

the training set.

In kernel regression, an instance label is predicted using the Nadaraya-Watson estimator [38], as an average of the training instance labels weighted using some similarity measure. If we consider n_s training samples $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{n_s}\}$ associated with corresponding labels $\{y_1, y_2, \dots, y_{n_s}\}$, the label corresponding to a feature vector \mathbf{x}_t will be approximated by:

$$\hat{y}_t = \frac{\sum_{i=1}^{n_s} y_i k_{i,t}}{\sum_{i=1}^{n_s} k_{i,t}} \quad (1)$$

using a kernel $k_{i,t} = k(\mathbf{x}_i, \mathbf{x}_t)$ as a similarity metric between samples i and t .

MLKR proposes a direct optimization of the kernel regression error for the commonly used Gaussian kernel, which can be defined as follows:

$$k_{i,j} = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{d_{i,j}^2}{\sigma^2}} \quad (2)$$

with σ the Gaussian spread and $d_{i,j} = d(\mathbf{x}_i, \mathbf{x}_j)$ the euclidean distance between samples i and j . Let us consider an original space of dimension n_d and an output space of dimension n_r . MLKR aims at finding a projection matrix $\mathbf{A} \in \mathcal{M}_{n_d, n_r}(\mathbb{R})$ that minimizes the squared error \mathcal{L} on the training samples

$$\mathcal{L}(\mathbf{A}) = \sum_{i=1}^{n_s} (\hat{y}_i - y_i)^2 \quad (3)$$

where

$$\hat{y}_i = \frac{\sum_{j \neq i} y_j k_{ji}(\mathbf{A})}{\sum_{j \neq i} k_{ji}(\mathbf{A})}$$

with

$$k_{i,j}(\mathbf{A}) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{d_{i,j}(\mathbf{A})^2}{\sigma^2}}$$

$$d_{i,j}(\mathbf{A})^2 = \|\mathbf{A}(\mathbf{x}_i - \mathbf{x}_j)\|^2 = (\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{A}^\top \mathbf{A} (\mathbf{x}_i - \mathbf{x}_j)$$

being the squared distance in the reduced subspace of dimension n_r . The optimization process of the squared error is done with a gradient descent. We obtain by an analytical calculation:

$$\frac{\partial \mathcal{L}(\mathbf{A})}{\partial \mathbf{A}} = 4\mathbf{A} \sum_i \frac{(\hat{y}_i - y_i)}{\sum_{j \neq i} k_{ij}} \sum_j (\hat{y}_i - y_j) k_{ij} (\mathbf{x}_i - \mathbf{x}_j) (\mathbf{x}_i - \mathbf{x}_j)^\top \quad (4)$$

Metric learning for kernel regression (MLKR) lets us project data points in a low-dimensional space suited for a nonlinear prediction via Gaussian kernel regression.

In this paragraph, we explain our choice to use MLKR by discussing some advantages and limitations of the method.

First, MLKR does not directly learn a prediction function but learns a space in which a Nadaraya-Watson estimator is performed using a set of data and labels. We can then project new data points in the learned space for predicting without re-learning the system (for instance in order to easily adapt to a new database or to a specific subject). Second, Nadaraya-Watson estimator is able to adapt easily to heterogeneous point distribution because of the normalization by $\sum_{i=1}^{n_s} k_{i,t}$, which helps AUs intensity prediction when trying to predict an unknown subject lying in a sparsely populated part of the space. However, MLKR has several drawbacks. First, it is non convex. However, experiments have shown that local minima lead to accurate predictions on standard regression datasets [37]. We arrived at the same conclusion with our experiments on AUs databases. Second, it has a quadratic complexity relatively to both the number of features and the number of samples, which makes it difficult to use on high dimensional and large datasets.

In the next section, we introduce our regression method, which is based on an adaptation of MLKR for high dimensional spaces.

5. Iterative Regularized Kernel Regression (IRKR)

In MLKR algorithm, the number of estimated parameters when reducing a space of dimension n_d into a sub-

space of dimension n_r is $n_{par} = n_d \cdot n_r$. If the number of training samples is too small compared to the number of model parameters, the risk of overfitting rises. We propose in Section 5.1 to modify the original formulation by regularizing it using a Lasso-penalty for overfitting risk reduction.

Moreover, the gradient computation for a projection of n_s samples into a space of dimension n_d has a complexity in $O(n_s^2 \cdot n_d^2)$ making it difficult to use in high dimension spaces. We propose a complete framework improving the original MLKR formulation to make it efficient on high-dimensional datasets.

A widely used step for supervised dimensionality reduction is filter feature selection [39], which aims at characterizing the relevance of the features independently of the predictor's choice, often one by one, for predicting the label. In other words, it means to compute a similarity (or dissimilarity) measure between each feature and the label and to select the highest ones (or the smallest ones respectively). We propose to use conditional entropy measure that is able to find nonlinear relationships between features and labels. Details are given in Section 5.2.

Furthermore, we propose to include it within an iterative framework because filter-based methods have a high risk of selecting redundant information (see paragraph 5.3).

5.1. Lasso-MLKR

Original MLKR minimizes the training reconstruction error with respect to the coefficients of the projection matrix \mathbf{A} . We propose to regularize this original MLKR formulation using a Lasso-penalty, meaning that we add a weight to the cost function corresponding to the L^1 -norm of the matrix \mathbf{A} (which is the sum of the absolute values of its coefficients). This penalty has been proven to induce sparsity in the estimated parameters, reducing the risk of overfitting [40]. Some of the coefficients are shrunk all the way to zero. Corresponding solutions, with multiple values that are identically zero, are said to be sparse. The new energy formulation becomes:

$$\mathcal{L}(\mathbf{A}) = \sum_{i=1}^{n_s} (\hat{y}_i - y_i)^2 + \lambda \cdot L_1(\mathbf{A}) \quad (5)$$

where λ controls the regularization rate. The associate gradient becomes:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{A}} = 4\mathbf{A} \sum_i \frac{(\hat{y}_i - y_i)}{\sum_{j \neq i} k_{ij}} \sum_j (\hat{y}_i - y_j) k_{ij} (\mathbf{x}_i - \mathbf{x}_j) (\mathbf{x}_i - \mathbf{x}_j)^\top + \lambda \cdot s(\mathbf{A}) \quad (6)$$

with s the sign function.

The optimization of the regularization rate λ can be done in cross-validation. In IRKR, this Lasso-MLKR method is used after a filter-based selection method and within an iterative process. It corresponds to the Metric Learning step in our system's schema (see figure 1).

It is common in face-related machine learning problems to extract tens of thousands of features for characterizing face appearance. However, the complexity of each step of MLKR algorithm, quadratic with respect to the number of features, makes it complicated to use with such a high number of features. This motivates the feature selection we perform using a nonlinear dissimilarity metric described in the next section.

5.2. Conditional Entropy Feature Selection

The purpose of supervised filter-based feature selection is to identify features that contain relevant information for predicting a label. Different prior assumptions can be made on the functional relationship between features and label. The simplest prior assumption that can be made between features and label is linear dependency. The similarity measure associated with that dependency is Correlation Coefficient. Because our regression method is nonlinear, we chose to use conditional entropy, that is able to discover nonlinear relationships between features and labels. Conditional entropy of a label l given a feature f is defined as follows:

$$H(l|f) = - \sum_{x \in \mathcal{F}} p(x) \sum_{y \in \mathcal{L}} p(y|x) \log(p(y|x))$$

with \mathcal{F} and \mathcal{L} the sample spaces in which the feature and label are respectively defined. Because fine estimations of conditional probabilities can be time-consuming with a high number of samples, we decided to compute the probabilities using six-level quantization of the features.

This metric allows relevant feature selection for predicting labels by assuming nonlinear functional relationships between features and labels. It is used in the Feature Selection step of IRKR (see figure 1).

5.3. Iterative Feature Selection

Filter-based methods have been commonly included in iterative frameworks to select feature sets containing uncorrelated information [41]. In our framework, we firstly select a set of features and apply our regression on the learning database. Then, we select features correlated (in terms of conditional entropy) to the prediction error (for selecting uncorrelated information) and features correlated to the samples with the highest errors

(to rapidly reduce the prediction error). The final framework, Iterative Regularized Kernel Regression (IRKR), that combines our three proposed contributions (Lasso-MLKR, conditional entropy and iterative feature selection), is presented in algorithm 1.

Algorithm 1 Iterative Regularized Kernel Regression

- 1: select a subset of n_s features $\{F\}$ calculating $H(l|f_j)$ for all j
 - 2: $v_{sel} = \{F\}$
 - 3: compute \mathbf{A} using Lasso-MLKR on the feature set v_{sel}
 - 4: calculate the prediction \hat{l}^1 on the training set
 - 5: calculate the prediction squared error $e^1 = (\hat{l}^1 - l)^2$ on the training set
 - 6: calculate the sum of errors of training samples s^1
 - 7: identify the subset S^1 of samples whose errors are superior to the mean of e^1
 - 8: $u = 1$
 - 9: **repeat**
 - 10: $u = u + 1$
 - 11: select a subset of n_s features $\{F_e\}$ calculating $H(e^{u-1}|f_j)$ for all j
 - 12: select a subset of n_s features $\{F_m\}$ calculating $H(l(S^{u-1})|f_j(S^{u-1}))$ for all j
 - 13: $v_{sel} = v_{sel} \cup \{F_e\} \cup \{F_m\}$
 - 14: compute \mathbf{A} using Lasso-MLKR on the feature set v_{sel}
 - 15: calculate the prediction \hat{l}^u on the training set
 - 16: calculate the prediction error $e^u = (\hat{l}^u - l)^2$ on the training set
 - 17: calculate the sum of errors of training samples s^u
 - 18: identify the subset S^u of samples whose errors are superior to the mean of e^u
 - 19: **until** $\frac{s^u}{s^{u-1}} < 0.99$
 - 20: perform Kernel Regression on the test samples in the projected space defined by the lastly learned matrix \mathbf{A}
-

In the next Section, we present the application of IRKR on the task of AUs intensity prediction.

6. Application to AUs Prediction

The feature extraction process is described in Section 6.1, followed by a presentation of the databases we used in Section 6.2. Different metrics commonly used for measuring AUs system performance are discussed in Section 6.3. Finally, we detail our evaluation protocol in Section 6.4, we present the evaluations of the different

parts of IRKR in Section 6.5 followed by our results in Section 6.6.

6.1. Feature Extraction

Most of the methods in facial-related information prediction combine two kinds of features: shape-based features and appearance-based features. Shape-based features are information relative to the positions of key landmarks in faces (eyes, nose, eyebrows and mouth contours), and appearance-based ones aim at describing image texture (globally or locally). For our task, landmark positions contain particularly interesting information because some AUs activations directly induce important key point movements (as when rising the eyebrows or smiling). However, it is important to combine shape-related features with appearance-related ones for at least two reasons. First, shape-based features cannot encode some crucial information for AUs prediction as expression-relative wrinkle characterization. Second, current trackers may suffer from a lack of precision or robustness in challenging conditions and appearance information may compensate for those errors in landmark prediction.

Shape-based features

We use Intraface tracker [42] that localizes 49 facial landmarks in real-time. In order to be insensitive to scaling and rotation in the image plane, we extract features relative to point triplets (as in [43]). Some works on facial expression analysis proposed to use features obtained after projection onto a manifold learned by PCA [44] [36]. However, those features encode global information. AUs prediction is a local task because each AU corresponds to one facial muscle. Thus, we chose to extract information relative to point triplets. For each triplet of points $\mathbf{t}_{k_1 k_2 k_3} = (\mathbf{p}_{k_1}, \mathbf{p}_{k_2}, \mathbf{p}_{k_3})$, we calculate the ratio of both vectors

$$\mathbf{v}_{k_2 k_3} = \mathbf{p}_{k_3} - \mathbf{p}_{k_2} = (\mathbf{p}_{k_3}^x - \mathbf{p}_{k_2}^x) + i.(\mathbf{p}_{k_3}^y - \mathbf{p}_{k_2}^y)$$

and

$$\mathbf{v}_{k_2 k_1} = \mathbf{p}_{k_1} - \mathbf{p}_{k_2} = (\mathbf{p}_{k_1}^x - \mathbf{p}_{k_2}^x) + i.(\mathbf{p}_{k_1}^y - \mathbf{p}_{k_2}^y)$$

to form

$$f(\mathbf{t}_{k_1 k_2 k_3}) = \frac{\mathbf{v}_{k_2 k_1}}{\mathbf{v}_{k_2 k_3}} = \frac{\|\mathbf{v}_{k_2 k_1}\|}{\|\mathbf{v}_{k_2 k_3}\|} \cdot e^{i(\widehat{\mathbf{v}_{k_2 k_3}}, \widehat{\mathbf{v}_{k_2 k_1}})}$$

that indicates the location of \mathbf{p}_{k_1} relatively to \mathbf{p}_{k_2} and \mathbf{p}_{k_3} . In this work, we take the real part and the imaginary part of $f(\mathbf{t}_{k_1 k_2 k_3})$ as features.

Appearance-based features

Before extracting appearance features, we cancel the rotation in the image plane and normalize the image using the estimation of the centers of the eyes. Then, we extract HOG descriptors (Histograms of Oriented Gradients) on different patches in the image. Some of them are centered on the landmarks in order to describe local texture and be able to capture expression-related wrinkles, and others are obtained by a 4x4 division of the image (see figure 2), letting us the possibility to catch up for potential point tracking errors. The patches centered using the landmarks we chose are presented figure 3.

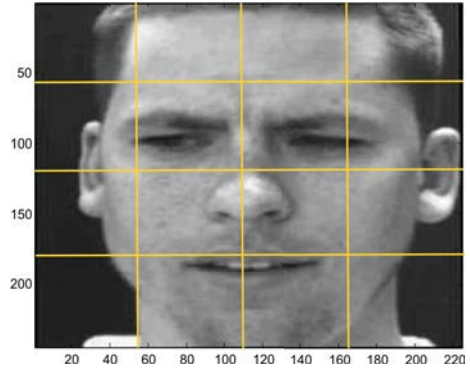


Figure 2: Patches located without the landmarks

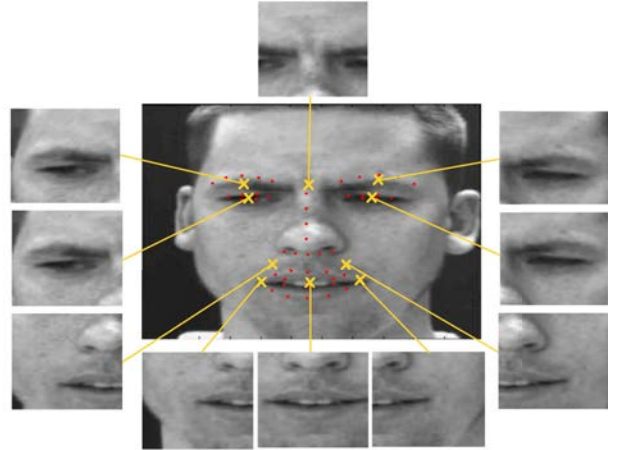


Figure 3: Patches centered using the landmarks

6.2. Databases

We used the Enhanced Cohn-Kanade Dataset for evaluating the different key points of our framework. This database contains prototypical behaviors recorded

in controlled conditions. We compared our algorithm results with state-of-the-art methods using the more natural DISFA Dataset.

Enhanced Cohn-Kanade Dataset

The CK dataset [25] consists of small video sequences in which subjects go from neutral faces to expressive ones. Each sequence is labeled in discrete emotions as well as in FACS. A second version with more sequences (CK+) has been released [30], bringing the number of different subjects up to 123. However, the labels are only available for the last frames of the sequences. The Intelligent Systems Lab of Rensselaar Polytechnic Institute added manual re-labeling of the dataset, frame by frame, and with three different intensity levels for each AU (Enhanced Cohn-Kanade dataset). The different intensity levels are: 0 if the AU is not activated, 1 if it is activated with small intensity, and 2 if it is completely activated. Image samples of the database are presented in figure 4.



Figure 4: Examples of images extracted from the CK+ dataset

DISFA Dataset

The Denver Intensity of Spontaneous Facial Actions (DISFA) dataset [11] contains videos of 27 subjects (12 females and 15 males) with different ethnicities recorded watching a 4-minute emotive video stimulus. Data has been manually labeled frame by frame for 12 AUs on a six-level scale by a human FACS expert, and verified by a second FACS coder. Image samples are presented in figure 5.

6.3. Metrics

Different metrics exist for evaluating the performance of regression systems. In this paragraph, we define three commonly used metrics, namely Root Mean Squared Error (RMSE), Pearson's Correlation Coefficient (CC) and Intraclass Correlation Coefficient (ICC), then we introduce a new metric called r-AUC



Figure 5: Examples of images extracted from the DISFA dataset

and empirically show its advantages over other metrics on two examples.

The most commonly used metric for regression evaluation is RMSE, defined as follows for a label l and an estimated label \hat{l} :

$$RMSE(\hat{l}, l) = \sqrt{\frac{\sum_{i=1}^n (\hat{l}(i) - l(i))^2}{n}} \quad (7)$$

RMSE is often combined with Pearson's Correlation Coefficient (CC) for evaluating the performance of a regression system. CC is defined as follows:

$$CC(\hat{l}, l) = \frac{\sum_{i=1}^n (l(i) - \bar{l})(\hat{l}(i) - \bar{\hat{l}})}{\sqrt{\sum_{i=1}^n (l(i) - \bar{l})^2} \sqrt{\sum_{i=1}^n (\hat{l}(i) - \bar{\hat{l}})^2}} \quad (8)$$

where \bar{l} denotes the mean of the label.

Another commonly used metric is Intraclass Correlation Coefficient (ICC), which, for k judges, is defined as:

$$ICC = \frac{W - S}{W + (k - 1)W} \quad (9)$$

with W the Within-target Mean Squares and S the Residual Sum of Squares. Details of computation can be found in [45].

Data used for learning AUs prediction systems have a particular characteristic: they are highly imbalanced, meaning that there are in many cases very few positive samples compared to zero-valued samples (AU unactivated). In this context, and considering AUs prediction as a classification task, Jeni et al. [46] have investigated different performance measures (accuracy, F1 measure, AUC score...) and concluded that Area Under ROC Curve (AUC) was the most robust and reliable

metric for this task. The Receiver Operating Characteristic (ROC) curve represents the rate of true positives (positive samples correctly detected) as a function of the rate of false positives (negative samples that are incorrectly detected).

In order to take advantage of the robustness of this AUC metric for imbalanced data in the context of regression, we propose a new metric, called regression Area Under ROC Curve (r-AUC), defined as a mean of AUC scores for different binary quantizations of the label. Let us consider a label l varying from 0 to 1. We define a set $\{l_j, j \in \llbracket 1; n_s \rrbracket\}$ of n_s binary quantizations of the label. $l_j(i)$ values 0 if $l(i) < \frac{j}{n_s+1}$ and 1 otherwise. r-AUC corresponds to the mean of the n_s AUCs calculated using the prediction and the different binary quantizations of the label. For a label l and a prediction p , we can explicit r-AUC score in a continuous manner as:

$$r\text{-AUC}(l, p) = \frac{1}{\max(l) - \min(l)} \int_{\min(l)}^{\max(l)} \text{AUC}(p, l_s) ds$$

where l_s is the binary quantization of label l using the threshold s .

Lets us consider two examples in order to illustrate the interest of r-AUC. If the system predicts a linear transformation of the label $\hat{l} = \alpha.l + \beta$, RMSE can be high, even for α close to one and β close to zero. We illustrated this issue in figure 6, where we can notice that the noisy prediction on the lower part leads to a smaller RMSE than the prediction on the upper part. For many applications, this latter prediction would nevertheless be of great value, because it contains all the dynamic information. We can notice that using r-AUC metric, as CC or ICC, the first prediction is evaluated as the most relevant one. Note that a random prediction leads to a r-AUC of 0.5, as random predictions in binary classification lead to a AUC of 0.5.

The Pearson's Correlation Coefficient (CC) lets us consider that linear transformations of the label are accurate predictions. However, in some cases, CC can be misleading. On the upper part of figure 7, the prediction is successful for the four main activations of the AU, but with wrong intensities, and on the lower part, the system only succeeds to predict the most important activation. We can notice that all three metrics (RMSE, CC and ICC) indicate on this example that the second prediction is the best one. The proposed r-AUC metric in this case would evaluate the first prediction more

favorably as more activations are detected.

We believe that this metric lets us overcome important limitations of others standard metrics in the context of imbalanced data. We decided to use it along with CC for evaluating the different parts of our method. We used RMSE and CC for comparing our system's performance with recent state-of-the-art methods, as those metrics were reported on the corresponding papers.

6.4. Experimental setup

All presented results for both datasets correspond to a subject-independent 4-fold cross-validation. All evaluations are performed on a global prediction signal corresponding to the concatenation of the 4 predictions. We extract 22960 features on each frame (19632 geometric features extracted from triplets of points and 3328 appearance features). Our HOG features are extracted with 8 directions on a 4x4 grid for each of the 26 patches. The λ regularization rate optimal value we found is 0.06. We add 10 features at each step of our iterative feature selection strategy, obtaining 70 final selected features for each AU. Our Lasso-MLKR algorithm performs projections on 4-dimensional spaces.

For Enhanced Cohn-Kanade dataset, we used 2600 images for each of the four training folds and for each AU. We selected them in order to have 1300 unactivated samples (corresponding AU of value 0) and 1300 activated samples (randomly selected). The total training for the 14 AUs takes approximately 8 hours on an Intel Core i7-3770 at 3.4 GHz.

For DISFA database, we used 6000 images for each of the four training folds and for each AU. We selected them in order to have 3000 unactivated samples (corresponding AU of value 0) and 3000 activated samples (randomly selected). The total training for the 12 AUs takes approximately 14 hours on an Intel Core i7-3770 at 3.4 GHz.

6.5. Evaluations on CK enhanced

In this section, we evaluate the contribution of the different key points of our framework on Enhanced Cohn-Kanade Dataset (which is annotated using three different intensity levels).

Conditional entropy

In this paragraph, we compare feature selection with conditional entropy similarity measure and Pearson's

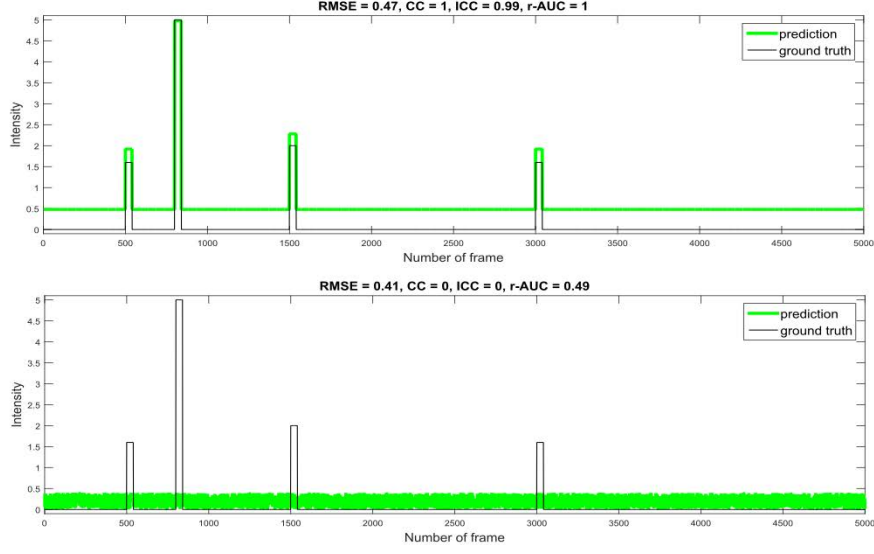


Figure 6: Comparison of different evaluation metrics, first example

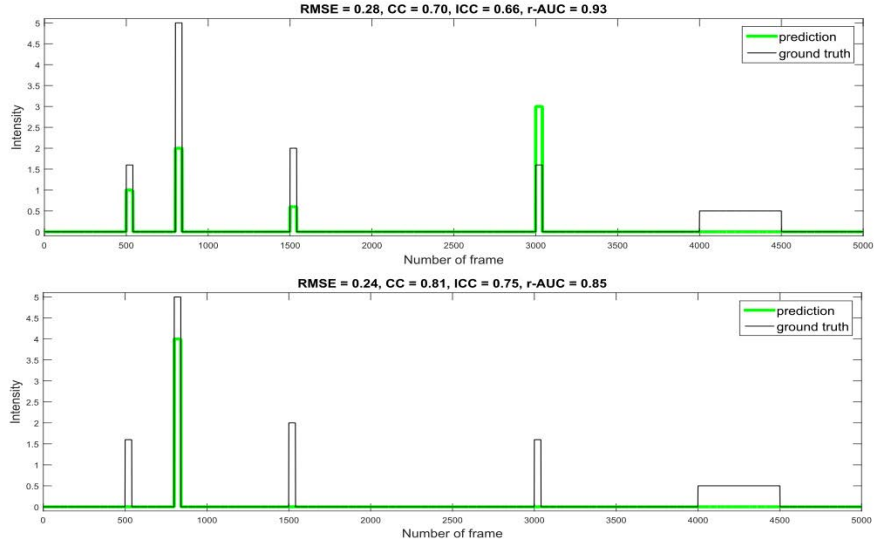


Figure 7: Comparison of different evaluation metrics, second example

Correlation Coefficient on Enhanced CK dataset. We consider the simplest configuration, without iterative feature selection nor regularization of the MLKR formulation. We present the results obtained in terms of CC and r-AUC scores on table 1. We can observe a global improvement of 1.7 % when using conditional entropy metric for feature selection in terms of CC that

is consistent for an important number of AUs (12 of 14 AUs are better predicted). This improvement is significant for several AUs. Main improvements correspond to AU4 (Brow Lowerer), AU5 (Upper Lid Raiser), AU6 (Cheek Raiser), AU7 (Lid Tightener), AU23 (Lip Tightener), AU24 (Lip Pressor) and AU25 (Lips Part). Most of those AUs have the common characteristic

of provoking small landmark displacements, making appearance-based information of primary interest. We can explain those improvements by the important amount of nonlinearities between appearance-based features and labels, making conditional entropy particularly relevant for those AUs.

Table 1: Comparison of Pearson’s Correlation Coefficient (CC) and Conditional Entropy (C-Ent) for feature selection on Enhanced CK dataset

| Evaluation measure | CC (%) | | r-AUC (%) | |
|--------------------|-------------|-------------|-------------|-------------|
| | CC | C-Ent | CC | C-Ent |
| Feature selection | | | | |
| AU1 | 82.4 | 83.1 | 94.4 | 94 |
| AU2 | 89.4 | 87.7 | 96.4 | 96.2 |
| AU4 | 79.4 | 80.8 | 90.7 | 92 |
| AU5 | 68.5 | 73.1 | 91.7 | 92.9 |
| AU6 | 73.9 | 75.2 | 93 | 94.2 |
| AU7 | 66.7 | 67.9 | 89.6 | 89.9 |
| AU9 | 79.7 | 74.6 | 95.3 | 93.8 |
| AU12 | 88.9 | 90 | 96.5 | 97.2 |
| AU15 | 68.7 | 69.7 | 91 | 91.4 |
| AU17 | 73.8 | 74.7 | 92 | 92 |
| AU23 | 50.7 | 53.8 | 88.8 | 90.7 |
| AU24 | 50.4 | 53.2 | 80.2 | 82.9 |
| AU25 | 79.5 | 85.8 | 95.5 | 95.6 |
| AU27 | 92.8 | 92.9 | 99.2 | 99.3 |
| Mean | 74.6 | 75.9 | 92.4 | 93 |

Lasso-MLKR

In this paragraph, we evaluate the contribution of MLKR Lasso-regularization. We consider a configuration with conditional entropy-based feature selection without iterative feature selection. We present the results obtained on table 2. We can observe a global improvement of 2.1 % in terms of CC when adding the regularization that is consistent for an important number of AUs (11 of 14 AUs are better predicted). The regularization, that lets us reduce the overfitting and increase the generalization power of our models, has a significant impact for some AUs, as for AU4 (Brow Lowerer), AU15 (Lip Corner Depressor), AU17 (Chin Raiser), AU23 (Lip Tightener) and AU24 (Lip Pressor). We can observe an important negative correlation between scores without regularization and the gain provided by the regularization, meaning a greater improvement for AUs that are the most difficult to predict. This can be explained by the regularization, which is useful when the training samples are not sufficient to

learn models without overfitting. For AUs having high scores without regularization, the training samples were in sufficient number and contained enough variability for learning models. In those cases, the gain provided by the Lasso-regularization is less important.

Table 2: Comparison of MLKR (M) and Lasso-MLKR (L-M) on Enhanced CK dataset

| Evaluation measure | CC (%) | | r-AUC (%) | |
|--------------------|-------------|-------------|-------------|-------------|
| | M | L-M | M | L-M |
| Algorithm | | | | |
| AU1 | 83.1 | 83.4 | 94 | 94.1 |
| AU2 | 87.7 | 88.5 | 96.2 | 96.5 |
| AU4 | 80.8 | 84.2 | 92 | 93.6 |
| AU5 | 73.1 | 73.8 | 92.9 | 93.5 |
| AU6 | 75.2 | 75 | 94.2 | 93.9 |
| AU7 | 67.9 | 68.8 | 89.9 | 90.7 |
| AU9 | 74.6 | 74.9 | 93.8 | 93.5 |
| AU12 | 90 | 91.2 | 97.2 | 98 |
| AU15 | 69.7 | 71.3 | 91.4 | 92.3 |
| AU17 | 74.7 | 79.7 | 92 | 94.8 |
| AU23 | 53.8 | 57 | 90.7 | 92.3 |
| AU24 | 53.2 | 60.2 | 82.9 | 87.9 |
| AU25 | 85.8 | 84.5 | 95.6 | 95.1 |
| AU27 | 92.9 | 92.6 | 99.3 | 99.4 |
| Mean | 75.9 | 77.5 | 93 | 94 |

Iterative Feature Selection

In this paragraph, we evaluate the contribution of the iterative feature selection framework we propose. We consider a configuration with conditional entropy-based feature selection and Lasso-MLKR. We present in figure 8 the results obtained for CC scores averaged over all 14 AUs. For learning the first model, we selected 10 features, and then we add 10 features at each iteration. We can observe that the models learned using our iterative framework lead to greater CC score at every iteration. In applications where really fast predictions are needed, the number of features has to be restricted in order to save time during kernel computations. The iterative process we propose lets us perform better with the same number of features (for instance, using only 30 features selected in 2 iterations, we see an improvement of 2 % compared to a direct selection of 30 features). This iterative feature selection process leads to a more efficient and compact representation, by avoiding the selection of redundant information.

Those evaluations using the prototypical Enhanced Cohn-Kanade Dataset prove the relevance of the different key points of IRKR: conditional entropy similarity

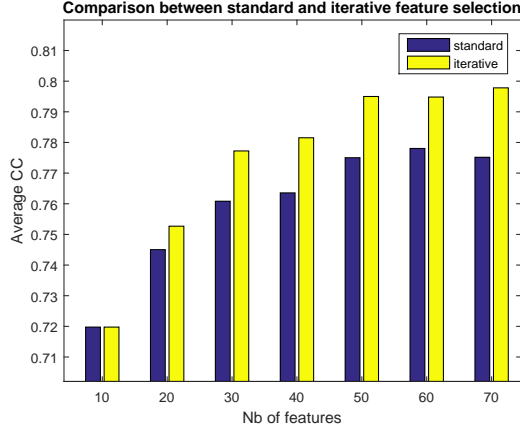


Figure 8: Comparison of standard conditional entropy feature selection and iterative conditional entropy feature selection on Enhanced CK dataset with different number of selected features

metric, our L^1 -regularization of MLKR original formulation and our iterative framework for feature selection.

6.6. Evaluations on DISFA dataset

In this section, we present and compare three versions of IRKR learned with three different sets of features, and compare IRKR with two recent state-of-the-art methods on natural DISFA dataset.

Comparison between different sets of features

In this paragraph, we present the results obtained by learning IRKR only with the geometric features (I1), only with the appearance features (I2) and with the complete set of geometric and appearance features (I3). We present in table 3 the results in terms of CC and r-AUC. In DISFA dataset, AUs are labeled on a six-level scale from 0 (no activation of AU) to 5 (activation with maximal intensity). For continuous signals, r-AUC can be calculated as follows:

$$r\text{-AUC}(l, p) = \frac{1}{\max(l) - \min(l)} \int_{\min(l)}^{\max(l)} AUC(p, l_s) ds$$

where l_s is the binary quantization of label l using the threshold s . For the six-level labels of DISFA, by considering the following vector of thresholds

$$\mathbf{v} = \{0.5, 1.5, 2.5, 3.5, 4.5\}$$

r-AUC can be simplified as:

$$r\text{-AUC}(l, p) = \frac{1}{5} \sum_{i=1}^5 AUC(p, l_{v_i})$$

which corresponds to an average of 5 AUC scores for the different thresholds (the first one separating the samples of values 0 from the others, the second one separating the samples of values 0 and 1 from the others, and so on).

Table 3: Comparison between three versions of IRKR on the DISFA dataset. I1 corresponds to a model learned only using shape features, I2 to a model learned with only appearance features and I3 to a model learned with both shape and appearance features

| Evaluation measure Feature set | CC (%) | | | r-AUC (%) | | |
|-----------------------------------|-----------|-----------|-----------|-----------|----|-----------|
| | I1 | I2 | I3 | I1 | I2 | I3 |
| AU1 | 57 | 60 | 70 | 91 | 91 | 94 |
| AU2 | 61 | 60 | 68 | 94 | 93 | 94 |
| AU4 | 54 | 61 | 68 | 87 | 89 | 91 |
| AU5 | 36 | 39 | 49 | 77 | 96 | 98 |
| AU6 | 63 | 58 | 65 | 93 | 91 | 94 |
| AU9 | 40 | 34 | 43 | 90 | 88 | 92 |
| AU12 | 83 | 75 | 83 | 96 | 94 | 96 |
| AU15 | 19 | 33 | 34 | 74 | 77 | 83 |
| AU17 | 29 | 26 | 35 | 79 | 82 | 86 |
| AU20 | 10 | 25 | 21 | 65 | 68 | 74 |
| AU25 | 87 | 78 | 86 | 94 | 90 | 94 |
| AU26 | 55 | 37 | 62 | 90 | 88 | 92 |
| Mean | 50 | 49 | 57 | 86 | 87 | 91 |

We observe a gain of 14% of the average CC score when adding appearance features to geometric ones. We can see that for AU12 (Lip Corner Puller) and AU25 (Lips Part), appearance features did not improve the prediction. Geometric features were sufficient to obtain relatively precise predictions for those AUs. However, for more subtle AUs inducing smaller facial movements, appearance features have been considerably improving the predictions as for AU5 (Upper Lid Raiser), AU15 (Lip Corner Depressor), AU17 (Chin Raiser) and AU20 (Lip Stretcher).

In the next paragraph, for comparing our method, we use the complete version of IRKR (I3), that includes both geometric and appearance features.

Comparison with state-of-the-art methods

IRKR is compared to the method proposed by Sandbach et al. [47] and the one proposed by Baltrušaitis et al. [48] in terms of Root Mean Square Error (RMSE) and Correlation Coefficient (CC) in table 4. In [47], the authors used Support Vector Regression on Local Binary Pattern (LBP) features and included priors via

Markov Random Fields (MRF). In [48], Continuous Conditional Neural Fields (CCNF) are used after modeling the appearance with Non-negative Matrix Factorization (NMF) on local patches. In [47], only the AUs corresponding to the upper face have been predicted. For the upper face AUs, the average RMSE of IRKR is 0.576 compared to 0.66 for [47]. The average CC of IRKR is 60.35 compared to 34.2 for [47]. We obtain a mean RMSE on all 12 AUs of 0.59 compared to 0.66 obtain by [48] and a mean CC of 57 compared to 49 for [48].

We can notice, for AU9, that the RMSE error of [47] is lower than IRKR's, but that the CC score of IRKR is higher. This contradiction illustrates the metric problematic we discussed in Section 6.3. We proposed r-AUC score to overcome this issue. In figure 9, we present the 5 AUC scores we obtain for the different thresholds as well as the average r-AUC for each AU. For most AUs, we can see that the AUC scores corresponding to the low thresholds are lower than the AUC scores for higher thresholds. It means that the algorithm succeeds more easily to separate high-intensity activations from the others, and has more difficulties for separating the non-activated from the rest. If we consider that an AU is activated when its intensity is equal or higher than level 3 (corresponding to the third threshold), 8 of 12 AUs are predicted with an AUC score higher than 0.9 (excepting AU9, AU15, AU17 and AU20).

On figure 10, we show the prediction of IRKR algorithm on a part of the third sequence of DISFA dataset for AU4 (Brow-Lowerer). We can observe that the algorithm succeeds well to predict two brow lowering actions in the middle and in the end of the sequence. In those actions, the intensity reaches level 4. For the beginning of the sequence, our algorithm succeeds to differentiate level 3 from non-activation but with a certain amount of noise. The small activation reaching level 2 around frame 1450 is not predicted by our system. This example illustrates the AUC scores of AU4 on figure 9. It is more difficult to differentiate activations of level 0, 1 and 2, but recognition is relevant from level 3 (AUC score is higher than 0.9 for those thresholds).

The obtained results show the relevance of our regression method for AUs intensity prediction. Our Matlab implementation of IRKR algorithm predicts 16 frames by second on an Intel Core i7-3770 at 3.4 GHz, making it usable in real-time applications.

7. Discussion and conclusion

In this paper, we presented the Iterative Regularized Kernel Regression (IRKR) framework, a generic regression method built upon Metric Learning for Kernel Regression (MLKR) [37]. We applied it to real-time prediction of AUs intensity, improving state-of-the-art results on several databases. In this work, we propose a L^1 -regularization of the original MLKR formulation in order to reduce overfitting. We use conditional entropy for selecting features with nonlinear functional relationships with labels. Then, we perform an iterative framework in order to avoid selecting redundant information. Finally, we introduce r-AUC, a new evaluation metric for regression in the context of imbalanced data.

We evaluated and compared our method using two AUs databases containing multi-levels annotations. The first one, enhanced Cohn-Kanade dataset, is a widely-used prototypical database upon which we evaluated the different key points of our method. We compared IRKR with state-of-art methods on the natural DISFA dataset on 12 AUs, letting to mean improvements of 10.3% and 11.6% for Root Mean Squared Error (RMSE) and Correlation Coefficient (CC) respectively.

The most accurate predictions were obtained for AU1 (Inner Brow Raiser), AU2 (Outer Brow Raiser), AU4 (Brow Lowerer), AU12 (Lip Corner Puller) and AU25 (Lips Part) that are frequently activated in natural behaviors. Other AUs appear to be more complex to model and predict. This can be explained by the small number of positive samples for some AUs in natural databases. Indeed, some AUs are only activated in particular and rare emotional states, which can be difficult to induce in natural setups when acquiring data. Considering this imbalanced data distribution, it is particularly important to focus on overfitting reduction, which is the purpose of the Lasso-penalty we added to the original cost function of MLKR, that lets to important gains, especially for those complex AUs.

However, the results show that for Lip Corner Depressor (AU15) and Lip Stretcher (AU20), the number of positive samples may not be sufficient for modeling the activations accurately. The amount of labeled data available still remains a brake on AUs intensity prediction and natural protocols inducing rare facial expressions may be very important for continuing increasing accuracy in face-centered human behavior automatic analysis.

The comparison between IRKR learned only with geometric features and IRKR learned with geometric and appearance features stresses the importance of appearance characterization for AUs intensity prediction. Re-

Table 4: Comparison between our algorithm and the ones proposed by Sandbach et al. [47] and Baltrušaitis [48] on the DISFA dataset in terms of Root Mean Square Error (R) and Correlation Coefficient (CC)

| AU | R, IRKR | R, [47] | R, [48] | CC, IRKR | CC, [47] | CC, [48] |
|------|-------------|-------------|-------------|-------------|----------|-----------|
| 1 | 0.57 | 0.63 | 0.74 | 69.7 | 56.3 | 48 |
| 2 | 0.49 | 0.58 | 0.63 | 68.2 | 54.1 | 50 |
| 4 | 0.85 | 1.10 | 1.13 | 67.7 | 43.8 | 52 |
| 5 | 0.29 | 0.30 | 0.33 | 49.2 | 22.6 | 48 |
| 6 | 0.63 | 0.77 | 0.75 | 64.7 | 11.9 | 45 |
| 9 | 0.62 | 0.58 | 0.67 | 42.6 | 16.8 | 36 |
| 12 | 0.58 | - | 0.71 | 83.2 | - | 70 |
| 15 | 0.47 | - | 0.46 | 34.2 | - | 41 |
| 17 | 0.62 | - | 0.67 | 35 | - | 39 |
| 20 | 0.59 | - | 0.58 | 21.0 | - | 11 |
| 25 | 0.69 | - | 0.63 | 86.2 | - | 89 |
| 26 | 0.69 | - | 0.63 | 61.5 | - | 57 |
| Mean | 0.59 | - | 0.66 | 56.9 | - | 49 |

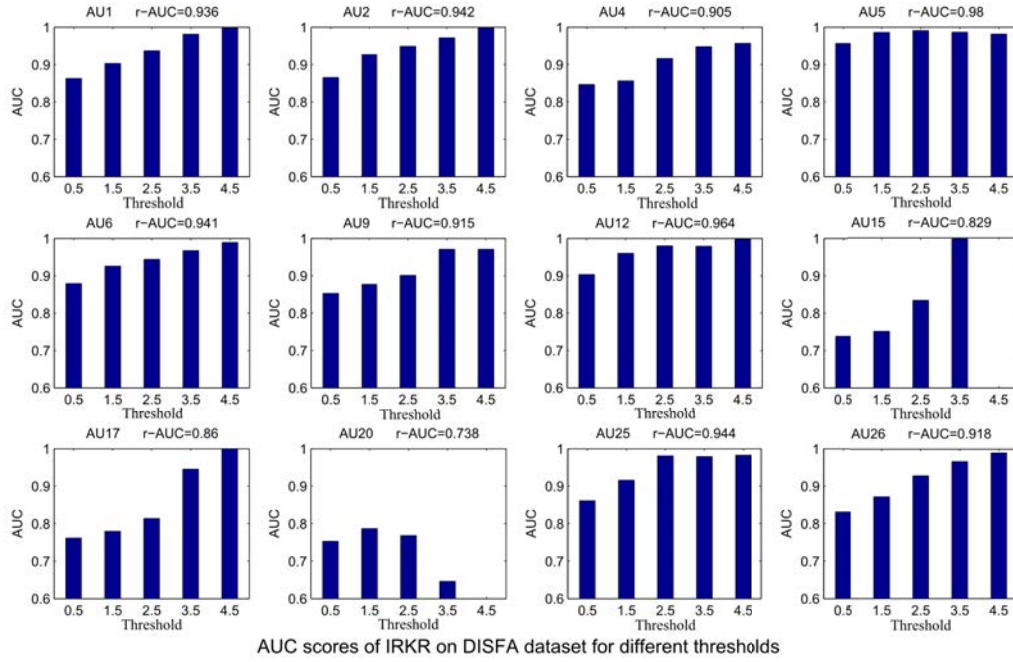


Figure 9: AUC scores of IRKR on DISFA dataset for different thresholds. There is only 4 thresholds for AU15 and AU20, because the database does not include samples of intensity 5 for those AUs.

cent advances in facial landmarks tracking considerably improves AUs prediction scores but research on appearance features stays of great interest for this domain as pointed out by those results. Although the relationship between facial landmarks and AUs activations have

an important chance of being close to linear, it is not the same for appearance features. Thus, it is important to model those relations in a nonlinear way. This is why we decided to use the nonlinear conditional entropy metric for selecting features. The obtained results

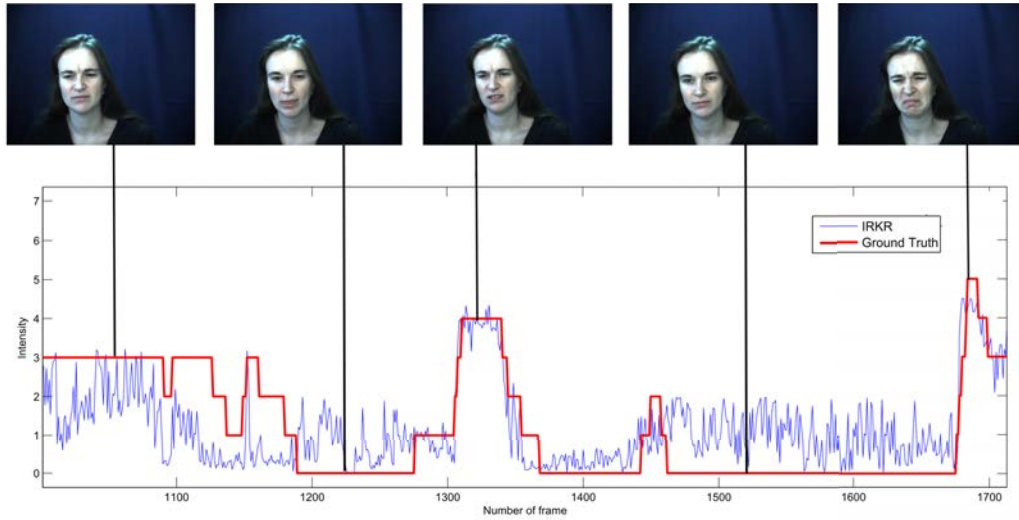


Figure 10: Prediction of AU4 on a part of sequence 3 of DISFA dataset

show that using this metric compared to Correlation Coefficient improves predictions of AUs that are linked to appearance characterizations, as AU4 (Brow Lowerer), AU5 (Upper Lid Raiser), AU6 (Cheek Raiser), AU7 (Lid Tightener), AU23 (Lip Tightener) and AU24 (Lip Pressor).

We used this metric within an iterative framework for feature selection in order to avoid selecting redundant information. It leads to a more compact representation, obtaining higher scores with a reduced set of features. This compact representation can be interesting for several reasons. First, reducing the number of parameters in the model can reduce overfitting, and second, compact representations lead to faster predictions, which is useful considering the real-time constraints of many applications related to AUs automatic prediction.

Several metrics exist for evaluating a regression method. The most commonly used in AUs intensity prediction being Root Mean Square Error (RMSE) and Correlation Coefficient (CC). However, some issues occur with those metrics for imbalanced data. For solving those issues, we propose r-AUC, an adaptation of Area Under ROC Curve (AUC) suited for regression problems in an imbalanced context.

The results obtained on the natural DISFA dataset are very promising, especially for the most frequently activated facial muscles. However, AUs intensity prediction is a particularly difficult task and many improvements could still be made, for instance by proposing multi-tasks methods including other information such as age or head pose, playing a crucial role in face appearance

deformations, or by thinking about new database acquisition protocols inducing natural rare facial expressions.

- [1] P. Ekman, W. V. Friesen, Measuring facial movement, *Environmental Psychology and Nonverbal Behavior* 1 (1) (1976) 56–75.
- [2] R. El Kaliouby, P. Robinson, Real-time inference of complex mental states from facial expressions and head gestures, in: *Real-time vision for human-computer interaction*, Springer, 2005, pp. 181–200.
- [3] J. F. Cohn, T. S. Kruez, I. Matthews, Y. Yang, M. H. Nguyen, M. T. Padilla, F. Zhou, F. De la Torre, Detecting depression from facial actions and vocal prosody, in: *Affective Computing and Intelligent Interaction and Workshops, 2009. ACHI 2009. 3rd International Conference on*, IEEE, 2009, pp. 1–7.
- [4] A. B. Ashraf, S. Lucey, J. F. Cohn, T. Chen, Z. Ambadar, K. M. Prkachin, P. E. Solomon, The painful face—pain expression recognition using active appearance models, *Image and Vision Computing* 27 (12) (2009) 1788–1796.
- [5] S. Kaltwang, O. Rudovic, M. Pantic, Continuous pain intensity estimation from facial expressions, in: *Advances in Visual Computing*, Springer, 2012, pp. 368–377.
- [6] D. McDuff, R. El Kaliouby, T. Senechal, M. Amr, J. F. Cohn, R. Picard, Affectiva-mit facial expression dataset (am-fed): Naturalistic and spontaneous facial expressions collected “in-the-wild”, in: *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*, IEEE, 2013, pp. 881–888.
- [7] B. Zaman, T. Shrimpton-Smith, The facereader: Measuring instant fun of use, in: *Proceedings of the 4th Nordic conference on Human-computer interaction: changing roles*, ACM, 2006, pp. 457–460.
- [8] T. Pfister, X. Li, G. Zhao, M. Pietikainen, Recognising spontaneous facial micro-expressions, in: *Computer Vision (ICCV), 2011 IEEE International Conference on*, IEEE, 2011, pp. 1449–1456.
- [9] M. E. Hoque, D. J. McDuff, R. W. Picard, Exploring temporal patterns in classifying frustrated and delighted smiles, *Affective Computing, IEEE Transactions on* 3 (3) (2012) 323–334.
- [10] Y. Tong, W. Liao, Q. Ji, Facial action unit recognition by exploit-

- ing their dynamic and semantic relationships, *Pattern Analysis and Machine Intelligence*, IEEE Transactions on 29 (10) (2007) 1683–1699.
- [11] S. M. Mavadati, M. H. Mahoor, K. Bartlett, P. Trinh, J. F. Cohn, Disfa: A spontaneous facial action intensity database, *IEEE Transaction on Affective Computing* 4 (2) (2013) 151–160.
 - [12] Y. Sun, M. Reale, L. Yin, Recognizing partial facial action units based on 3d dynamic range data for facial expression recognition, in: *Automatic Face & Gesture Recognition*, 2008. FG'08. 8th IEEE International Conference on, IEEE, 2008, pp. 1–8.
 - [13] A. Savran, B. Sankur, M. Taha Bilge, Regression-based intensity estimation of facial action units, *Image and Vision Computing* 30 (10) (2012) 774–784.
 - [14] Y. Li, J. Chen, Y. Zhao, Q. Ji, Data-free prior model for facial action unit recognition, *Affective Computing*, IEEE Transactions on 4 (2) (2013) 127–141.
 - [15] S. Wan, J. Aggarwal, Spontaneous facial expression recognition: A robust metric learning approach, *Pattern Recognition*.
 - [16] L. A. Jeni, J. M. Girard, J. F. Cohn, F. De La Torre, Continuous au intensity estimation using localized, sparse facial feature space, in: *Automatic Face and Gesture Recognition (FG)*, 2013 10th IEEE International Conference and Workshops on, IEEE, 2013, pp. 1–7.
 - [17] W.-S. Chu, F. De la Torre, J. F. Cohn, Selective transfer machine for personalized facial action unit detection, in: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.
 - [18] P. Yang, Q. Liu, D. N. Metaxas, Boosting coded dynamic features for facial action units and facial expression recognition, in: *Computer Vision and Pattern Recognition*, 2007. CVPR'07. IEEE Conference on, IEEE, 2007, pp. 1–6.
 - [19] C.-F. Chuang, F. Y. Shih, Recognizing facial action units using independent component analysis and support vector machine, *Pattern recognition* 39 (9) (2006) 1795–1798.
 - [20] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, in: *Computer Vision and Pattern Recognition*, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, Vol. 1, IEEE, 2001, pp. 1–511.
 - [21] Y. Freund, R. E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, in: *Computational learning theory*, Springer, 1995, pp. 23–37.
 - [22] T. Senechal, V. Rapp, H. Salam, R. Segulier, K. Bailly, L. Prevost, Combining aam coefficients with lgbp histograms in the multi-kernel svm framework to detect facial action units, in: *Automatic Face & Gesture Recognition and Workshops (FG 2011)*, 2011 IEEE International Conference on, IEEE, 2011, pp. 860–865.
 - [23] O. Rudovic, V. Pavlovic, M. Pantic, Kernel conditional ordinal random fields for temporal segmentation of facial action units, in: *Computer Vision–ECCV 2012. Workshops and Demonstrations*, Springer, 2012, pp. 260–269.
 - [24] G. Littlewort, M. S. Bartlett, I. Fasel, J. Susskind, J. Movellan, Dynamics of facial expression extracted automatically from video, *Image and Vision Computing* 24 (6) (2006) 615–625.
 - [25] T. Kanade, J. Cohn, Y. Tian, Comprehensive database for facial expression analysis, in: *Automatic Face and Gesture Recognition*, 2000. Proceedings. Fourth IEEE International Conference on, 2000, pp. 46–53.
 - [26] T. Sim, S. Baker, M. Bsat, The cmu pose, illumination, and expression database, *Pattern Analysis and Machine Intelligence*, IEEE Transactions on 25 (12) (2003) 1615–1618.
 - [27] M. F. Valstar, B. Jiang, M. Mehu, M. Pantic, K. Scherer, The first facial expression recognition and analysis challenge, in: *Automatic Face & Gesture Recognition and Workshops (FG 2011)*, 2011 IEEE International Conference on, IEEE, 2011, pp. 921–926.
 - [28] M. F. Valstar, M. Pantic, Fully automatic recognition of the temporal phases of facial actions, *Systems, Man, and Cybernetics, Part B: Cybernetics*, IEEE Transactions on 42 (1) (2012) 28–43.
 - [29] A. Savran, N. Alyüz, H. Dibeklioglu, O. Çeliktutan, B. Gökberk, B. Sankur, L. Akarun, Bosphorus database for 3d face analysis, in: *Biometrics and Identity Management*, Springer, 2008, pp. 47–56.
 - [30] P. Lucey, J. Cohn, T. Kanade, J. Saragih, Z. Ambadar, I. Matthews, The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression, in: *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2010 IEEE Computer Society Conference on, 2010, pp. 94–101.
 - [31] P. Lucey, J. F. Cohn, K. M. Prkachin, P. E. Solomon, I. Matthews, Painful data: The unbc-mcmaster shoulder pain expression archive database, in: *Automatic Face & Gesture Recognition and Workshops (FG 2011)*, 2011 IEEE International Conference on, IEEE, 2011, pp. 57–64.
 - [32] J. M. Girard, J. F. Cohn, F. De la Torre, Estimating smile intensity: A better way, *Pattern Recognition Letters*.
 - [33] P. Ekman, An argument for basic emotions, *Cognition & Emotion* 6 (3–4) (1992) 169–200.
 - [34] H. Takeda, S. Farsiu, P. Milanfar, Deblurring using regularized locally adaptive kernel regression, *Image Processing*, IEEE Transactions on 17 (4) (2008) 550–563.
 - [35] M. Schaap, L. Neefjes, C. Metz, A. van der Giessen, A. Weustink, N. Mollet, J. Wentzel, T. van Walsum, W. Niessen, Coronary lumen segmentation using graph cuts and robust kernel regression, in: *Information Processing in Medical Imaging*, Springer, 2009, pp. 528–539.
 - [36] J. Nicolle, V. Rapp, K. Bailly, L. Prevost, M. Chetouani, Robust continuous prediction of human emotions using multiscale dynamic cues, in: *Proceedings of the 14th ACM international conference on Multimodal interaction*, ACM, 2012, pp. 501–508.
 - [37] K. Q. Weinberger, G. Tesauro, Metric learning for kernel regression, in: *International Conference on Artificial Intelligence and Statistics*, 2007, pp. 612–619.
 - [38] E. A. Nadaraya, On estimating regression, *Theory of Probability & Its Applications* 9 (1) (1964) 141–142.
 - [39] L. Yu, H. Liu, Feature selection for high-dimensional data: A fast correlation-based filter solution, in: *ICML*, Vol. 3, 2003, pp. 856–863.
 - [40] R. Tibshirani, Regression shrinkage and selection via the lasso, *Journal of the Royal Statistical Society. Series B (Methodological)* (1996) 267–288.
 - [41] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, *The Journal of Machine Learning Research* 3 (2003) 1157–1182.
 - [42] X. Xiong, F. De la Torre, Supervised descent method and its applications to face alignment, in: *Computer Vision and Pattern Recognition (CVPR)*, 2013 IEEE Conference on, 2013, pp. 532–539.
 - [43] J. Nicolle, K. Bailly, V. Rapp, M. Chetouani, Locating facial landmarks with binary map cross-correlations., in: *ICIP*, 2013, pp. 2978–2982.
 - [44] G. Murthy, R. Jadon, Effectiveness of eigenspaces for facial expressions recognition, *International Journal of Computer Theory and Engineering* 1 (5) (2009) 1793–8201.
 - [45] P. E. Shrout, J. L. Fleiss, Intraclass correlations: uses in assessing rater reliability., *Psychological bulletin* 86 (2) (1979) 420.
 - [46] L. A. Jeni, J. F. Cohn, F. De La Torre, Facing imbalanced data: recommendations for the use of performance metrics, in: *Affective Computing and Intelligent Interaction (ACII)*, 2013 Humaine Association Conference on, IEEE, 2013, pp. 245–251.

- [47] G. Sandbach, S. Zafeiriou, M. Pantic, Markov random field structures for facial action unit intensity estimation, in: Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on, IEEE, 2013, pp. 738–745.
- [48] T. Baltrušaitis, P. Robinson, L.-P. Morency, Continuous conditional neural fields for structured regression, in: Computer Vision–ECCV 2014, Springer, 2014, pp. 593–608.

Appendix B

Emotion Prediction in a Continuous Space

This appendix contains a work on emotion prediction that is based on the Nadaraya-Watson regressor. It has been published in IEEE International Conference on Multimodal Interaction (ICMI 2012).

Robust Continuous Prediction of Human Emotions using Multiscale Dynamic Cues

Jérémie Nicolle^{*}
Univ. Pierre & Marie Curie
ISIR - CNRS UMR 7222
F-75005, Paris - France
nicolle@isir.upmc.fr

Vincent Rapp^{*}
Univ. Pierre & Marie Curie
ISIR - CNRS UMR 7222
F-75005, Paris - France
rapp@isir.upmc.fr

Kévin Bailly
Univ. Pierre & Marie Curie
ISIR - CNRS UMR 7222
F-75005, Paris - France
bailly@isir.upmc.fr

Lionel Prevost
Univ. of French West Indies
& Guiana - LAMIA - EA 4540
Guadeloupe - France
lionel.prevost@univ-ag.fr

Mohamed Chetouani
Univ. Pierre & Marie Curie
ISIR - CNRS UMR 7222
F-75005, Paris - France
chetouani@isir.upmc.fr

ABSTRACT

Designing systems able to interact with humans in a natural manner is a complex and far from solved problem. A key aspect of natural interaction is the ability to understand and appropriately respond to human emotions. This paper details our response to the Audio/Visual Emotion Challenge (AVEC'12) whose goal is to continuously predict four affective signals describing human emotions (namely valence, arousal, expectancy and power). The proposed method uses log-magnitude Fourier spectra to extract multiscale dynamic descriptions of signals characterizing global and local face appearance as well as head movements and voice. We perform a kernel regression with very few representative samples selected via a supervised weighted-distance-based clustering, that leads to a high generalization power. For selecting features, we introduce a new correlation-based measure that takes into account a possible delay between the labels and the data and significantly increases robustness. We also propose a particularly fast regressor-level fusion framework to merge systems based on different modalities. Experiments have proven the efficiency of each key point of the proposed method and we obtain very promising results.

Categories and Subject Descriptors

H.4 [Information Systems Applications]: Miscellaneous;
D.2.8 [Software Engineering]: Metrics—*complexity measures, performance measures*

^{*}These authors equally contributed to this work.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ICMI'12, October 22–26, 2012, Santa Monica, California, USA.
Copyright 2012 ACM 978-1-4503-1467-1/12/10 ...\$15.00.

Keywords

Affective computing, Dynamic features, Multimodal fusion, Feature selection, Facial expressions

1. INTRODUCTION

In Human-Computer Intelligent Interaction systems, a current challenge is to give the computer the ability to interact naturally with the user with some kind of emotional intelligence. Interactive systems should be able to perceive pain, stress or inattention and to adapt and respond to these affective states, or, in other words, to interact with humans vocally and visually in a natural way. An essential step towards this goal is the acquisition, interpretation and integration of human affective state within the Human-Machine communication system. To recognize affective states, human-centered interfaces should interpret various social cues from both audio and video modalities, mainly linguistic messages, prosody, body language, eye contact and facial expressions.

Automatic recognition of human emotions from both modalities has been an active field of research over the last decade. Most of the proposed systems have focused on the recognition of acted or prototypal emotions recorded in a constrained environment and leading to high recognition rates. These systems usually describe affects via a prototypal modeling approach using the six basic emotions introduced in the early 70s by Ekman [3]. Another standard way to describe facial expressions is to analyze the set of muscles movements produced by a subject. These movements are called facial Action Units (AUs) and the corresponding code is the Facial Action Coding System (FACS) [4]. The first challenge on Facial Expression Recognition and Analysis (FERA'11) focused on these two kinds of affect description. Meta-analysis of challenge results are summarized in [21]. These methods generally use discrete systems whether based on static descriptors (geometrical or appearance features) and/or on static classifiers such as Support Vector Machines [20].

However, these descriptions do not reflect real-life interac-

tions and the resulted systems can be irrelevant to an everyday interaction where people may display subtle and complex affective states. To take this complexity into account, this classical description via prototypal modeling approach has recently evolved to a dimensional approach where emotions are described continuously within an affect space. The choice of the dimensions of this space remains an open question but Fontaine & al. [5] showed that four dimensions cover the majority of affective variability: Valence (positivity or negativity), Arousal (activity), Expectancy (anticipation) and Power (control). The Affective Computing research community has recently focused on the area of dimensional emotion prediction and the first workshop on this topic (EmoSPACE'11, [7]) was organized last year, followed by the first Audio/Visual Emotion Challenge (AVEC'11 [19]).

Usually, the most important parts of multimodal emotion recognition systems are the learning database, the extracted features, the predictor and the fusion method. More precisely, one of the main key points concerns the features' semantic level. Some methods use low-level features. For example, Wollmer et al. [23] propose an approach using features based on the optical flow. Dahmane et al. [2] use Gabor filter energies to compute their visual features. Ramirez et al. [15], conversely, prefer to extract high-level features such as gaze direction, head tilt or smile intensity. Similarly, Gunes et al. [6] focus on spontaneous head movements.

Another key aspect of this new dimensional approach is the need for the system to take the dynamic of human emotions into account. Some methods propose to directly encode dynamic information in the features. For example, Jiang et al. [8] extend the purely spatial representation LPQ to a dynamic texture descriptor called Local Phase Quantisation from Three Orthogonal Planes (LPQ-TOP). Cruz et al. [1] propose an approach that aligns the faces with Avatar Image Registration, and subsequently compute LPQ features. McDuff et al. [9] predict valence using facial Action Unit spectrograms as features. In this study, we focus on mid-level dynamic features, extracted using different visual cues: head movements, face deformations and also global and local face appearance variations. Most methods use visual cues directly as features. In our method, dynamic information is included by computing the log-magnitude Fourier spectra of the temporal signals that describe the evolution of the previously introduced visual cues. Since an accurate and robust system should take advantage of interpreting signals from various modalities, we also include audio features to bring complementary information.

For the prediction step, different machine learning algorithms can be used. Several methods are based on context-dependent frameworks. For example, Meng et al. [11] propose a system based on Hidden Markov Models. Wollmer et al. [23] investigate a more advanced technique based on context modeling using Long Short-Term Memory neural networks. These systems provide the advantage to encode dynamics within the learning algorithm. Another solution is to base the system on a static predictor as, for instance, the well-known Support Vector Machine [1, 17]. Dynamic information being already included in our features, we chose a static predictor. The proposed method uses a kernel regressor based on the Nadaraya-Watson estimator [12]. For selecting representative samples, we perform a clustering step in a space of preselected relevant features.

To merge all visual and vocal information, various fusion

strategies may be relevant. Feature-level fusion (also called early fusion) can be performed by merging extracted features from each modality into one cumulative structure and feeding it to a single classifier. This technique is appropriate for synchronized modalities but some issues may appear for unsynchronized or heterogeneous features. Another solution is decision-level fusion (or late fusion); each extracted feature set feeds one classifier and all the classifier outputs are merged to provide the final response. For example, Nicolaou et al [14] propose an output-associative fusion framework. In our case, the fusion is based on a simple method linearly combining outputs corresponding to the predictions of the four dimensions with different systems to make the final predictions. This way, the system is able to capture the correlations between the different emotion dimensions and to increase robustness by using different modalities.

The designed system is our response to the second Audio/Visual Emotion Challenge (AVEC'12) [18]. This challenge uses the SEMAINE [10] corpus as benchmarking database. Concerning SEMAINE, as Nicolaou et al. [13], we noticed some annotation issues which may directly impact the system performance. This database has been continuously annotated by humans in real-time and a delay between the affect events and the labels has thus been introduced. To avoid this issue, we present in this paper a delay probability estimation method directly used in the feature selection process.

The main contributions presented in this paper for affective signals prediction are the followings:

1. The use of the log-magnitude Fourier spectrum to include dynamic information for human emotions prediction.
2. A new correlation-based measure for the feature selection process that increases robustness to possibly time-delayed labels.
3. A fast efficient framework for regression and fusion designed for real-time implementation.

The proposed framework, presented in Fig. 1, is based on audio-visual dynamic information detailed in section 2. As visual cues, we propose a set of features based on facial shape deformations, and two sets respectively based on global and local face appearance. For each visual cue, we obtain a set of temporal signals and encode their dynamic using log-magnitude Fourier spectra. Audio information is added using the provided audio features. Regarding the prediction, we propose a method based on independent systems for each set of features and for each dimension (section 3). For each system, a new correlation-based feature selection is performed using a delay probability estimator. This process is particularly well-adapted to unsure and possibly time-delayed labels. The prediction is then done by a non-parametric regression using representative samples selected via a k-means clustering process. We finally linearly combine the 16 outputs during a fusion process to take into account dependencies between each modality and each affective dimension (section 4). Section 5 is dedicated to evaluation and analysis. Finally, conclusion and future work are presented in section 7.

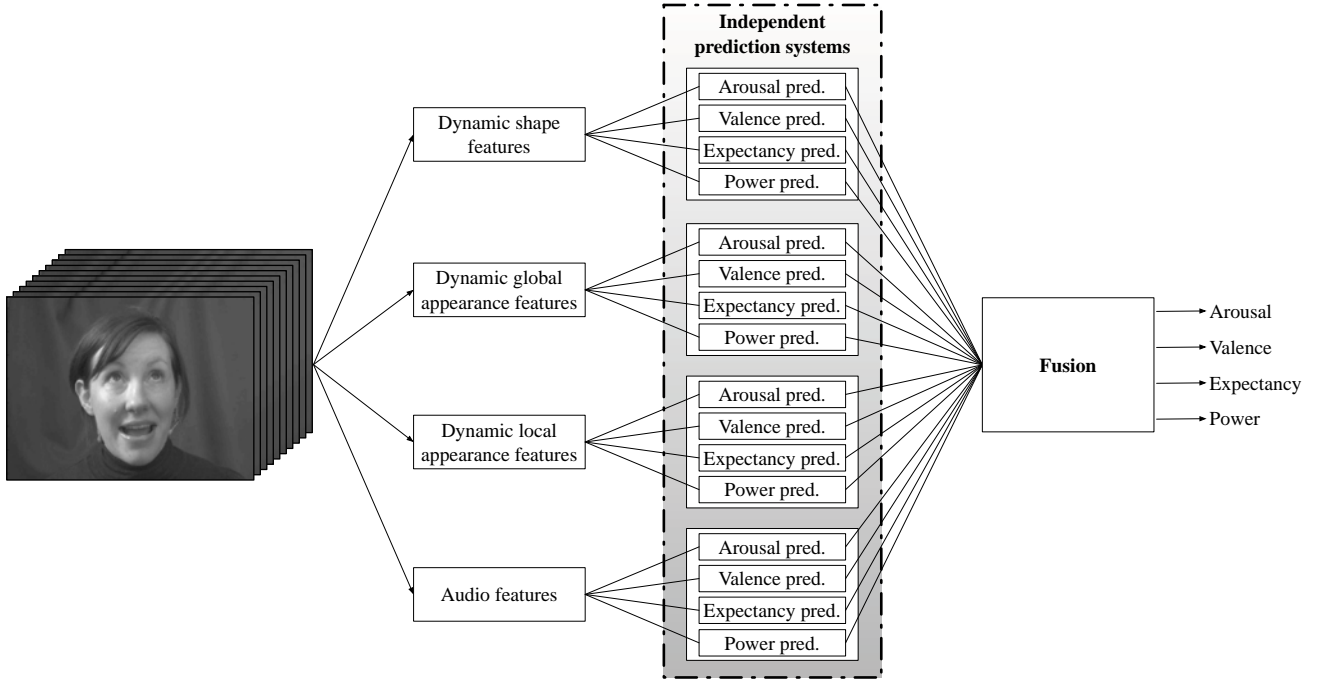


Figure 1: Overview of the proposed framework.

2. FEATURES

In this section, we present the four different sets of features we used. We propose three multiscale dynamic feature sets based on video; the fourth one is based on audio.

For the sets of visual cues, we first extract temporal signals describing the evolution of facial shape and appearance movements before calculating multiscale dynamic features on these signals. The feature extraction process is described in Fig. 2.

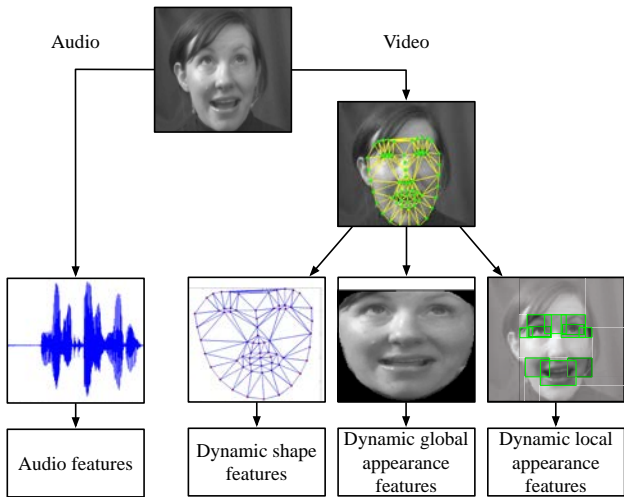


Figure 2: Feature extraction overview.

2.1 Signals extraction

We extract three kinds of signals: one based on shape parameters, and two others based on global and local face appearance.

1. Shape parameters:

The first set of features we used is based on face deformation shape parameters. The initial step of this feature extraction process is the use of the 3D face tracker proposed in [16]. It detects the face area in the images with a Viola-Jones algorithm [22] before estimating the relative position of 66 landmarks using a Point Distribution Model (PDM). The position of the i^{th} landmark \mathbf{s}_i in the image can be expressed as:

$$\mathbf{s}_i(\mathbf{p}) = s\mathbf{R}(\bar{\mathbf{s}}_i + \Phi_i\mathbf{q}) + \mathbf{t} \quad (1)$$

where $\bar{\mathbf{s}}_i$ denotes the mean location of the i^{th} landmark and Φ_i the principal subspace matrix computed from training shape samples using principal component analysis (PCA). Here, $\mathbf{p} = \{s, \mathbf{R}, \mathbf{t}, \mathbf{q}\}$ denotes the PDM parameters, which consist of global scaling s , rotation \mathbf{R} and translation \mathbf{t} . Vector \mathbf{q} represents the deformation parameters that describe the deformation of \mathbf{s}_i along each principal direction.

As output of this system, we obtain temporal signals: some of them correspond to the external parameters and give information on the head position, and the others characterize deformations related to facial expressions.

2. Global appearance:

The second set of features we used is based on global face appearance. First, we warp the faces into a mean model using the point locations obtained with the face tracker. This way, the global appearance will be less sensitive to shape variations and head movements, already encoded in the first set. Then, we select the most important modes of appearance variations using PCA. We obtain a set of temporal signals by projecting the warped images on the principal modes.

3. Local appearance:

The third set is based on local face appearance. First, we extract local patches of possibly interesting areas regarding deformations related to facial expressions. We extract an area around the mouth in order to capture smiles, areas around the eyes to capture the gaze direction, around the eyebrows to capture their movements, and areas where the most common expression-related lines can appear (periorbital lines, glabellar lines, nasolabial folds and smile lines). We chose to avoid the worry lines area because of the high probability it has to be occluded by hairs. Then, we use PCA as for the global warped images to compute temporal signals corresponding to the evolution of the local appearance of the face during time.

2.2 Dynamic features

For each of these three sets, we calculate the log-magnitude Fourier spectra of the associated temporal signals in order to include dynamic information. We also calculate the mean, the standard deviation, the global energy, and the first and second-order spectral moments. We chose to compute these features every second for different sizes of windows (from one to four seconds). This multiscale extraction gives information about short-term and longer-term dynamics.

2.3 Audio features

The last set of features we used is the audio feature set given to the participants of the AVEC'12 Challenge. It contains the most commonly used audio features for the aimed task of predicting emotions from speech (energy, spectral and voice-related features).

2.4 Feature normalization

Within a set of features, the range of values can be highly different from one feature to another. In order to give the same prior to each feature, we need to normalize them. A global standardization on the whole database would be a solution but we chose to standardize each feature by subject in order to reduce the inter-subject variability. This method should be efficient under the hypothesis that the amount of data for each subject is sufficiently representative of the whole emotion space.

3. PREDICTION SYSTEM

Using each of the four feature sets, we make separate predictions for the four dimensions, leading to a total of 16 signals. The method used for each prediction is described below.

3.1 Delay probability estimation

The SEMAINE database has been continuously annotated by humans. Therefore, a delay exists between videos and labels, which may significantly corrupt the learning system [13]. We introduce in this paragraph a delay probability estimation method to avoid this issue. Let $y(t)$ be the label signal and $\{f_i(t), i \in \llbracket 1, n \rrbracket\}$ be a set of n features. Making the assumption that the features that are relevant for our prediction will be more correlated to the undelayed label, we can use the sum of the correlations between the features and the τ seconds delayed label signal as a probability indice for the label to be delayed by τ seconds. Thus, we can estimate the delay probability $P(\tau)$ as follows:

$$P(\tau) = \frac{1}{A} \sum_{i=1}^n r(f_i(t), y(t - \tau)) \quad (2)$$

where r is the Pearson product-moment correlation coefficient defined, for two random variables X and Y as:

$$r(X, Y) = \frac{E(X - \bar{X})E(Y - \bar{Y})}{\sigma_X \sigma_Y}$$

where σ_X refers to the standard deviation of variable X and \bar{X} refers to its mean. A is the normalization coefficient defined as:

$$A = \int_{-\infty}^{\infty} \sum_{i=1}^n r(f_i(t), y(t - \tau)) d\tau$$

We calculate $P(\tau)$ for τ varying in a range $\llbracket 0, T \rrbracket$ where T is the largest expected delay that we fixed at 20 seconds to obtain an estimate of the delay probability distribution in this range. Eq. 2 requires continuous functions. In our case, the data contain different video sequences. We thus estimate the delay probability as the mean of the delay probabilities estimated for the different sequences. To simplify notations, we refer to this estimate as $P(\tau)$.

In Fig. 3, we represent the four different delay probability distributions that have been learned on the training database for the first feature set. By looking at those distributions' maxima, we identify an averaged delay between 3 and 4 seconds for valence and arousal, and between 5 and 6 seconds for expectancy and power. The differences between those delays could be explained by the higher complexity of human evaluation for expectancy and power.

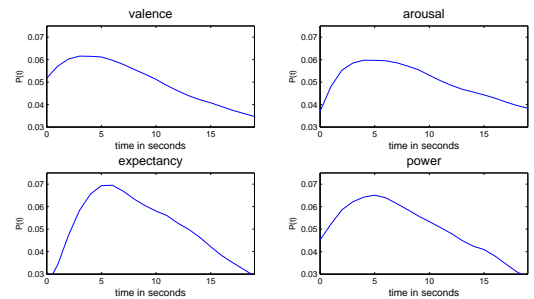


Figure 3: Delay probability distributions

3.2 Correlation-based feature selection

We present in this paragraph a feature selection method adapted to a possibly time-delayed label. The kernel regression proposed in this paper uses a similarity measure based on distances between samples. Using all the features (including the ones that are not useful for our prediction) would corrupt the regression by adding an important noise. We need to identify the most relevant ones and then reduce the number of features that will be used in our distance calculation. In order to only select the features that are correlated to the label knowing the delay probability distribution (Eq. 2), we introduce a time-persistent-correlation-based measure, defined as follows:

$$\rho(f_i(t), y(t)) = \int_{-\infty}^{\infty} r(f_i(t), y(t - \tau)) P(\tau) d\tau \quad (3)$$

This way, we consider the correlation between the feature and the label, but also between the feature and different delayed versions of the label weighted by an estimation of the delay probability. As before, with different separate video sequences, we need to calculate the mean of this measure for the different sequences to obtain a correlation score between the i^{th} feature and the label. To simplify notations, we refer to this score as $\rho(f_i(t), y(t))$. This measure is more robust than a simple correlation measure in the case of possibly time-delayed label (see section 5.3). By selecting features maximizing $\rho(f_i(t), y(t))$, we select a relevant set of features.

3.3 Clustering

We present in this paragraph a clustering step with supervised weighted-distance learning. The feature selection step presented in the previous paragraph gives a correlation score between the label and each selected feature using Eq. 3. We use these scores as the weights of a diagonally-weighted distance d_w , defined as follows:

$$d_w(X, Y) = \sqrt{X^T W Y}$$

where $W \in \mathbb{M}_n(\mathbb{R})$ such as:

$$W_{ij} = \rho(f_i(t), y(t)) \delta_{ij}$$

We perform a k-means clustering algorithm to reduce the uncertainty of the label by grouping samples that are close in the sense of the learned distance d_w . We calculate the label of each group as the mean of the labels of the group. In order to initialize the algorithm, we sort out the samples by label values and gather them in k groups of the same size. We calculate the initialization seeds as the means of the features of each group's samples. This initialization is done to ease the repeatability of the clustering and because we expect to gather samples with neighboring labels after the clustering algorithm by using the learned distance d_w . This step leads to the identification of a set of representative samples.

3.4 Kernel regression

After these learning steps, the prediction is done by a kernel regression using the Nadaraya-Watson estimator ([12]). We use a radial basis function (RBF) combined with the previously learned weighted-distance d_w as kernel. Let $\{\mathbf{x}_j \in \mathbb{R}^n, j \in \llbracket 1, m \rrbracket\}$ be the feature vectors of the m representative samples obtained after the clustering step, and $\{y_j, j \in$

$\llbracket 1, m \rrbracket\}$ be the associated labels. The prediction for a sample s described by feature vector $\mathbf{x}_s \in \mathbb{R}^n$ is given by the following formula:

$$\hat{y}(s) = \frac{\sum_{j=1}^m K_\sigma(\mathbf{x}_s, \mathbf{x}_j) y_j}{\sum_{j=1}^m K_\sigma(\mathbf{x}_s, \mathbf{x}_j)} \quad (4)$$

where σ is the spread of the radial basis function and K is defined as:

$$K_\sigma(\mathbf{x}_s, \mathbf{x}_j) = e^{-\frac{d_w(\mathbf{x}_s, \mathbf{x}_j)^2}{\sigma}} \quad (5)$$

As a final step, we proceed to a temporal smoothing to reduce the noise of the regressor output.

4. FUSION

Using the regression method described in the previous section, we obtain 16 signals, which are the predictions of the four dimensions using the four different sets of features. In order to fuse these signals and make the final prediction of the four dimensions, we chose to use local linear regressions to estimate linear relationships between the signals and the labels. More precisely, the coefficients of these linear relationships are estimated as the means of the different linear regressions coefficients weighted by the Pearson's correlation between the predicted signal and the label of each sequence. Let $\{y_i^j, i \in \llbracket 1, n_s \rrbracket, j \in \{V, A, E, P\}\}$ be the labels of the n_s video sequences of the learning set. Let $\{S_i, i \in \llbracket 1, n_s \rrbracket\}$ be the matrices containing the 16 predictions of our system on the n_s sequences of the training set (previously standardized). We estimate the four vectors of coefficients $\alpha_j \in \mathbb{R}^{16}$ of the linear relationships as follows:

$$\alpha_j = \frac{\sum_{i=1}^{n_s} r(\beta_i^j S_i, y_i^j) \beta_i^j}{\sum_{i=1}^{n_s} r(\beta_i^j S_i, y_i^j)} \quad (6)$$

where $\beta_i^j = (S_i^T S_i)^{-1} S_i^T y_i^j$ is the ordinary least squares coefficients vector for sequence i and label j . We can then calculate our final predictions for the four dimensions $\{\hat{y}_j, j \in \{V, A, E, P\}\}$ as: $\hat{y}_j = \alpha_j^T S_t$ where S_t is a matrix containing the 16 standardized predictions of our regressors on the test sequence we aim to predict.

5. EXPERIMENTS

In this section, we present some experiments we carried out to evaluate the different key points of our method. In order to be robust in generalization, we chose to optimize the hyperparameters in subject-independent cross-validation (each training partition does not contain the tested subject).

As evaluation procedure, we first present the results of the full system (with feature normalization by subject, our time-persistent-correlation measure and our regression framework). Then, we evaluate the contribution of each key point by replacing it by a more commonly used process (global normalization, Pearson's correlation and Support Vector Regression):

1. Normalization by subject, Time-persistent-correlation, Kernel regression (sect. 5.1)

2. Global normalization, Time-persistent-correlation, Kernel regression. (sect. 5.2)
3. Normalization by subject, Standard correlation, Kernel regression. (sect. 5.3)
4. Normalization by subject, Time-persistent-correlation, SVR. (sect. 5.4)

5.1 Fusion evaluation

The proposed fusion method, which is based on a simple linear combination of the inputs learned via local linear regressions, is particularly fast and well-suited for a real-time system. To evaluate the efficiency of this fusion method and the contribution of each feature set, we present the results we obtained by learning on the training set and testing on the development set in Table 1.

Table 1: Pearson’s correlations averaged over all sequences of the AVEC’12 development set. Results are given for valence, arousal, expectancy and power. We also indicate the mean of these four dimensions. S corresponds to the shape features. GA to the global appearance features. LA to the local appearance features and A to the audio features. F corresponds to the fusion.

| | Val | Aro | Exp | Pow | Mean |
|----|--------------|--------------|--------------|--------------|--------------|
| S | 0.319 | 0.538 | 0.365 | 0.429 | 0.413 |
| GA | 0.281 | 0.498 | 0.347 | 0.431 | 0.389 |
| LA | 0.354 | 0.470 | 0.323 | 0.432 | 0.395 |
| A | -0.057 | 0.445 | 0.280 | 0.298 | 0.241 |
| F | 0.350 | 0.644 | 0.341 | 0.511 | 0.461 |

We can see that visual features give better results than audio features. Local appearance-based features give a better valence prediction than the other sets. The fusion system significantly improves arousal and power predictions giving a mean score increased by 11.7%. We can notice that when the four predictions (using each set of features) are accurate, the fusion is more successful. On the contrary, the prediction scores of valence and expectancy are lower and the fusion does not improve the system performance.

5.2 Normalization evaluation

In this subsection, we evaluate the effect of the standardization of the features that we performed by subject in order to reduce the inter-subject variability. We compare the results we obtained (presented in the previous table) to those achieved with a global standardization on the whole training set (Table 2).

The normalization by subject has increased the mean score by 9.8%. The effect on valence is more important than on the other dimensions, which can be explained because the selected features for valence predictions are more sensitive to human morphological variations. Most of the features selected for the three other dimensions are high-frequency sub-bands energies extracted from the temporal signals, which are more robust to morphological variations than the signals’ mean values that seem to be useful to predict valence.

Table 2: Pearson’s correlations averaged over all sequences of the AVEC’12 development set in the case of a global standardization instead of a standardization by subject.

| | Val | Aro | Exp | Pow | Mean |
|----|--------------|--------------|--------------|--------------|--------------|
| S | 0.079 | 0.526 | 0.373 | 0.463 | 0.361 |
| GA | 0.102 | 0.471 | 0.353 | 0.416 | 0.335 |
| LA | 0.314 | 0.436 | 0.311 | 0.441 | 0.376 |
| A | -0.069 | 0.509 | 0.227 | 0.254 | 0.230 |
| F | 0.199 | 0.633 | 0.331 | 0.515 | 0.420 |

5.3 Time-persistent-correlation evaluation

For evaluating the efficiency of the new proposed correlation-based measure, we compare our results to those we obtain by selecting the features with a standard Pearson’s correlation measure which does not take the delay into account. The results are presented in Table 3.

Table 3: Pearson’s correlations averaged over all sequences of the AVEC’12 development set in the case of the use of Pearson’s correlation instead of our new time-persistent-correlation for feature selection.

| | Val | Aro | Exp | Pow | Mean |
|----|--------------|--------------|--------------|--------------|--------------|
| S | 0.299 | 0.527 | 0.273 | 0.413 | 0.378 |
| GA | 0.297 | 0.489 | 0.279 | 0.392 | 0.364 |
| LA | 0.303 | 0.464 | 0.294 | 0.411 | 0.368 |
| A | 0.017 | 0.426 | 0.261 | 0.265 | 0.242 |
| F | 0.333 | 0.652 | 0.301 | 0.453 | 0.435 |

The use of the proposed time-persistent-correlation-based measure has increased the mean score by 6%, which can be explained by the improved robustness of the proposed measure to possibly time-delayed labels.

5.4 Regressor evaluation

Our regression method, which consists of a clustering and a kernel regression, is particularly fast to learn and compute and is therefore suited for real-time implementation. We compare our method to the commonly used Support Vector Regression combined with the kernel defined in Eq. 5. As for our method, the hyperparameters are optimized in subject-independent cross-validation. The obtained results are presented on Table 4.

Table 4: Pearson’s correlations averaged over all sequences of the AVEC’12 development set with a Support Vector Regression.

| | Val | Aro | Exp | Pow | Mean |
|----|--------------|--------------|--------------|--------------|--------------|
| S | 0.286 | 0.504 | 0.360 | 0.442 | 0.398 |
| GA | 0.252 | 0.393 | 0.347 | 0.404 | 0.349 |
| LA | 0.363 | 0.473 | 0.309 | 0.411 | 0.389 |
| A | -0.089 | 0.400 | 0.232 | 0.380 | 0.231 |
| F | 0.275 | 0.591 | 0.297 | 0.493 | 0.414 |

The mean score after fusion has increased by 11% by using our method. However, we can see that the fusion is less

efficient with SVR than with our regression method. It can be explained by the fact that the fusion coefficients have been estimated using the SVR predictions on the training set. A likely explanation could be that SVR are prone to over-fitting. A solution to this issue would be to learn the fusion coefficients in cross-validation. It is thus not relevant to compare the results after fusion. It is more reliable to compare our regression method to the SVR feature set by feature set. We obtain, in this case, an averaged gain of 5%.

6. RESULTS ON THE TEST SET

We learned our system on the concatenation of the training and the development sets to compute our predictions on the test set. We compare in Table 5 our results to those given in the baseline paper [18]. We can notice that the results obtained on the test set are quite similar to those obtained on the development set. This highlights the high generalization power of the proposed framework. It can be explained by the small number of representative samples for the kernel regression (60 in our system) which limits the flexibility of the model and allows the system to only capture important trends in the data.

Table 5: Pearson’s correlations averaged over all sequences of the AVEC’12 test set.

| | Val | Aro | Exp | Pow | Mean |
|------------|--------------|--------------|--------------|--------------|--------------|
| Our method | 0.341 | 0.612 | 0.314 | 0.556 | 0.456 |
| Baseline | 0.146 | 0.149 | 0.110 | 0.138 | 0.136 |

7. CONCLUSION

We presented a complete framework for continuous prediction of human emotions based on features characterizing head movements, face appearance and voice in a dynamic manner by using log-magnitude Fourier spectra. We introduced a new correlation-based measure for feature selection and evaluated its efficiency and robustness in the case of possibly time-delayed labels. We proposed a fast regression framework based on a supervised clustering followed by a Nadaraya-Watson kernel regression that appears to outperform, for the aimed task, Support Vector Regression. Our fusion method is based on simple local linear regressions and significantly improves our results. Because of the high power of generalization of our method, we directly learned our fusion parameters using our regressors outputs on the training set. In order to improve the fusion for methods that are more prone to over-fitting, we would have to learn these parameters in cross-validation. Our system has been designed for the Audio/Visual Emotion Challenge (AVEC’12) which uses Pearson’s correlation as evaluation measure. Therefore, every step of our method has been built and optimized to maximize this measure. An accurate system for everyday interactions would need to be efficient in terms of correlation but also in terms of Root-Mean-Square Error (RMSE). Some modifications on our system would be needed to increase its performance regarding this measure. The SEMAINE database on which our system has been learned and tested contains videos of natural interactions but recorded in a very constraint environment. A perspective for adapting these kinds of human emotion prediction systems to real

conditions, as for assistance robotics, would be to learn the system on “in the wild” data

8. ACKNOWLEDGMENTS

This work has been partially supported by the French National Agency (ANR) in the frame of its Technological Research CONTINT program (IMMEMO, project number ANR-09-CORD-012), the French FUI project PRAMAD2 (project number J11P159) and the Cap Digital Business cluster for digital content

9. REFERENCES

- [1] A. Cruz, B. Bhanu, and S. Yang. A psychologically-inspired match-score fusion model for video-based facial expression recognition. In *Proc. of Affective Computing and Intelligent Interaction (ACII’11)*, pages 341–350, 2011.
- [2] M. Dahmane and J. Meunier. Continuous emotion recognition using gabor energy filters. In *Proc. of Affective Computing and Intelligent Interaction (ACII’11)*, pages 351–358, 2011.
- [3] P. Ekman. Facial expression and emotion. *American Psychologist*, 48(4):384, 1993.
- [4] P. Ekman and W. Friesen. Facial action coding system: A technique for the measurement of facial action. *Manual for the Facial Action Coding System*, 1978.
- [5] J. Fontaine, K. Scherer, E. Roesch, and P. Ellsworth. The world of emotions is not two-dimensional. *Psychological science*, 18(12):1050, 2007.
- [6] H. Gunes and M. Pantic. Dimensional emotion prediction from spontaneous head gestures for interaction with sensitive artificial listeners. In *Proc. of Intelligent Virtual Agents (IVA’10)*, pages 371–377, 2010.
- [7] H. Gunes, B. Schuller, M. Pantic, and R. Cowie. Emotion representation, analysis and synthesis in continuous space: A survey. In *Proc. IEEE Int’l Conf. Face & Gesture Recognition (FG’11)*, pages 827–834, 2011.
- [8] B. Jiang, M. Valstar, and M. Pantic. Action unit detection using sparse appearance descriptors in space-time video volumes. In *Proc. IEEE Int’l Conf. Face & Gesture Recognition (FG’11)*, pages 314–321, 2011.
- [9] D. McDuff, R. El Kaliouby, K. Kassam, and R. Picard. Affect valence inference from facial action unit spectrograms. In *Proc. IEEE Int’l Conf. Computer Vision and Pattern Recognition Workshops (CVPRW’10)*, pages 17–24, 2010.
- [10] G. McKeown, M. Valstar, R. Cowie, and M. Pantic. The semaine corpus of emotionally coloured character interactions. In *Proc. IEEE Int’l Conf. on Multimedia and Expo (ICME’10)*, pages 1079–1084, 2010.
- [11] H. Meng and N. Bianchi-Berthouze. Naturalistic affective expression classification by a multi-stage approach based on hidden markov models. In *Proc. of Affective Computing and Intelligent Interaction (ACII’11)*, pages 378–387, 2011.
- [12] E. Nadaraya. On estimating regression. *Theory of Prob. and Appl.*, 9:141–142, 1964.

- [13] M. Nicolaou, H. Gunes, and M. Pantic. Automatic segmentation of spontaneous data using dimensional labels from multiple coders. In *Workshop on Multimodal Corpora: Advances in Capturing, Coding and Analyzing Multimodality*, pages 43–48, 2010.
- [14] M. Nicolaou, H. Gunes, and M. Pantic. Output-associative rvm regression for dimensional and continuous emotion prediction. *Image and Vision Computing*, 2012.
- [15] G. Ramirez, T. Baltrušaitis, and L. Morency. Modeling latent discriminative dynamic of multi-dimensional affective signals. In *Proc. of Affective Computing and Intelligent Interaction (ACII'11)*, pages 396–406, 2011.
- [16] J. M. Saragih, S. Lucey, and J. F. Cohn. Deformable Model Fitting by Regularized Landmark Mean-Shift. *International Journal of Computer Vision*, 91(2):200–215, Sept. 2010.
- [17] A. Sayedelahl, P. Fewzee, M. Kamel, and F. Karray. Audio-based emotion recognition from natural conversations based on co-occurrence matrix and frequency domain energy distribution features. In *Proc. of Affective Computing and Intelligent Interaction (ACII'11)*, pages 407–414, 2011.
- [18] B. Schuller, M. Valstar, F. Eyben, R. Cowie, and M. Pantic. Avec 2012 – the continuous audio/visual emotion challenge. In *Proc. Second International Audio/Visual Emotion Challenge and Workshop (AVEC'12) to appear*, 2012.
- [19] B. Schuller, M. Valstar, F. Eyben, G. McKeown, R. Cowie, and M. Pantic. Avec 2011—the first international audio/visual emotion challenge. In *Proc. of Affective Computing and Intelligent Interaction (ACII'11)*, pages 415–424, 2011.
- [20] T. Senechal, V. Rapp, H. Salam, R. Segulier, K. Bailly, and L. Prevost. Facial action recognition combining heterogeneous features via multi-kernel learning. *IEEE Transactions on Systems, Man, and Cybernetics–Part B*, 42(4):993–1005, 2012.
- [21] M. Valstar, M. Mehu, B. Jiang, M. Pantic, K. Scherer, B. Jiang, M. Valstar, M. Pantic, M. Valstar, B. Jiang, et al. Meta-analysis of the first facial expression recognition challenge. *IEEE Transactions on Systems, Man, and Cybernetics–Part B*, 2012.
- [22] P. Viola and M. Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.
- [23] M. Wöllmer, M. Kaiser, F. Eyben, and B. Schuller. Lstm-modeling of continuous emotions in an audiovisual affect recognition framework. *Image and Vision Computing*, 2012.

Appendix C

Binary Map based Landmark Localization

This appendix contains a work describing a new way of using Local Binary Patterns for landmark localization. It has been published in IEEE International Conference on Image Processing (ICIP 2013).

LOCATING FACIAL LANDMARKS WITH BINARY MAP CROSS-CORRELATIONS

Jérémie Nicolle Kévin Bailly Vincent Rapp Mohamed Chetouani

Univ. Pierre & Marie Curie, ISIR - CNRS UMR 7222, F-75005, Paris - France
{nicolle, baily, rapp, chetouani}@isir.upmc.fr

ABSTRACT

Precise facial landmark localization in still images is a key step for many face analysis applications, such as biometrics or automatic emotion recognition. In this paper, we propose a framework for facial point detection in frontal and near-frontal images. We introduce a new appearance model based on binary map cross-correlations that efficiently uses LBP and LPQ in a localization context. Inclusion of shape-related constraints is performed by a nonparametric voting method using relational properties within triplets of points, designed to correct outliers without losing precision for accurately detected points. We tested our system’s performance on the widely used as benchmark BioID database obtaining state-of-the-art results. We also discuss evaluation metrics used to compare facial landmarking systems and which have been mixed up in recent literature.

Index Terms — LBP, LPQ, facial landmarks, shape model, binary maps.

1. INTRODUCTION

The goal of facial landmarking is to precisely locate a set of key points in faces, delimiting the eyes, the eyebrows, the nose and the mouth. This task is particularly challenging because of the variations in head pose, morphology, expression and illumination. Most state-of-the-art methods combine appearance-based information with shape-related constraints to increase robustness. However, important differences exist among those methods, concerning features, image exploration techniques and shape-related constraints.

Feature choice is a key point in localization methods. A balanced trade-off must be found between performance and computation time. Particularly fast-to-compute features can be used (Milborrow and Nicolls [1] use an Active Shape Model based on gray levels and Cootes *et al.* [2] use Haar-like features). More discriminant features, like dense SIFT used by Belhumeur *et al.* [3], can lead to very precise results

but are time-consuming. Because of their low computation time and their robustness towards illumination and blur, we chose to use LBP and LPQ and to include them in a new detection-oriented framework (details in §2.1).

Various image exploration techniques for point regression have been used in literature. Some methods estimate probability maps on large areas and predict point locations within these areas (dense exploration techniques). To estimate these probability maps, generative methods can be used (for example using a distance to a manifold estimated by PCA [1] or cross-correlating gray level mean patches [4]), as well as discriminant methods (for instance using a distance to an hyperplane estimated by SVM [5]). Other methods use more local areas to estimate point locations, that are potentially outside the areas used for feature extraction (sparse exploration techniques), as in [6] where the authors use SVR, or in [2, 7, 8] where random forests are used and lead to very fast algorithms. Nevertheless, sparse exploration based methods have the disadvantage of highly depending on initialization. To avoid this issue and because of the robustness it induces, we opted for a generative dense exploration technique.

Combining shape-related constraints with appearance-based detection has proven to increase robustness. A commonly used approach for modeling these constraints is to perform PCA to learn admissible deformations and optimize a cost function in the space of the parameters controlling these global deformations [9]. However, in order to stay within the learned manifold, many accurately located points may be displaced. We propose a nonparametric voting method based on relational properties within triplets of points that lets us introduce more local constraints correcting outliers without losing precision for accurately detected points (details in §2.2).

In this paper, we detail our facial feature detection algorithm and present our results on the well-know BioID database. Different evaluation metrics for facial feature localization algorithms can be used: one represents the cumulative distribution of image mean errors and the other the cumulative distribution of landmark errors. These curves have been mixed up in recent literature and have led to questions raised in [2] concerning the lack of distinctive “S” shape of some result curves. We propose a discussion about these evaluation metrics in §3.1.

2. FRAMEWORK

In our method, appearance-based regression is first performed cross-correlating LBP and LPQ binary maps with mean patches calculated on the learning database. Then, we iteratively correct potential outliers with shape-related constraints based on relational properties within triplets of points.

2.1. LBP-LPQ based probability maps

Local Binary Patterns (LBP) and Local Phase Quantization (LPQ) have proven their efficiency to characterize appearance [10, 11], mainly because of their robustness towards illumination and blur. Most facial analysis methods involving LBP or LPQ use them by computing histograms on different areas within the images [10, 11]. Histograms are commonly used because finding a relevant distance between LBP (or LPQ) values is not straightforward (appearance of pixels coded by close values can be very different). However, histograms do not keep information about the spatial distribution of the appearance within the areas of computation. Moreover, reducing the size of these areas can increase precision but raises the issue of finding an appropriate distance for sparse histograms. Thus, using them for precise localization can be difficult.

We propose a solution to efficiently use these features for precise point detection. In our method, we learn a mean patch for each point and each LBP (and LPQ) value and calculate probability maps by cross-correlating a few selected mean patches with the corresponding LBP (or LPQ) binary maps extracted from the test images.

2.1.1. LBP-LPQ mean patch learning

For each image of our learning database, we compute 2^8 binary LBP-maps and 2^8 binary LPQ-maps. The b^{th} map takes the value 1 for the pixels coded by the LBP (or LPQ) value b . We extract binary patches centered on each landmark and average them over all the images to obtain our mean patches. These mean patches give information about the probability of presence of pixels coded by each LBP or LPQ value. This way, we extract illumination and blur invariant features characterizing appearance around each landmark keeping precise spatial distribution related information that would have been lost by histogram computation. Figure 1 illustrates the extraction of LBP and LPQ mean patches for an area centered on the right eye.

2.1.2. Feature selection and weighting

In order to select the maps that are relevant for each of the landmarks and weight them appropriately, we calculate an accuracy score for all the 2^9 weak regressors on the learning set. Let $\{\mathbf{P}^{k,b}, k \in \llbracket 1, n_p \rrbracket, b \in \llbracket 1, 2^9 \rrbracket\}$ be the previously learned mean patches for each of the n_p landmarks and $\{\mathbf{M}^b(l)\}$ be

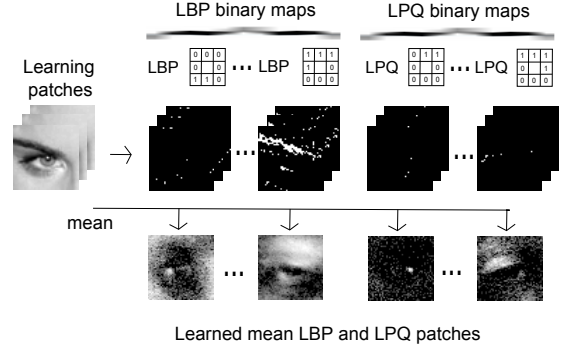


Fig. 1. Mean patch calculation process.

the binary maps for image l . Each patch gives an estimation of the location of the k^{th} landmark using:

$$\mathbf{p}_e^{k,b}(l) = \operatorname{argmax}(\mathbf{M}^b(l) * \mathbf{P}^{k,b})$$

where $*$ denotes a normalized cross-correlation. We compute a response map for each landmark and each patch on the whole training database following:

$$R^{k,b} = \sum_{l=1}^{n_l} \delta_{(\mathbf{p}_e^{k,b}(l) - \mathbf{p}_t^k(l))}$$

where n_l is the number of images in the learning set, $\delta_{\mathbf{a}}(x, y)$ takes the value 1 when $(x, y) = \mathbf{a}$ and $\mathbf{p}_t^k(l)$ is the true location of the k^{th} landmark on image l . Then, we calculate the accuracy scores as: $S^{k,b} = \iint R^{k,b} \cdot G$ where G is a gaussian with zero-mean, thus according more weight to the weak regressors that have often been placing the landmark close to its true location in the learning images. We use these accuracy scores to select the more relevant weak regressors for each point and appropriately weight them.

2.1.3. Probability map calculation

We perform these previous steps for two different sizes of mean patches. Large patches aim at roughly locating points using information about areas that can be relatively far and small patches aim at placing points more precisely, only using local information. Using these two sets of patches and their associated accuracy scores, we define our probability maps for the test images as follows:

$$\mathbf{J}^k(l) = \mathbf{I}_{large}^k(l) + \alpha \cdot \mathbf{I}_{small}^k(l)$$

where

$$\mathbf{I}^k(l) = \sum_{j=1}^{n_k} S^{k, \mathbf{sel}^k(j)} \cdot (\mathbf{M}^{\mathbf{sel}^k(j)}(l) * \mathbf{P}^{k, \mathbf{sel}^k(j)})$$

The parameter α sets the relative impact of the two different sizes of patches. The vectors \mathbf{sel} contain the indices of

the previously selected maps and n_k is their length. These appearance-based maps give information about the probability of presence of each landmark and will be combined with attraction maps calculated using our shape model to make the final algorithm.

2.2. Triplet-based shape model

The purpose of the shape model in landmark localization is to correct potential outliers. We learn relational properties (ratios of distances and angles) within all triplets of points and select for each point the more stable triplets. During the test phase, we use a k-nearest neighbor algorithm to obtain a similar model and generate attraction maps used to correct, step by step, potential outliers.

2.2.1. Triplet model

For each triplet of points $\mathbf{t}_{k_1 k_2 k_3} = (\mathbf{p}_{k_1}, \mathbf{p}_{k_2}, \mathbf{p}_{k_3})$ of a learning image l , we calculate the ratio of distances and the angle between the vectors $\mathbf{v}_{k_2 k_3} = \mathbf{p}_{k_3} - \mathbf{p}_{k_2}$ and $\mathbf{v}_{k_2 k_1} = \mathbf{p}_{k_1} - \mathbf{p}_{k_2}$ to form

$$f^l(\mathbf{t}_{k_1 k_2 k_3}) = \frac{\|\mathbf{v}_{k_2 k_1}\|}{\|\mathbf{v}_{k_2 k_3}\|} \cdot e^{i(\widehat{\mathbf{v}_{k_2 k_3}}, \mathbf{v}_{k_2 k_1})}$$

that indicates the location of \mathbf{p}_{k_1} relatively to \mathbf{p}_{k_2} and \mathbf{p}_{k_3} . For each point, we select the more stable triplets on the learning database.

2.2.2. Attraction map calculation

During the test phase, for a configuration of points $C = \{\mathbf{p}_k, k \in \llbracket 1, n_p \rrbracket\}$, we define an attraction map for each point, which is computed using a voting method and the previously selected triplets. First, we use a k-nearest neighbor algorithm to obtain a similar model s using f as features. With enough variety in the learning database, we can find a model with close head pose, expression and morphology that will let us appropriately constrain our local detector responses. Then, each selected triplet (k_1, k_2, k_3) votes for a location for point \mathbf{p}_{k_1} using points \mathbf{p}_{k_2} and \mathbf{p}_{k_3} of configuration C . Each vote is a gaussian centered at location:

$$\mathbf{p}_{k_1}^{k_2 k_3} = \mathbf{p}_{k_2} + \mathbf{v}_{k_2 k_3} \cdot f^s(\mathbf{t}_{k_1 k_2 k_3})$$

The accumulation of these votes for all selected triplets gives an attraction map \mathbf{A}^k for each point. We weight probability maps with these attraction maps to obtain model-weighted probability maps (with \circ the Hadamard product) :

$$\mathbf{W}^k = \mathbf{A}^k \circ \mathbf{J}^k$$

An illustration is given figure 2. Our approach introduces local constraints that let us keep the accurately located points in place, as opposed to a global and stronger constraint, for instance forcing the point configuration to stay within a manifold learned via PCA.

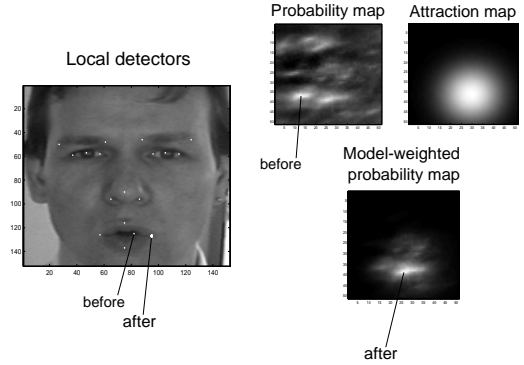


Fig. 2. Outlier correction.

2.2.3. Iterative correction

We present the final algorithm of our landmark detector, which iteratively corrects outliers, and finally selects the likeliest configuration (algorithm 1). We define the likelihood function $L(C)$, based on the shape model, as follows:

$$\frac{1}{L(C)} = \sum_k \sum_{j=1}^{m_k} H(d_{k,j} - \beta)$$

with $d_{k,j}$ the distance between $f^C(\mathbf{t}_j^k)$ and $f^s(\mathbf{t}_j^k)$, m_k the number of selected triplets for point k , H the Heaviside step function and β an acceptance threshold. We use a step function in order to let all admissible locations unpenalized. Thus, the inverse likelihood indicates an estimation of the number of triplets that appear to contain an outlier.

start

initialize a configuration C_0 by calculating probability maps \mathbf{J}^k

$C_0 = \{\mathbf{p}_k\}$ with $\mathbf{p}_k = \text{argmax}(\mathbf{J}^k)$

for step u do

calculate $f^{C_{u-1}}$

generate similar model via k-NN

calculate attraction maps \mathbf{A}^k for C_{u-1}

calculate model-weighted maps using

$\mathbf{W}^k = \mathbf{A}^k \circ \mathbf{J}^k$

estimate the new configuration C_u

$C_u = \{\mathbf{p}_k\}$ with $\mathbf{p}_k = \text{argmax}(\mathbf{W}^k)$

calculate the likelihood of C_u

$l(u) = L(C_u)$

end

find the best configuration

$u_{final} = \text{argmax}(l)$

$C_{final} = C_{u_{final}}$

stop

Algorithm 1: Binary Map based Point Localization (BiM-PoL)

3. RESULTS

In this section, we first discuss evaluation metrics used to compare landmarking systems before comparing our results with recent state-of-the-art methods.

3.1. Evaluation metrics

Two different kinds of evaluation metrics have been used in recent literature in the domain, leading to relevant questions raised in [2]. One is based on the m_{e17} measure proposed in [1] and represents the cumulative distribution of the image errors (a mean error is calculated for each image and the curve indicates the proportion of images whose mean errors are inferior to a certain threshold). The other represents the cumulative distribution of the landmark errors (indicating the proportion of landmarks whose errors are inferior to a certain threshold). These curves have been mixed up in a lot of recent papers and are definitely not alike (as shown comparing figures 3 and 4). Considering that the predictions follow normal distributions centered on the true landmark locations, then the error obtained for one landmark follows half-normal distribution, which is why the intermediate mean operation and the number of landmarks used for this mean calculation has influence on the shape of image errors cumulative distribution. The starting threshold and the slope increase with the number of landmarks, explaining the differences between figure 3 (image mean errors with 17 landmarks leading to a distinctive "S" shape) and figure 4 (landmark errors).

3.2. Performance evaluation

In this paragraph, we present our results and compare them to other recent state-of-the-art systems (RFRV [2], STASM [1], Cons [3]) with the evaluation metrics used in respective papers. The different parameters of our algorithm (number of selected mean patches, number of triplets used for shape-related constraints inclusion...) have been optimized in cross-validation on the learning database. We learned our system on 500 images of LFPW database (proposed in [3]) that includes interesting variability in illumination, morphology or head pose, and tested it on the well-known BioID database. For testing, we used the 1083 images on which Viola-Jones face detector gave a relevant response. Because of the different point annotations between the learning and the test database, constant biases have been introduced as in [3]. Our results are shown in figure 3 in terms of proportion of images whose m_{e17} errors are inferior to a certain threshold. We obtain results slightly better than the recent regression forest approach proposed by Cootes *et al.* in [2]. Figure 4 shows our results in terms of proportion of landmarks whose errors are inferior to a certain threshold. Our results are equivalent to the recent consensus of exemplars approach proposed by Belhumeur *et al.* in [3].

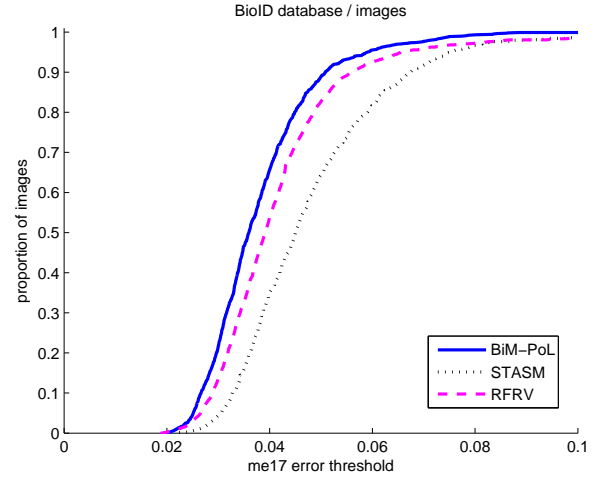


Fig. 3. Cumulative distribution by image errors for BioID database.

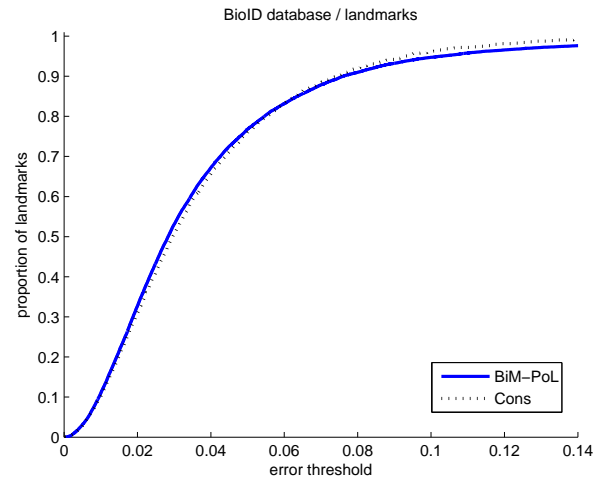


Fig. 4. Cumulative distribution by landmark errors for BioID database.

4. CONCLUSION

We presented in this paper a new method for facial landmark localization in frontal and near-frontal images leading to state-of-the-art results. We proposed a new appearance model for using LBP and LPQ in the context of detection by using binary map cross-correlations to estimate probability maps. In our method, we include shape-related constraints via a voting method using relational properties within triplets of points. This shape model lets us introduce more local constraints than using the widely used global PCA approach and avoid small displacements for accurately located points. This paper also intends to clarify evaluation metrics that have often been mixed up in recent literature in the domain.

5. REFERENCES

- [1] S. Milborrow and F. Nicolls, “Locating facial features with an extended active shape model,” in *European Conference on Computer Vision (ECCV)*, 2008, 2008, pp. 504–513.
- [2] T. Cootes, M. Ionita, C. Lindner, and P. Sauer, “Robust and accurate shape model fitting using random forest regression voting,” in *European Conference on Computer Vision (ECCV)*, 2012, 2012, pp. 278–291.
- [3] P.N. Belhumeur, D.W. Jacobs, D.J. Kriegman, and N. Kumar, “Localizing parts of faces using a consensus of exemplars,” in *Computer Vision and Pattern Recognition (CVPR)*, 2011 IEEE Conference on, June 2011, pp. 545–552.
- [4] J. M. Saragih, S. Lucey, and J. Cohn, “Face alignment through subspace constrained mean-shifts,” in *International Conference on Computer Vision (ICCV)*, 2009, Sep. 2009.
- [5] V. Rapp, T. Senechal, K. Bailly, and L. Prevost, “Multiple kernel learning svm and statistical validation for facial landmark detection,” in *Automatic Face Gesture Recognition and Workshops (FG)*, 2011 IEEE International Conference on, Mar. 2011, pp. 265–271.
- [6] B. Martinez, M. Valstar, X. Binefa, and M. Pantic, “Local evidence aggregation for regression based facial point detection,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2012.
- [7] M. Dantone, J. Gall, G. Fanelli, and L. Van Gool, “Real-time facial feature detection using conditional regression forests,” in *Computer Vision and Pattern Recognition (CVPR)*, 2012 IEEE Conference on, 2012.
- [8] X. Cao, Y. Wei, F. Wen, and J. Sun, “Face alignment by explicit shape regression,” in *Computer Vision and Pattern Recognition (CVPR)*, 2012 IEEE Conference on, June 2012, pp. 2887–2894.
- [9] K. Bailly, M. Milgram, P. Phothisane, and E. Bigorgne, “Learning global cost function for face alignment,” in *International Conference on Pattern Recognition (ICPR)*, 2012, 2012.
- [10] G. Zhao and M. Pietikainen, “Dynamic texture recognition using local binary patterns with an application to facial expressions,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 6, pp. 915–928, Jun. 2007.
- [11] T. Ahonen, E. Rahtu, V. Ojansivu, and J. Heikkil, “Recognition of blurred faces using local phase quantization,” in *International Conference on Pattern Recognition (ICPR)*, 2008, 2008.