Latent Dynamic Space-Time Volumes for Predicting Human Facial Behavior in Videos

A dissertation submitted in partial satisfaction of the requirements for the degree of Doctor of Philosophy

in

Electrical Engineering (Intelligent Systems, Robotics and Control)

by

Karan Sikka

Committee in charge:

Truong Nguyen, Chair
Marian Bartlett
Sanjoy Dasgupta
Bhaskar Rao
Zhuowen Tu
Nuno Vasconcelos

2016
The Dissertation of Karan Sikka is approved and is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California, San Diego

2016
DEDICATION

To my grandparents and parents
## TABLE OF CONTENTS

Signature Page ................................................................. iii
Dedication ................................................................. iv
Table of Contents .............................................................. v
List of Figures ................................................................. viii
List of Tables ................................................................. x
Acknowledgements ............................................................ xi
Vita ................................................................. xiv
Abstract of the Dissertation .................................................. xvi

Chapter 1 Introduction ...................................................... 1
  1.1 Human Behavior Prediction in Videos ................................ 2
  1.2 Overview and Organization ........................................... 5

Chapter 2 Weakly Supervised Pain Localization and Classification with Multiple Segment Learning ........................................... 9
  2.1 Introduction .......................................................... 9
  2.2 Related Work and Motivation ...................................... 13
  2.3 MIL ............................................................... 16
    2.3.1 Gradient Boosting ............................................. 17
    2.3.2 MilBoost ....................................................... 19
  2.4 Multiple Instance Learning based on Multiple Segments (MS-MIL) ......................................................... 22
    2.4.1 Overview ....................................................... 22
    2.4.2 Bag of Words based Representation (BoW) ................. 26
    2.4.3 Multiple Segment (MS) Representation .................... 27
  2.5 Experimental Design ................................................ 28
    2.5.1 Dataset ......................................................... 28
    2.5.2 Performance Metrics ......................................... 29
  2.6 Results and Discussion ............................................. 31
    2.6.1 Performance Evaluation of Pain Classification .......... 31
    2.6.2 Performance Evaluation of Pain Localization .......... 34
    2.6.3 How does the Multiple Segment Representation effect MS-MIL ...................................................... 38
    2.6.4 Approaches for Generating Multiple Segments (Normalized Cuts vs Scanning Windows) .......................... 40
  2.7 Visualizing the Classifier and Benefits of Bagging .............. 41
  2.8 Experiments on FEEDTUM dataset ................................ 43
# Chapter 3: Exemplar Hidden Markov Models for Classification of Facial Expressions in Videos

## 3.1 Introduction

## Related Work

## Methods

### 3.3.1 Problem Statement

### 3.3.2 HMMs for AFER

### 3.3.3 Probabilistic Kernels

### 3.3.4 Exemplar-HMMs for AFER

## 3.4 Experiments

### 3.4.1 Experiments on Posed and Spontaneous Basic Emotions

### 3.4.2 Experiments on Facial Action Time Series

### 3.4.3 Algorithms compared

## 3.5 Results and Discussion

## 3.6 Conclusion and Future Work

## 3.7 Acknowledgment

---

# Chapter 4: Joint Clustering and Classification for Multiple Instance Learning

## 4.1 Introduction

## Related Work

## Joint Clustering and Classification for Multiple Instance Learning (JC^2ML)

### 4.3.1 Model

### 4.3.2 Joint Optimization

## 4.4 Experiments

## 4.5 Results and Discussion

### 4.5.1 Advantages of Discovering Discriminative Concepts

## 4.6 Conclusion and Future Directions

## 4.7 Acknowledgment

---

# Chapter 5: LOMo: Latent Ordinal Models for Facial Analysis in Videos

## 5.1 Introduction

## Related works

## Approach

### 5.3.1 Scoring function

### 5.3.2 Learning

## Experimental Results

### 5.4.1 Implementation Details and Baselines

### 5.4.2 Quantitative Results
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.4.3</td>
<td>Qualitative Results</td>
<td>101</td>
</tr>
<tr>
<td>5.4.4</td>
<td>Comparison with State-of-the-Art</td>
<td>101</td>
</tr>
<tr>
<td>5.5</td>
<td>Conclusion</td>
<td>104</td>
</tr>
<tr>
<td>5.6</td>
<td>Appendix</td>
<td>105</td>
</tr>
<tr>
<td>5.6.1</td>
<td>Effect of Parameters</td>
<td>105</td>
</tr>
<tr>
<td>5.6.2</td>
<td>Intuitive Example for Understanding the Scoring Function</td>
<td>105</td>
</tr>
<tr>
<td>5.6.3</td>
<td>Visualization of Detected Events</td>
<td>106</td>
</tr>
<tr>
<td>5.6.4</td>
<td>Additional Quantitative Results</td>
<td>107</td>
</tr>
<tr>
<td>5.7</td>
<td>Acknowledgement</td>
<td>108</td>
</tr>
<tr>
<td>Chapter 6</td>
<td>Conclusion and Future Work</td>
<td>111</td>
</tr>
<tr>
<td>6.0.1</td>
<td>Future Work</td>
<td>115</td>
</tr>
<tr>
<td>Bibliography</td>
<td></td>
<td>117</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>------</td>
</tr>
<tr>
<td>1.1</td>
<td>Some applications of automated behavior prediction systems.</td>
<td>2</td>
</tr>
<tr>
<td>1.2</td>
<td>This figure visually shows an example of the problem that is being tackled in this work.</td>
<td>5</td>
</tr>
<tr>
<td>2.1</td>
<td>Figure showing positive and negative bags used in MIL. A positive bag contains at least one positive instance and negative contains only negative instance.</td>
<td>17</td>
</tr>
<tr>
<td>2.2</td>
<td>MilBoost Algorithm</td>
<td>23</td>
</tr>
<tr>
<td>2.3</td>
<td>Plots for classification and localization performance across different configurations of multiple segment representation (Section 2.6.3).</td>
<td>37</td>
</tr>
<tr>
<td>2.4</td>
<td>Visualization of the weights learned by MS-MIL classifier.</td>
<td>44</td>
</tr>
<tr>
<td>2.5</td>
<td>Discriminative facial patches for pain detection as learned by our algorithm.</td>
<td>45</td>
</tr>
<tr>
<td>2.6</td>
<td>Example showing the performance of our algorithm for pain localization vs ground-truth frame labels (PSPI).</td>
<td>46</td>
</tr>
<tr>
<td>3.1</td>
<td>Shows a facial expression video modeled via an HMM, where hidden states.</td>
<td>54</td>
</tr>
<tr>
<td>4.1</td>
<td>We illustrate the inherent idea in Embedding Space based MIL approaches.</td>
<td>71</td>
</tr>
<tr>
<td>4.2</td>
<td>This graph highlights the advantage of learning discriminative concepts simultaneously.</td>
<td>84</td>
</tr>
<tr>
<td>5.1</td>
<td>Illustration of the proposed approach.</td>
<td>87</td>
</tr>
<tr>
<td>5.2</td>
<td>SGD based learning for LOMo.</td>
<td>92</td>
</tr>
<tr>
<td>5.3</td>
<td>Detection of multiple discriminative sub-events, discovered by LOMo.</td>
<td>100</td>
</tr>
<tr>
<td>5.4</td>
<td>Detections made by LOMo trained ($M = 3$) for classifying ‘happy’ expression on two expression sequences from Oulu-CASIA VIS dataset.</td>
<td>102</td>
</tr>
</tbody>
</table>
Figure 5.5. Performances of the methods for different values of $\lambda$ and PCA dimensions (best viewed in color). ................................. 106

Figure 5.6. Frames corresponding to latent sub-events as identified by our algorithm on different subjects ........................................ 107

Figure 5.7. Detections made by LOMo trained ($M = 3$) for classifying ‘disgust’ expression ............................................................ 109

Figure 5.8. Detection of multiple discriminative sub-events, discovered by LOMo ................................................................. 110
| Table 2.1. | Formulation of different soft-max functions along with $w_{ij}$ in each case. | 22 |
| Table 2.2. | Comparison of MS-MIL with different algorithms for pain classification in videos. | 31 |
| Table 2.3. | Comparison of MS-MIL with different methods for the pain localization. | 32 |
| Table 2.4. | Evaluation across different methods for generating multiple segments. Classification accuracy is reported in %. | 39 |
| Table 2.5. | Experiments on FEEDTUM dataset highlighting the generality of MS-MIL. | 48 |
| Table 3.1. | Results from experimental evaluation. | 65 |
| Table 3.2. | Results (mean % accuracy) from FEEDTUM. | 65 |
| Table 4.1. | Comparison of our algorithm (% accuracy) with different. | 81 |
| Table 4.2. | Evaluation (multiclass % accuracy) of Instance Space and Embedding Space based MIL algorithms on the Corel-2000 dataset [17] along with 95% confidence interval. | 82 |
| Table 4.3. | Evaluation (mean Area Under the ROC curve) on the UCSB Breast Cancer dataset [50]. | 82 |
| Table 5.1. | Comparison of LOMo with Baseline methods on 4 facial behavior prediction datasets using SIFT based facial features (see §5.4.1). | 98 |
| Table 5.2. | Comparison of the proposed approach with several state-of-the-art algorithms on three datasets. | 103 |
ACKNOWLEDGEMENTS

Finishing this thesis has been a long and strenuous journey but it was equally enjoyable and exciting. I want to first thank my grandparents—Mr. Harbans Lal Sikka and Mrs. Nirmal Kanta Sikka, and my parents—Rakesh Sikka and Asha Sikka, for their blessings and always seeking out the best in me. It is a result of their hard work and sacrifices that I could attend UCSD and finish my doctorate. Although my grandfather is not with us today, I know that his memories will always be with me and he would have been the happiest person on earth to see me graduate. I will always regret not being able to spend more time with him while I was away from my home. I want to also thank my brother—Sahil Sikka for his support and wish him good luck to his graduate studies. I also want to thank my extended family for their blessings. I am also grateful to all my school and undergraduate teachers who saw some potential in me and worked hard on me over the years.

I want to thank my advisor—Marian Bartlett, for believing in me and giving me the independence to pursue research ideas. It was always exciting interacting with her and being able to get more clarity on my work. I also thank her for being understanding and always putting my interests in front of other issues. I want to thank my co-advisor—Truong Nguyen, for supporting and motivating me and helping with smooth completion of my PhD. I want to thank my committee—Sanjoy Dasgupta, Bhaskar Rao, Zhuowen Tu and Nuno Vasconcelos, for their constructive suggestions on my work. Next I want to thank my collaborator and good friend—Gaurav Sharma, with whom I started working in November 2015 and this collaboration resulted in some good work. I learned a great deal from his technical expertise and understanding of computer vision, and hope to keep working with him in the near future. I want to thank my collaborator and good friend—Abhinav Dhall, with whom I wrote my first conference paper at our lab and learned
a lot from him. He has not only been a helpful collaborator but also a good, trusting
and long-term friend. I want to thank all my labmates at MPLAB— Mohsen Malmir,
Tingfan Wu, Jake Whitehall, Suchi Sathyanarayana, Gwen Littlewort, Karmen
Dijkstra, Crane Huang, Walter Tablott, Javier Movellan and Joshua Susskind, for
both their technical and non-technical support. In particular, I want to thank
Tingfan Wu, who helped and taught me about the field when I joined the lab. I
also want to thank Mohsen Malmir, who has not only been an excellent labmate
but a good friend and it would have been difficult to finish the thesis without him.
I want to thank my NTAP project collaborators— Jeannie Huang, Damaris Diaz,
Ken Craig, Matthew Goodwin, Virginia De Sa, Alex Ahmed and Rick Palumbo. In
particular, I want to thank Jeannie for not only being an excellent collaborator but
also managing the project and at the same time letting me work on my own thesis.
I want to thank my lab administrators— Luis Palacious and Robert Buffington
without whom I would not have managed to run experiments so comfortably.

Finally I want to thank my friends— Prarthana Majumdar, Ritwik Giri,
Varish Diddi, Chetan Verma, David Nunez, Yogesh Kumar, Mariza Suarez, Keyur
Karandikar, Mehrad Yazdani, Praneeth Naidu, Shailendra Singh, Ravi Satazoda,
Seemeen Karimi, Jyoti Joshi, and others. This journey would not have been
that fun and smooth without support from my friends. I will indeed miss all the
moments— long and philosophical chats, teas and coffees, visits to beach, trying
food at local restaurants and playing FIFA with my local friends in San Diego. Apart
from technical expertise, staying in San Diego was a great learning experience and
needless to say I will always miss San Diego and UCSD.

Chapter 2, in full, is a reprint of the material as it appears in Image and
Vision Computing. Sikka, Karan; Dhall, Abhinav; Bartlett, Marian., copyright
Elsevier, 2014. The dissertation author was the primary investigator and author of

xii
this paper.

Chapter 3, in full, is a reprint of the material as it appears in Computer Vision and Pattern Recognition Workshops 2015. Sikka, Karan; Dhall, Abhinav; Bartlett, Marian., copyright IEEE, 2014. The dissertation author was the primary investigator and author of this paper.

Chapter 4, in full, is a reprint of the material as it appears in British Machine and Vision Conference 2015. Sikka, Karan; Giri, Ritwik; Bartlett, Marian., BMVA Press, 2015. The dissertation author was the primary investigator and author of this paper.

Chapter 5, in part, is a reprint of the material as it appears in Computer Vision and Pattern Recognition Conference 2016. Sikka, Karan; Sharma, Gaurav; Bartlett, Marian., copyright IEEE, 2016. The dissertation author was the primary investigator and author of this paper.
VITA

2010 Bachelor of Technology, Indian Institute of Technology Guwahati
2012 Masters of Science, University of California San Diego
2016 Doctor of Philosophy, University of California, San Diego

PUBLICATIONS


AWARDS
Best Paper Award at EmotiW2013 Challenge, International conference on Multimodal Interaction 2013


Best Student Paper Honorable Mention Awards, IEEE Automatic Face and Gesture 2013

FIELDS OF STUDY

Major Field: Electrical Engineering (Computer Vision)
ABSTRACT OF THE DISSERTATION

Latent Dynamic Space-Time Volumes for Predicting Human Facial Behavior in Videos

by

Karan Sikka

Doctor of Philosophy in Electrical Engineering (Intelligent Systems, Robotics and Control)

University of California, San Diego, 2016

Truong Nguyen, Chair

Enabling machines to understand non-verbal facial behavior from visual data is crucial for building smart interactive systems. This thesis focuses on human behavior analysis in videos. Previous state-of-the-art methods generally employed global temporal pooling approaches that, (i) assume presence of a single uniform event spanning the sequence, and (ii) discard temporal ordering by squashing all information along the temporal dimension. In this dissertation we focus on two specific modeling challenges unaddressed by previous approaches. First issue
is training with weak labels that only provide video-level annotations and are much cheaper to obtain than fine (frame-level) annotations. The second concerns modeling temporal dynamics during prediction, as facial expressions are dynamic actions with sub-events. We propose to tackle these issues by proposing methods based on Weakly Supervised Latent Variable Models (WSLVM) and evaluate them on real-world spontaneous expressions.

We begin with addressing these challenges by combining Multiple Instance Learning (MIL) framework and Multiple Segment representation (MS-MIL). MS-MIL can simultaneously classify and localize target behavior in videos despite training with weak annotations. However, this method lacks the capability to explicitly model multiple latent concepts or global temporal order. We address this issue in the next chapter by explicitly modeling temporal orderings by learning an exemplar Hidden Markov Model for each sequence. This algorithm models dependencies between segments but is limited in its modeling capacity due to the use of generative modeling. Chapter 4 extends MIL to learn multiple discriminative concepts in a novel formulation for joint clustering and classification. This algorithm shows consistent performance improvement but does not capture temporal structure.

We finally present a unified learning framework that combines the strengths of the previously proposed algorithms in that it (i) addresses weakly labeled data (ii) learns multiple discriminative concepts, and (iii) models the temporal ordering structure of the concepts. This method is a novel WSLVM that models a video as a sequence of automatically mined, multiple discriminative sub-events with a loose temporal structure. We show both qualitative and quantitative results highlighting improvements over state-of-the-art algorithms by jointly addressing weak labels and temporal dynamics.
Chapter 1

Introduction

We have witnessed an explosion of visual data in the past few years with the availability of cheaper and high resolution cameras, and also an increase in data storage and computing power. These factors have also led to a significant progress in the field of computer vision for analyzing and making sense of visual data. Notably, these algorithms are no longer restricted to academic labs but are being used by humans in their daily lives. For example, computer vision algorithms are being used in image and video search engines [116, 81, 86], organizing photos [135, 84], autonomous cars [39, 10], surveillance systems [19, 88] etc. An emerging sub area in computer vision is designing human-centric vision systems that can understand and predict human behaviour from multiple sensors. This includes using visual data to predict human pose [122, 121], gaze [14, 62], activity [118, 88, 34], identity [156, 84] and facial expressions [112, 23, 96, 141]. The research in area has also been rampant due to an increase of interactions between humans and machine with smart devices such as mobiles. In this work our focus is on tackling the problem of predicting human facial behaviour in videos. These algorithms have several applications such as drowsiness prediction during driving [129], analyzing customer behavior in advertisements or movies [92] and long-term pain monitoring in clinical settings [105] (visual examples are shown in Fig. 1.1).
1.1 Human Behavior Prediction in Videos

Compared to images, videos contain appearance changes over time, and thus require learning procedures different from images to effectively exploit the rich spatial and temporal structure [56, 34, 139, 26]. Videos are quite relevant in the case of human facial behavior as facial expression are dynamic events that may be described in phases\(^1\) such as neutral, onset, apex [23, 114, 9, 67, 15]. Further temporal information could be vital to understand non-verbal communication behaviors such as understanding vs agreeing [73, 108, 9, 152, 48, 103].

---

\(^1\)We use ‘phases’, ‘sub-events’ and ‘concepts’ interchangeably to denote basic (atomic) temporal events that describe an activity.
Many previous state-of-the-art approaches to facial behavior prediction in videos employed global spatio-temporal approaches that proceed by first extracting visual features from a sequence and then pooling them across the temporal dimension [56, 155]. This is followed by training and testing with a standard classifier such as Support Vector Machine (SVM). Over the years there has been significant research in visual features such as local Spatio-Temporal features with Bag of Words [113, 116] or Fisher Vector encoding [98], trajectory aligned features [132], Local Binary Patterns [82], and recently followed by advances in Convolutional Neural Networks [149, 48] that jointly learn the low-level features as well as the final classifier.

Before discussing the challenges addressed in this work, we would like to mention about spontaneous and posed expressions. Much of the prior work on facial behavior employed datasets that were collected in controlled laboratory environments and consisted of posed expressions [9, 70]. The subjects are generally asked to act or mimic a displayed expression for recording visual data. The collected expressions have less variability and are relatively easier to recognize compared to spontaneous or real-world expressions. Natural facial expressions are spontaneous, rather than posed, and they are often collected in relatively uncontrolled settings such as in hospitals and during natural conversations [71, 105, 103]. These expressions are generally harder to recognize since they can be subtle, involve distraction such as out-of-plane head movements, verbal conversations, and have higher inter-subject variability [9, 71]. Moreover, owing to their short durations they can be harder to detect in unsegmented videos. Our aim is to test the algorithms proposed in this work on not only posed but also spontaneous expressions.

This work focusses on addressing two key issues with previous approaches:
1. **Weak labels**: In many real-world applications, labels are available to indicate whether a target action occurs in the video clip, but the precise timing of human action within the clip is unlabeled. Particularly for learning human states such as pain, videos are often provided a single label for subject state or stimulus condition, without labeling the muscle actions in the video or when they occur. These video-level labels are incomplete in the sense that they inform about ‘what’ happened in the video but not ‘when’ it happened. As a result, not all frames in a video might belong to the class described by the video label. This property introduces an ambiguity in the data that is not addressed by temporal pooling based methods. Moreover, spontaneous expressions are often sparse and localized in time. In such cases, pooling operations might result in summarizing information from irrelevant parts of the video leading to noisy features. A classification algorithm should try to address the issue of weak labels \[108, 80, 4, 128\].

2. **Temporal dynamics**: Facial expression are dynamic events and are often composed of multiple phases. The duration and intensity of these phases may vary for each instance and being able to correctly detect, align and match them could aid in final prediction especially for spontaneous expressions. However, previous approaches based on temporal pooling don’t preserve any temporal ordering since they squash all the temporal information across a single dimension, leading to loss of any temporal ordering \[67, 34, 128, 15, 144, 125\].

The algorithms proposed in this thesis revolve around these two modeling issues and try to solve them using variants of *latent* variable statistical models. The specific problem setting is to learn video classification models while training with
Figure 1.2. This figure visually shows an example of the problem that is being tackled in this work. We aim to learn a statistical model for video classification with weakly-labelled training data. These weak labels only provide coarse or global information about the video i.e. they provide information about what is happening in a video but not where it is happening.

only video-level or weak labels, as shown in Fig. 1.2.

1.2 Overview and Organization

Multiple Instance Learning with Multiple Segments: As the first step, we begin by addressing these issues using a Weakly Supervised Learning (WSL) framework in Ch. 2. We propose a novel algorithm that combines Multiple Instance Learning (MIL) [128] for tackling weak labels, with a Multiple Segment Representation to incorporate temporal dynamics. The algorithm is referred to as Multiple Instance Learning with Multiple Segments (MSMIL) [108]. This algorithm jointly classifies and localizes the target facial action within the videos. In the Multiple Segment representation, each instance is a video segment, rather than a single image. Owing to weak labels we have no apriori information about the duration or starting point of the target expression. We address this aspect by extracting multiple video segments at multiple starting points and temporal scales. This benefits the MIL based learner by increasing the hypothesis space and results in both better classification and localization performance (shown both qualitatively and quantitatively in Ch. 2). We show that this model outperforms
both baseline and state-of-the-art algorithms on the problem of pain classification. Localization performance is similar to algorithms trained with full frame-by-frame supervision.

**Exemplar Hidden Markov Models:** The MSMIL algorithm was able to address the issue of weak labels but didn’t completely exploit and model temporal dynamics. This was the case since MIL only models a unique discriminative action and assumes independence between different segments in a video. In the second work, introduced in Ch. 3, we propose a sequential strategy that models each video as a set of observations at regular intervals such that these observations are generated by an underlying probabilistic model [125, 15]. In particular we model each sequence with a Hidden Markov Model (HMM) that factorizes each sequence as a progression of multiple sub-events (modeled as Gaussians) with an underlying first order temporal structure. However, different from previous methods that learn an HMM for each class, we train an HMM model for each example. In order to do so we propose to use fully Bayesian HMM models that use priors to learn with small amounts of data (per video). We then compute distances between these exemplar models using a probabilistic kernel that effectively measures the similarity between both static and dynamic components of the individual HMMs. We use these distances to learn a kernel SVM classifier for each class. Combining generative and discriminative modeling results in an effective algorithm that learns an explicit spatio-temporal structure for each sequence and performs comparable to state-of-the-art methods relying on a larger feature space. We refer to this algorithm as **Exemplar Hidden Markov Models** (EHMM) [107].

**Joint Clustering and Classification for Multiple Instance Learning:**
Up to this point, we have described two WSL based algorithms that use MIL and exemplar HMMs respectively for learning from weakly labelled video sequences.
The MSMIL work modeled a single discriminative event, while EHMM modeled a video as multiple events and their dynamics using generative modeling. Borrowing taxonomy from the WSL literature [3], the MSMIL algorithm belongs to the class of Instance Space (IS) based approaches since it first learns to classify a single discriminative instance followed by extending the classification to a bag (or a video). On the other hand EHMM belongs to Bag Space (BS) based methods since it first embeds a bag in a manifold and uses that representation for classification. Although EHMM was able to learn multiple concepts, the dimensionality of visual descriptors that it could use was restricted due to the need to estimate the parameters of the generative models, and descriptors were assumed to have a Gaussian distribution. Also it is generally preferable to jointly learn the latent structure with the classifier as this results in a latent structure that is aligned with the final goal. In the next work (described in Ch. 4) we propose a novel algorithm that extends the standard MIL algorithm to learn multiple concepts. This approach jointly learns latent structure with the classifier, and is less restrictive on the dimensionality of the visual descriptors or their distributions than EHMM. We refer to this algorithm as Joint Clustering and Classification for Multiple Instance Learning (JC²MIL) [110]. This method belongs to the class of Embedding Space (ES) based algorithms that discover latent concepts using unsupervised or semi-supervised algorithms and use them to embed a bag in a concept space. The proposed algorithm is distinguished from previous ES based methods by using a novel formulation that jointly learns the concepts and the bag classifier. We show that this algorithm yields state-of-the-art results on several MIL datasets compared to IS based and other ES space based methods.

**Latent Ordinal Models for Facial Analysis in Videos**: Both JC²MIL and EHMM proposed to extend standard WSL algorithms to model richer latent
structures by either modeling multiple concepts or temporal structure or both. However, both of these algorithms suffer from assumptions that limit their modeling capabilities. For instance JC$^2$MIL does not model any temporal structure, while EHMM has a generative component that restricts its modeling capabilities, and does not learn discriminative latent structure, since the latent parameters are not jointly learned with the classification parameters. In the next work we propose an algorithm that combines the strengths of the previous algorithms by modeling both (i) multiple sub-events that occur sparsely in time, and (ii) temporal dynamics by learning a prior on the temporal ordering of sub-events. In contrast to HMMs or Hidden Conditional Random Fields, this is a loosely structured Latent variable model that only penalizes the relative ordering between events. Further it doesn’t need to model the background class. We refer to this method as Latent Ordinal Model (LOMo) and introduce it in Ch. 5 for facial analysis in videos. [112]. This model combines the strengths of each of the previous methods by (i) learning discriminative concepts, (ii) able to learn and detect multiple concepts, and (iii) modeling the loose temporal ordering structure of the concepts. Further we show that this model outperforms previous state-of-the-art algorithms and relevant baselines, namely global temporal pooling and standard MIL. We also show several qualitative examples that highlight that LOMo is capable of learning temporal phases for the target expression in a video.
Chapter 2

Weakly Supervised Pain Localization and Classification with Multiple Segment Learning

2.1 Introduction

Pain is one of the most challenging problems in medicine and biology and has substantial eco-social costs associated with it [27]. It has been estimated that there might be more than 30 million people in USA with chronic or recurrent pain [124]. Also nearly half of Americans seeking treatment from a physician report pain as their primary symptom. The United States Bureau of the Census estimated the total cost for chronic pain to exceed $150 billion annually in year 1995-96 [124, 27]. Thus there has been a significant research effort in improving pain management over the years.

Identifying pain among patients is considered critical in clinical settings since it is used for regulating medications, long-term monitoring, and gauging the effectiveness of a treatment. Pain assessment in most cases involves patient self-report, obtained either through clinical interview or visual analog scale (VAS) [27]. For the latter case the nurse asks the patient to mark his pain on a linear scale with ratings from 0 to 10, denoting no-pain to unbearable-pain. The fact that
VAS is easy to use and returns a numerical rating of pain has made VAS the most prevalent pain assessment tool. However VAS suffers from a number of drawbacks such as subjective differences, and patient idiosyncrasies. Therefore it cannot be used for unconscious or verbally-impaired patients [20] and may suffer from high individual bias. These drawbacks have led to a considerable research effort to identify and quantify objective pain indicators using human facial expression [123]. However most of these methods entail manual labeling of facial action units or evaluations by highly trained observers, which in most cases is time consuming and unfit for real-time applications.

Over the years there has been a significant progress in analyzing facial expressions related to emotions using machine learning (ML) and computer vision [60]. Most of this work has focused on posed facial expressions that are obtained under controlled laboratory settings and differ from spontaneous facial expression in a number of ways [21, 9]. We refer our readers to a survey on automatic facial expression recognition (AFER) by Bartlett et al. [9] that has identified the difficulties faced by AFER on spontaneous expressions. A major challenge of spontaneous expressions is temporal segmentation of the target expressions. Videos may exist in which the target emotion or state was elicited, but the onset, duration, and frequency of facial expressions within the video are unknown.

A significant contribution to in research on spontaneous expressions was the introduction of UNBC-McMaster Shoulder Pain dataset [71] that involves subjects experiencing shoulder pain in a clinical setting. This dataset was provided with two levels of annotations for measuring pain- (1) per-frame pain ratings based on a formula applied to Action Unit (AU) annotations, and (2) per-video pain ratings as measured by experts (see Section .2.5.1). This work utilizes the per-video pain ratings for training a binary pain classification system. Pain localization is then
evaluated using the per-frame pain ratings based on AU labels, which are more costly to obtain. Thus our setting is such that each video is labeled for presence or absence of pain, but there is no information about the location or duration of facial expressions within each video. This setting is referred to as weakly labeled data and poses a challenge for training sliding window classifiers and further limits the performance of the standard approach of obtaining fixed length features through averaging and training a classifier. Previous approaches [73, 5] follow a common paradigm of assigning each frame the label of the corresponding video and using them to train a support vector machine (SVM). Pain is detected in a video if the average output score (distance from separating hyperplane) of member frames is above a pre-computed threshold. Such approaches suffer from two major limitations: (1) not all frames in a video have the same label, (2) averaging output scores across all the frames may dampen the signal of interest. This paper proposes to address these challenges by employing multiple instance learning (MIL) learning framework [128].

MIL is an approach for handling ‘weakly labeled’ training data. In such cases the training data only specifies the presence (or absence) of a signal of interest in the data without indicating where it might be present. For instance in the case of pain vs no-pain detection, a sequence label only specifies if a subject is/not in pain without any details regarding the time point or duration of pain. Other techniques for tackling weakly labeled data includes part-based models [31] and latent models such as pLSA and LDA [131]. Most of these approaches try to identify the signal of interest by inferring the values of some latent variables while minimizing a loss function. MIL was introduced to address the problem of weakly supervised object detection [128] [35]. Compared to other approaches, MIL offers a tractable way to train a discriminative classifier that avoids complex inference procedures. MIL has
been successfully employed for face recognition from video [128] and more recently has been proposed for handling labeling noise in video classification [59].

This work focuses on detecting spontaneous pain expression in video when given only sequence level ground-truths. The phrase *detection* is used throughout the paper to denote the joint tasks of pain classification and localization in time. Explicitly, classification refers to predicting absence/presence of pain in a video, while localization refers to predicting pain/no-pain at the frame level. The novelty of this work lies in combining MIL with a dynamic extension of concept frames, into a novel framework called **Multiple-Segment Multiple Instance Learning (MS-MIL)**. Our major contributions are as follows:

1. Inherent drawbacks in previous approaches for pain detection in videos are identified and a pipeline has been proposed to address these concerns. The most salient feature of our approach is that it can jointly classify and localize pain by using only sequence level labels (Section. 2.2).

2. For addressing the demands of the pain detection task, we propose to represent each video as a bag containing multiple segments which are modeled using MIL. The multiple segment based representation and MIL are able to address spontaneous expressions, such as pain, that can have uncertain locations, durations and occurrences (Section. 2.4).

3. The performance of MS-MIL is compared on the detection task with other competitive algorithms. We also perform systematic evaluation to highlight the contribution of multiple segment representation and MIL, in MS-MIL, separately. These results indicate the advantage of using the MS-MIL approach along with some interesting insights. (Section. 2.6)

The problem of detecting pain through facial expressions in general includes
many challenges and this work is trying to focus on a particular aspect of the problem. Other challenges in objective pain measurement include differences between acute and chronic pain, as well as differences in personality including pain catastrophizing, which may affect the intensity of pain expression. We are undertaking a separate study to begin to address some of these factors [43].

2.2 Related Work and Motivation

The first computer vision work on automatic pain detection in videos on the UNBC Mc-Master Pain dataset was by Ashraf et al. [5]. Their approach started by first extracting AAM based features from each frames and using these to cluster the frames in order to create a training data with size that is manageable by a SVM. Following this, each of these clustered frames were assigned the label of their corresponding sequence and used to train a linear SVM. Finally during prediction each test-frame was assigned a score based on its distance from separating hyperplane. Then a test-video was predicted to be in pain if the average score of its member frames exceeded a threshold. Lucey et al. [73] extended this work by borrowing ideas from the related field of visual speech recognition and proposed to compress the signal in the spatial rather than temporal domain using the Discrete Cosine Transform (DCT). Lucey et al. [73] used the system in [5] as their baseline system and showed significant improvement in performance using their idea.

Previous works didn’t address the ambiguity introduced by weakly labeled data, and each member frame was assigned the label of the sequence. Such approaches lead to a lower performance compared to the case when ground-truth for each frame is known [5][6]. We address this particular concern by proposing to use MIL (in-place of SVM) which has been designed specifically to handle weakly labeled data.
Secondly, [73] highlighted that incorporating the dynamics of the pain signal is difficult since there is no information about the number of times pain expressions can occur or their location and duration in a sequence. Following this, [73] suggested to add temporal information by appending adjacent frames onto the frame of interest, as input to the SVM [90]. [73] tested this idea of appending adjacent frames in their paper, however they found that their performance degraded. One possible explanation is that SVM classifiers are not well suited to weakly labeled training data and may suffer from mislabels when the data is in this form.

Motivated by the last idea we propose to incorporate temporal dynamics by representing each sequence not as individual frames (as done earlier) but as sets of frames, referred to as ‘multiple segments’. The benefits of such a representation are reaped by using MIL, which can efficiently handle data in such form. Since MIL handles data as bags, we can visualize every sequence as a bag containing multiple segments. Multiple segments (MS) has two fold advantages: (1) it allows pain expression to have random duration and occurrence, and (2) it incorporates temporal information by pooling across multiple frames in a segment. Thirdly, the earlier work performed prediction for each sequence them feel persecuted using the average decision score of its frames. Such an approach may not be optimal in all situations since the averaging operation tends to dampen the signal of interest. The MIL framework employed in this work avoids this limitation by using the max operation to predict the label of a bag based on the posterior probability of its instances (see Section. 2.3).

Another potential approach to the problem of pain detection comes from the classical approach to action recognition from computer vision literature [56, 137]. This approach is based on BoW architecture and composed of three steps: feature extraction, encoding features using a dictionary of visual words and pooling with $l_1$
normalization. Since each video is represented as a fixed length vector, we shall refer to these techniques as global-feature based approaches. [137] has provided a systematic evaluation of different components of this pipeline on two human action datasets. These techniques are known to work well for problems with uniform actions that span the entire video such as CK+ facial expression dataset [70] or KTH human action dataset [55]. However their performance falls down when actions have high intra-class variations and are localized in the video, which is true for the pain detection problem as well. We also found this hypothesis to be true during our experiments and attribute it to the argument that pooling features across the entire video tends to reduce discriminative ability of the features.

In a recent paper [119] Tax et al. explored the question of whether it is always necessary to fully model the entire sequence, or whether the presence of specific frames, called ‘concept frames’, might be sufficient for reliable detection of facial expressions. In their study two different approaches for AFER were investigated: (1) modeling full sequences using approaches such as Hidden Markov Models and Conditional Random Fields, and (2) modeling only certain frames, for AU detection in sequences. The author in [119] also suggested that for modeling only particular key frames, algorithms such as MIL are required and investigated one such approach. Through extensive experiments the authors showed that for reliable classification, modeling certain key frames is sufficient compared to modeling the entire sequence. A limitation of ‘concept frames’, however, is that they do not incorporate temporal information, which could potentially be exploited by learning algorithms such as MIL (and to some extent SVM [114]).

The present paper takes a leap forward by proposing a dynamic variant of ‘concept frames’. Here we extend the idea of ‘concept frames’ to ‘concept segments’ consisting of multiple frames. These ‘concept segments’ can be thought of as
localized sub-expressions that contain the expression of interest in a sequence. We propose that Reliable detection of facial expression can be achieved by detection of key localized segments using tailored algorithms such as MIL. [114] explored a segment based approach, called k-Seg SVM, and employed a structured-SVM to detect temporal events (AU segments in their case). Our work differs from this work in several respects, most notably that [114] is a completely supervised algorithm requiring location information in the training data, whereas the approach presented here operates on weakly labeled data. Authors in [24] represented a video by concatenating features from 6 key-frames (segments) that were identified by clustering based on the output of an emotion classification task. We overcome the possible limitations of this work by allowing the videos to be represented by a variable number of segments of varying lengths and performing classification by explicitly spotting the segment containing target expression.

2.3 MIL

The general machine learning paradigm involves finding a classification function that minimizes a loss function $\mathcal{L}(D, h(x))$ over training data provided as $N$ samples and their corresponding labels, $D = \{x_i, y_i\}_{i=1}^N$, where $x_i \in X$ and $y_i \in Y$. Rather than handing training data in the form of individual samples, the MIL paradigm is designed to handle problems involving training data in the form of bags, $B = \{X_i, y_i\}_{i=1}^N$, where $X_i = \{x_{ij}\}_{j=1}^{N_i}$, $y_i \in Y$ and $N_i$ are the number of instances in $X_i$. Since this work deals with only binary classification problems, the output space $Y \in \{-1, 1\}$. Such problems occur frequently in computer vision since it is easier to obtain a group label for the data compared to individual labels and such labels can also suffer from annotator bias and noise [59]. Recently several works have adopted MIL to address these concerns in domains such as handling
Figure 2.1. Figure showing positive and negative bags used in MIL. A positive bag contains at least one positive instance and negative contains only negative instance.

label noise in video classification [59], face recognition in videos with subtitles [142], and object localization [35], etc.

As shown in Figure 2.1 the MIL framework defines two kinds of bags, positive and negative, in a similar fashion to positive and negative instances in traditional machine learning. A bag is a positive bag if it contains at least one positive instance, while a negative bag contains no positive instance.

We have employed Multiple Instance Learning based on boosting (MilBoost) algorithm proposed by Viola et al. [128] for this work. In the next two sections we shall give an overview of Friedman’s gradient boosting framework [32], which is the backbone of MilBoost. This will be followed by the description of MilBoost.

2.3.1 Gradient Boosting

We shall define the gradient boosting in the realm of traditional learning framework and then discuss its extension to the MIL framework.

Boosting involves constructing a strong classifier \( H_T(x) \) by iteratively combining many weak classifiers \( h_t(x) \), where the subscript \( t \ (t = 1, \ldots, T) \) represents the
index of the classifier added at the $t^{th}$ iteration. All weak classifiers are constrained to belong to a certain family of functions $\mathcal{H}$, such as stumps or trees.

$$H_T(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$$ (2.1)

$$H_T(x) = H_{T-1}(x) + \alpha_T h_T(x)$$ (2.2)

Equation 2.2 can be seen as a numerical optimization strategy that iteratively minimizes a loss function $L(D, H_{T-1}(x))$ over training data $D$ by moving in certain optimal direction given by $h_T$. Under this strategy, the loss function at step $T$ can either be seen as a function of the current classifier $H_{T-1}$ or the parameters that define the family of functions $\mathcal{H}$.

Friedman suggested following the latter approach since it offers an intuitive way to solve the above optimization problem. $H_{T-1}(x)$ can be considered as a $n$ dimensional vector whose $i^{th}$ component is $H_{T-1}(x_i)$. Following this idea, the gradient descent strategy is employed to minimize the loss function by moving some steps in the direction of the negative-gradient of the loss function with respect to $H_{T-1}(x)$. This negative gradient is denoted by $w_i$ in Equation 2.3. In the remaining sections of this paper we shall refer to $w$ as weights and the rationale behind this will be evident in Section 2.3.2.

$$w_i = - \frac{\partial L}{\partial H_{T-1}(x)} \bigg|_{x=x_i}$$ (2.3)

Thus the gradient boosting framework prescribes to minimize the loss function by moving in the direction $w$ computed at each iteration. Since $H_T$ is a linear combination of $H_{T-1}$ and $w$, it would be smooth only when $w \in \mathcal{H}$. However it will be too idealistic to assume this in all cases. Friedman proposed to
tackle this problem by projecting $w$ over the function space $\mathcal{H}$ by finding the best approximation $h_t \in \mathcal{H}$ to $w$.

$$h_t = \arg\max_h \sum_{i=1}^N w_i h(x_i)$$  \hspace{1cm} (2.4)

We shall refer to Equation 2.4 as the ‘projection step’ and note that $h_t$ has the maximum correlation with $w$. Once $h_t$ is computed, step size $\alpha_t$ is found via a line search to minimize $L(D, H_T(x))$. In the next section we shall discuss how gradient boosting is extended to the MIL framework.

### 2.3.2 MilBoost

MilBoost combines the gradient boosting framework with the concept of MIL, where training data occurs as bags. As defined in Section 2.3, the $i^{th}$ bag is denoted by $X_i$ and the $j^{th}$ instance inside it is represented as $x_{ij}$. The posterior probabilities over bags and instances are defined as:

$$p_i = Pr(y_i = 1|X_i)$$ \hspace{1cm} (2.5)

$$p_{ij} = Pr(y_{ij} = 1|x_{ij})$$ \hspace{1cm} (2.6)

We shall be using the original formulation defined in [128] for the loss function given by the negative log-likelihood:

$$\mathcal{L} = -\sum_i^N t_i \log p_i + (1 - t_i) \log (1 - p_i)$$ \hspace{1cm} (2.7)

where $t_i = 1$ if $y_i = 1$ and $t_i = 0$ if $y_i = -1$.

This formulation for the loss function seems intuitive since the only information available about a MIL dataset is label information for each bag ($y_i$). We
lack any information about the probabilities (or labels) of individual instances \( p_{ij} \). These instance probabilities can also be seen as latent variables, whose values are inferred during the boosting process [7].

MIL assumes that a positive bag contains at least one positive instance. Hence the probability of a bag being positive \( p_i \) is defined in terms of individual instances as:

\[
p_i = \max_j (p_{ij}) \tag{2.8}
\]

Since the max function is not differentiable, a number of differentiable approximations to the max function have been proposed for MilBoost [128, 142, 7]. In this work we shall refer to these approximations as softmax functions \( g(p_{ij}) \). The most common choice of soft-max function in earlier works is noisy-or (NOR). A major disadvantage with NOR is that it deviates from the max function as the size of the bag increases, which we shall refer to as 'bagsize-bias'. To illustrate this shortcoming we consider a toy example which consists of two bags \( B_1 \) and \( B_2 \) of sizes of 3 and 5. The instance probabilities for these bags are given by \( B_1 = [.15 .15 .2] \) and \( B_2 = [.15 .15 .15 .2] \). As is evident, the max for both cases is .2, however the NOR formulation yields the maximum as .45 and .53 respectively. This observation clearly highlights the bagsize-bias associated with NOR. Such a problem is critical for cases where bag sizes might differ across training examples and ours is such a case since the number of frames per sequence vary from 60 – 600. Thus in this work we have addressed this problem by employing another soft-max function called Generalized mean (GM), which is known to be a better approximating function than NOR [7].

The instance probabilities \( p_{ij} \) for instance \( x_{ij} \) are obtained by the application of a sigmoid function over the raw classifier score \( h_{ij} \):
\[ p_{ij} = \sigma(h(x_{ij})) \] (2.9)

As described in Section 2.3.1, the negative gradient of the loss-function (for instance \( x_{ij} \)) is obtained as:

\[ w_{ij} = -\frac{\partial L}{\partial h_{ij}} \] (2.10)

We can easily calculate \( w_{ij} \) by exploiting the chain rule of differentiation and calculating each components as:

\[ w_{ij} = \frac{\partial L}{\partial h_{ij}} = \frac{\partial L}{\partial p_i} \frac{\partial p_i}{\partial p_{ij}} \frac{\partial p_{ij}}{\partial h_{ij}} \] (2.11a)

\[ \frac{\partial L}{\partial p_i} = \begin{cases} \frac{1}{p_i} & t_i = 1 \\ \frac{1-t_i}{1-p_i} & t_i = 0 \end{cases} \] (2.11b)

\[ \frac{\partial p_i}{\partial p_{ij}} = \frac{\partial g(p_{ij})}{\partial p_{ij}} \] (2.11c)

\[ \frac{\partial p_{ij}}{\partial h_{ij}} = \frac{\partial \sigma(h_{ij})}{\partial h_{ij}} = \sigma(h_{ij})(1 - \sigma(h_{ij})) \] (2.11d)

Next we explain the rationale behind referring to the negative instance-wise gradients \( (w_{ij}) \) as weights, using the NOR softmax function as an example. From Table 2.1, \( w_{ij} \) for the NOR soft-max function is defined as \( w_{ij} = \frac{1-p_i}{p_i} p_{ij} \) for a positive bag and \( w_{ij} = -p_{ij} \) for a negative bag. Thus these weights describe, (1) the label of the bag containing instance \( x_{ij} \) and, (2) the importance of the instance in learning procedure, by being high for an instance that lies in a positive bag but has a low classifier score and vice-versa. The idea of weighting instances during learning is common in boosting procedure [32].

As described in Section 2.3.1, the next step involves finding a new weak learner \( (h(x_{ij})) \) that has the highest correlation with the weights \( w_{ij} \) using the
Table 2.1. Formulation of different soft-max functions along with $w_{ij}$ in each case.

<table>
<thead>
<tr>
<th>Soft-max</th>
<th>$g(p_{ij})$</th>
<th>$w_{ij}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOR</td>
<td>$1 - \prod_j (1 - p_{ij})$</td>
<td>$\frac{t_i - p_i}{p_i}$ $p_{ij}$</td>
</tr>
<tr>
<td>GM</td>
<td>$\left( \frac{1}{n} \sum_k p_{rij} \right) \frac{1}{\sum_j p_{rij}}$</td>
<td>$\frac{1 - p_i}{1 - p_i}$ $p_{ij}$ $p_{rij}$ $p_{rij} - 1$ $\sum_j p_{rij}$</td>
</tr>
</tbody>
</table>

projection step (Equation 2.4). This work employs binary decision stumps as weak learners, which perform classification by assigning a threshold to a single feature and are a common choice in boosting frameworks [128]. Thus $\mathcal{H}$ belongs to the class of decision stumps. A simple mathematical formulation has been provided in Borris et al. [7] on how Equation 2.4 (the projection step) can be transformed into:

$$h_t = \arg\min_h \sum_{ij} [h(x_{ij}) \neq sgn(w_{ij})] w'_{ij}$$  

(2.12)

where $[.]$ is the Iverson bracket, $w'_{ij} = \frac{|w_{ij}|}{\sum_j |w_{ij}|}$ and $sgn(l)$ is the signum function.

Equation 2.12 is a general formulation for any learning algorithm that has training data with binary labels $sgn(w_{ij})$ and weights $w'_{ij}$. Thus we can easily find a function $h_t(x_{ij})$ at $t^{th}$ iteration that has the highest correlation with $w_{ij}$ by using training procedure for a decision stumps. All the steps of the MilBoost algorithm are mentioned in a sequential order in Algorithm 2.2.

### 2.4 Multiple Instance Learning based on Multiple Segments (MS-MIL)

#### 2.4.1 Overview

Each sequence $S_i$ is represented as a bag containing many segments or subsequences $\{s_{ij}\}_{j=1}^{N_i}$, where $N_i$ is the number of segments in sequence $S_i$. Temporal consistency is maintained inside a segment $s_{ij}$ by restricting it to contain only
contiguous frames (see Section 2.4.3), \( s_{ij} = \{f_i^k, f_i^{k+1}, \ldots, f_i^{N_{ij}-k-1}\} \), where \( k \) represents the time index (in the video) of the first frame inside segment \( s_{ij} \), \( f_i^k \) represents the \( k^{th} \) frame in the sequence \( S_i \) and \( N_{ij} \) is the number of frames in subsequence \( s_{ij} \). Thus a sub-segment \( s_{ij} \) is characterized by length of the segment (number of frames) \( N_{ij} \) and the time index \( k \) of first frame in the video. Two approaches are outlined in Section 2.4.3 for constructing multiple segments- (1) overlapping temporal scanning windows, and (2) multiple clustering. Depending upon the approach the number of frames inside a segment can either be fixed (in scanning windows) or sequence-dependent (in multiple clustering). Also the frames inside two different segments are allowed to overlap.
The only information available about a sequence during training is whether it has a pain expression i.e. $y_i = 1$ or not i.e. $y_i = -1$. We shall give a brief overview of the entire algorithm here.

**Representation:** The feature extraction process for a frame shall be denoted by a mapping $\phi_{Fr} : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^d$ that maps frames in image space $\mathbb{R}^{m \times n}$ to a $d$-dimensional vector space $\mathbb{R}^d$. The feature representation for a subsequence (or segment) is represented as a mapping $\phi_S : \mathcal{S} \rightarrow \mathbb{R}^d$ that transforms subsequences in space $\mathcal{S}$ to a $d$-dimensional vector space.

**Training:** Training data in the form of bags is trained using the MilBoost framework described in Section 2.3.2. This process yields a classifier $H_T : \mathbb{R}^d \rightarrow \mathbb{R}$. The number of iterations/weak-learners for MilBoost have been empirically set to 100 in our experiments.

**Prediction:** Suppose we have a test sequence $S_i = \{s_{i1}, ..., s_{iN_i}\}$. Each subsequence $s_{ij}$ is assigned a posterior probability $p_{ij}$ using the trained classifier $H_T$ and a sigmoid function $\sigma$ as:

$$p_{ij} = \sigma(H_T(\phi_S(s_{ij}))) \quad (2.13)$$

Here $\phi_S$ is the feature mapping for a sub-sequence.

The posterior probability of test sequence $S_i$ is predicted by using a soft-max function, as described in Section 2.3.2, over instance probabilities:

$$p_i = g(p_{ij}) \quad (2.14)$$

**Avoiding Local-Minima:** MilBoost algorithms can often overfit and converge to local minima. This issue is more critical for problems such as pain detection.
since theoretically the algorithm can converge even after learning a single instance of pain expression in a sequence, since the loss function is defined over bags. In such cases the learned function won’t be able to generalize well over unseen data. Hence we draw parallel ideas from bagging predictors proposed by Brieman [13], in which multiple versions of a predictor are combined to get an aggregated prediction. They showed improvement for predictors that are unstable/get caught up in multiple local minima. Since the problem formulation is very similar to ours, we also ran MilBoost over multiple initializations and bootstrapped data (random 90% subset). The final predictions for each segment were obtained by averaging the predictions $p_{ij}$ made from multiple MilBoost classifiers. Using this approach we found an improvement in predictions, and moreover this procedure allowed us to report results that would be reproducible. Based on our experiments we opted to run MilBoost 30 times. In practice we found that any number about this size or larger worked equally well.

**Pain Localization**: The prediction process estimates the posterior probability of each segment $s_{ij}$ in $S_i$. For assigning posterior probability to any frame in the sequence, we first identify the segments containing that frame. Following this, the frames are assigned a score based on their proximity to the center of that segment. We employ a hamming window, pivoted at the center of the segment, to assign a smoothly varying score to different frames in a segment. Since a frame could belong to multiple segments, it is assigned the maximum score from all these segments. In mathematical notations, the probability of frame $f_i^k$ in pain is predicted using the following formula:

$$p_{f_i^k} = p(y = 1|f_i^k) = \max_j (\tilde{w}(s_{ij}) \times p_{ij}|f_i^k \in s_{ij})$$  \hspace{1cm} (2.15)
where $\tilde{w}(s_{ij})$ is the hamming window function centered at the middle frame of segment $s_{ij}$. $p_{fk}$ is a discrete probability measure since it is bounded by 0 and 1 since $\tilde{w}(s_{ij}) \in (0, 1]$ and $p_{ij} \in [0, 1]$. Secondly $\sum_y p(y|f^k) = 1$. Thus our algorithm not only yields the probability for a sequence but also the probability for each frame that can be used to localize painful expression frames in a video using just sequence-level labels.

### 2.4.2 Bag of Words based Representation (BoW)

Recently computer vision has witnessed significant research in BoW models and their extensions, and as a result they have been applied across multiple domains. Sikka et al. [113] presents a survey of different BoW Architectures for AFER. They identified many advantages of BoW based approaches over previous approaches to AFER based on Gabor wavelets, or local binary patterns, passed directly through a classifier and have proposed a state-of-the-art feature pipeline through experimental analysis.

We employed the system proposed in [113] for the feature extraction and image representation. This representation consists of a spatial pyramid of level 4 on top of highly discriminative multi-scale dense SIFT (MSDF) features, which are encoded using LLC encoding followed by max-pooling. We also employed a separate dataset ($CK+ [70]$) for building a codebook (size 200 in this case) for encoding features. By using a separate dataset for creating the codebook, the feature extraction process is completely independent of the dataset. Our experiments yielded that MSDF features at two scales are sufficient for this problem and hence extracted MSDF features with window sizes of 4 and 8 and strides of 2 pixels. As mentioned in Section 2.4, the feature extraction operation using BoW is denoted as a mapping $\phi_{Fr}$. We refer readers to [113] for more information about
feature extraction and image representation in the BoW model including empirical comparisons of alternative feature extraction methods for AFER.

2.4.3 Multiple Segment (MS) Representation

This work defines a segment as a subset of an original sequence that contains only contiguous frames. Thus a sequence is represented as a bag of segments which are allowed to overlap. As highlighted in Section 2.2, the motivation behind the MS representation is that it allows random onset of pain expression, incorporates dynamic information, and can be efficiently handled by the MIL framework. It is assumed that for a sequence labeled as pain, at least one of the segments will contain a painful expression, and such a positive segment is referred to as a ‘concept segment’.

**Construction:** We propose two ways to generate multiple segments. A naive procedure is to run overlapping temporal scanning windows at multiple scales across the sequence and represent each subset of frames as a segment. This idea is motivated by the traditional approach in computer vision of running multi-scale scanning windows prior to a detection task. This idea has been exploited in previous work on weakly-supervised object localization [31] [128]. A parallel approach for generating multiple segments was explored in [35], where an image was segmented into many clusters using the idea of multiple stable segmentation. Each segmentation was obtained by varying the parameters of normalized cuts (referred to as Ncuts) [35]. We explored an analogous approach by clustering the frames in a sequence using Ncuts. Since we wanted to restrict a segment to contain only contiguous frames, the weight/similarity matrix used in Ncuts was defined to incorporate the similarity between the time index of two frames along with their feature similarity. Each element of this weight matrix $W_i(r, s)$ defines the similarity
between frames $f^r_i$ and $f^s_i$ of sequence $S_i$:

$$
W(r, s) = \exp \left( - \frac{\| \phi_{F_r}(f^r_i) - \phi_{F_r}(f^s_i) \|_2^2}{\sigma_f} - \frac{\| t_r - t_s \|_2^2}{\sigma_t} \right) \tag{2.16}
$$

where $t_r$ refers to time index of frame $f^r_i$.

Once the segments are constructed using either of the two approaches, it is important to represent them as fixed-length vectors while also preserving temporal information. [73] have highlighted that an elegant way of doing this is to append features from adjacent frames. We employed this idea along with max feature pooling, proposed for AFER in [113], for feature extraction. This process is represented as a mapping $\phi_S : S \rightarrow \mathbb{R}^d$ that maps a segment $s_{ij} = \{f^k_i, f^{k+1}_i, \ldots, f^N_{ij-k-1}\}$ belonging to set $S$ to a $d$-dimensional vector space and can be shown as:

$$
\phi_S(s_{ij}) = \max_k(\phi_{F_r}(f^k_i) | f^k_i \in s_{ij}) \tag{2.17}
$$

The idea of using a max operation for temporal pooling has also been explored in spatio-temporal deep learning approaches [120]. Also a number of recent works [101, 133, 113] have highlighted the performance advantages of the max pooling operation compared to average pooling.

### 2.5 Experimental Design

#### 2.5.1 Dataset

Our experiments employed data from the UNBC-McMaster Pain Shoulder Archive that was distributed to the research community in [71], and included 200 sequences from 25 subjects. Each subject was undergoing some kind of shoulder pain and was asked to perform a series of active and passive movements of their
affected and unaffected limbs. Active tests were self-initiated shoulder movements and in passive tests the physiotherapist was responsible for the movement. For complete details of the experimental settings we refer the readers to [71]. These sequences were then coded on a number of levels by experts. The coding of interest to this work was the Observer Pain Intensity (OPI) rating that was assigned to each sequence on a level of 0 (no-pain)−5 (strong pain) by an independent observer trained in identification of pain expressions. Following the protocol proposed in [73] [5], labels were binarized into 'pain' and 'no pain' by defining training instances with OPI\(\geq 3\) as the positive class (Pain) and OPI\(= 0\) as the negative class (No-Pain). Only those subjects were included in our experiments who had a minimum of one trial with an OPI rating of 0 (no pain) and one trial with an OPI rating of either 3, 4 or 5 (pain). Intermediate pain intensities of 1 and 2 were omitted, per the protocol in [73] [5]. This yielded 147 sequences from 23 subjects for our experiments. Since this work addressed two joint tasks i.e classification and localization of pain, two different performance metrics were employed to evaluate each tasks separately.

### 2.5.2 Performance Metrics

**Classification:** The classification task focuses on pain predictions at video-level. Experiments were conducted in a leave-one-subject-out cross-validation strategy. Thus there was no overlap between subjects in the training and testing data. For reporting the results, we followed the strategy employed in [73] [5], where they reported total classification rate or accuracy, which refers to the percentage of correctly classified sequences, computed at Equal Error Rate (EER) in the Receiver Operation Curve (ROC).

**Localization:** The localization task focuses on pain predictions at frame-level. This task was evaluated by employing the Prkachin and Solomon pain
intensity index (PSPI) that combines intensities of 4 Action units (AUs) from Facial Action Coding System (FACS) [91]. In particular PSPI combines the intensities of four “core” AUs for pain which are brow lowering (AU4), orbital tightening (AU6 and AU7), levator contraction (AU9 and AU10) and eye closure (AU43) [71]. The UNBC McMaster dataset provided FACS expert codes and PSPI metrics for each frame. We would like the readers to note that our algorithm used only OPI labels (sequence-level groundtruth) for training, while the PSPI labels were solely used for evaluation. The localization performance was evaluated across two sub-tasks, as explained below, with experiments conducted in leave-one-subject-out fashion.

The first task was designed to predict presence/absence of pain in each frame and compare these predictions against binarized PSPI score (where PSPI $> 0$ means pain). A similar idea of evaluating localization performance, when training with only sequence-level groundtruth, was also explored in [6]. The first metric for this frame based pain classification experiment was classification accuracy computed at EER in the ROC curve. Several previous works focusing on detection [114] have noted that metrics based on ROC curve are designed for balanced binary classification rather than detection tasks, and hence are unable to take into account the effect of the proportion of positive to negative samples. Thus in this work we also incorporated maximum $F1$ score (given by $\frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}$) for evaluating pain detection task. The $F1$ score is known to give a trade-off between high recall rates and accuracy for predictions [114].

The second task measured how well the the per-frame classification scores can predict PSPI pain intensities. This was accomplished by measuring the correlation between predictions and PSPI pain intensities for each frame. We opted for Spearman’s rank correlation [53] instead of Pearson correlation since the PSPI score occurs as ranked values in the range of $1 – 16$. For these experiments we
Table 2.2. Comparison of MS-MIL with different algorithms for pain classification in videos.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy (%) (at EER)</th>
<th>#subjects</th>
<th>#samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lucey et al. [73]</td>
<td>80.99</td>
<td>20</td>
<td>142</td>
</tr>
<tr>
<td>Ashraf et al. [5] (shown in [73])</td>
<td>68.31</td>
<td>20</td>
<td>142</td>
</tr>
<tr>
<td>MS-SVM_max</td>
<td>77.17</td>
<td>23</td>
<td>147</td>
</tr>
<tr>
<td>MS-SVM_avg</td>
<td>71.73</td>
<td>23</td>
<td>147</td>
</tr>
<tr>
<td>BoW+Avg+SVM [137]</td>
<td>66.30</td>
<td>23</td>
<td>147</td>
</tr>
<tr>
<td>BoW+Max+SVM [137]</td>
<td>81.52</td>
<td>23</td>
<td>147</td>
</tr>
<tr>
<td>MS-MIL</td>
<td>83.7</td>
<td>23</td>
<td>147</td>
</tr>
</tbody>
</table>

reported Spearman’s rank correlation coefficient, which is calculated between two observations \(X_i\) and \(Y_i\) as:

\[
\rho = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}}
\]

s.t \(-1 \leq \rho \leq 1\). \(\rho = 0\), \(\rho = 1\) and \(\rho = -1\) correspond to no-correlation, perfect correlation and perfect negative correlation respectively.

2.6 Results and Discussion

2.6.1 Performance Evaluation of Pain Classification

MS-MIL was compared with related algorithms for the problem of pain classification. We divided these related algorithms into 3 groups and have provided implementation details for each of these in the following Subsections. The result for MS-MIL is reported for the best configuration of the multiple segment representation, which was empirically estimated to be a combination of segments of length 31, 41 and 51 frames, generated using overlapping scanning windows (see Section 2.6.3).
Table 2.3. Comparison of MS-MIL with different methods for the pain localization.

<table>
<thead>
<tr>
<th>Method</th>
<th>Localization Accuracy (%)</th>
<th>Correlation</th>
<th>Max-F1</th>
<th>Video-level Classification accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MS-SVM(_{max})</td>
<td>72.64</td>
<td>.390</td>
<td>.471</td>
<td>77.17</td>
</tr>
<tr>
<td>MS-MIL</td>
<td>76.08</td>
<td>.432</td>
<td>.523</td>
<td>83.70</td>
</tr>
<tr>
<td>frame-SVM(^1)</td>
<td>70.47</td>
<td>.385</td>
<td>.477</td>
<td>84.78</td>
</tr>
<tr>
<td>frame-SVM(^2)</td>
<td>66.76</td>
<td>.282</td>
<td>.403</td>
<td>73.91</td>
</tr>
</tbody>
</table>

Previous State of the Art

MS-MIL was first compared with previous state of the art algorithms by Ashraf et al. [5] and Lucey et al. [73] as shown in Table. 2.2. We have reported results for Ashraf et al. as were reported by authors in [73] using their own implementation. number of subjects and samples, we would like to highlight that our experiments have been conducted with a larger number of samples (147 vs 142 and 84 in [73] and [5] respectively). Secondly there isn’t much difference between the number of samples used in [73] and ours (5/147 samples), in which case the results could be comparable to some extent.

Global-feature based Approaches

MS-MIL was also compared for pain classification performance with two global-feature based approaches constructed using BoW [56, 137], as discussed in Section. 2.2. Global-feature based methods represent a video by a fixed length vector. Hence by themselves these methods can only be used for pain classification task and not for pain localization. We used the same frame features, constructed using BoW, as used in MS-MIL. These frame features were then pooled using average [56, 137] and max pooling [137] to obtain a fixed dimensional representation for the entire video. Following feature extraction, classification was performed using a linear SVM [137]. Depending on the pooling strategy, these approaches are referred
to as BoW+Avg+SVM or BoW+Max+SVM in Table. 2.2. These approaches serve as a good baseline since they are amongst the classical approaches for action classification in computer vision [89].

**Evaluating the Contribution of MIL**

We have argued the aptness of MIL to handle sequences represented as multiple segments compared to traditional ML algorithms. This argument was validated by using the same MS representation but replacing MIL with a linear SVM. All the segments in the training data were assigned the label of the sequence and used to train this SVM. This strategy, if not same, is in spirit similar to that employed in previous works ([73] [5]). Finally during prediction a combining rule was used to assign each sequence a decision score based on the score of its member segments [119]. We had explored two common combining rules, namely maxima (similar to MIL and used in [119]) and average ([73] [5]) and the corresponding SVMs are referred to as MS-SVM\(_{max}\) and MS-SVM\(_{avg}\). Table. 2.2 reports the accuracy for both SVMs with the same MS representation as used in MS-MIL.

**Overview of Pain Classification Task**

Although it could be argued that a direct comparison with previous algorithms for pain detection by [73] and [5] is not possible owing to a different number of samples, some inferences could still be made since the sample set differs by only a small amount of data. Firstly the results of [5] (as published in [73]) and [73] showed an accuracy of 68.31% and 80.99% respectively, compared to 83.7% performance of MS-MIL. Thus it could be argued that MS-MIL shows significant performance improvement over [5] and is comparable to (or better) than [73]. This improvement can be attributed to the algorithmic improvements that MS-MIL has over these approaches (see Section. 2.2). The two global-feature based approaches,
BoW+Avg+SVM and BoW+Max+SVM, yielded a performance of 65.22% and 78.26% respectively. Our argument that global-feature based approaches discard discriminative information as a result of pooling is supported by the observation that they have a lower performance compared to MS-MIL (65.2% and 78.26% vs 83.7%). Also these results provide additional support that max pooling is preferable to average pooling.

Lastly the argument that the MS representation is efficiently handled by MIL is validated by the comparison of MS-MIL with SVM applied to the MS representation as shown in Table. 2.2. Here MS-MIL outperformed both MS-SVM\textsubscript{avg} and MS-SVM\textsubscript{max} by a margin of at least 6% points. The results also indicate that MS-SVM\textsubscript{max} performs better than MS-SVM\textsubscript{avg} for all cases since the averaging operation is known to dampen the signal of interest (Section. 2.2).

### 2.6.2 Performance Evaluation of Pain Localization

We evaluated the localization performance of MS-MIL for two different sub-tasks of (1) predicting presence/absence of pain, and (2) predicting pain intensity using per-frame classification scores, as discussed in Section. 2.5.2. The SVM based MS-SVM\textsubscript{max} algorithm was selected for comparison with MS-MIL. Both algorithms used the same MS representation, which was a combination of segments of length 31, 41 and 51 frames generated using overlapping scanning windows (Section. 2.6.3).

We were also interested in performance comparison of MS-MIL with a system that was trained particularly for a frame-by-frame pain prediction task. This was accomplished by training a linear SVM over the same frame features as used in MS-MIL, using two versions of frame-level groundtruth. The first version, referred to as Frame-SVM\textsuperscript{1}, was trained using binarized PSPI labels (PSPI> 0 is pain). While for the second version, referred to as Frame-SVM\textsuperscript{2}, the frames were assigned
the label of the video that contained them. Thus Frame-SVM$^1$ represents a fully supervised algorithm with complete label information, and Frame-SVM$^2$ represents a weakly-supervised algorithm (such as MS-MIL). Both methods had the same experimental settings as MS-MIL. We handled the massive amount of data (around 35K frames) for this task by training the linear SVM in its primal form using LIBLINEAR SVM library [28]. The results from these experiments are shown Table. 2.3.

Although the primary interest in this Section is pain-localization performance, we have also reported video-level classification accuracy for each of these methods so as to supplement current analysis. For MS-MIL and MS-SVM$_{max}$, the classification accuracy is the same as that reported in Table. 2.2. For the two frame based algorithms (Frame-SVM$^1$ and Frame-SVM$^2$), the video scores were estimated by taking a $max$ over the scores of member frames, as was done for MS-SVM$_{max}$ in Section. 2.6.1.

It is evident from Table. 2.3 that MS-MIL outperforms all other algorithms across both pain localization tasks. The performance of Frame-SVM$^1$ was lower than MS-MIL as reported by pain localization metrics. This was contrary to our expectations since Frame-SVM$^1$ was trained on actual (binarized PSPI) frames labels compared to weak-labels used for MS-MIL. The possible reason for higher performance of MS-MIL could be the use of the MS representation in MS-MIL, that is able to achieve some degree of temporal smoothing. This also shows that MIL framework used in MS-MIL is able handle label ambiguity elegantly. However one cannot neglect the benefit of having complete frame labels, and this is evident in the classification accuracy of Frame-SVM$^1$ (84.78%), which is slightly above MS-MIL (83.7%) and surpasses its weakly-supervised counterpart (Frame-SVM$^2$) (73.91%) by a large margin.
The advantage of using the MS representation is also evident in the higher performance of MS-SVM\textsubscript{max} compared to Frame-SVM\textsuperscript{2}, where the two algorithms were trained on the same sequence-level labels but employed the MS and the frame representation respectively. It was also interesting to note that MS-MIL was able to achieve a correlation of .432 with the PSPI intensity when it was trained using only weak-labels in the form of video-level labels. Moreover this correlation was higher as compared to the correlation achieved by the supervised frame-by-frame algorithm Frame-SVM\textsuperscript{1} (.432 vs .385). Thus these results conclude that MS-MIL has a performance advantage over its weakly supervised counterparts as well as over supervised frame-by-frame algorithms.

We have also shown visualization for 2 cases in Fig. 2.6 to highlight the ability of our algorithm to localize pain. These visualizations compare the per frame posterior probability as predicted by MS-MIL against the PSPI index (Section. 2.5.2). In order to facilitate a direct comparison between probabilities and the PSPI index on the same vertical scale, the PSPI index was normalized in the range of [0, 1] by dividing by maximum PSPI score of 16 ([71]). These visualizations qualitatively support our claims that MS-MIL is capable of joint classification and localization of pain. It is evident from Fig. 2.6a that our algorithm is able to identify multiple occurrences of pain. Secondly the posterior probabilities predicted by MS-MIL seem to correlate well with the PSPI index. Fig. 2.6b shows a case of a pain sequence whose PSPI ground-truth score was zero across all frames but the observer rated the facial expression as showing pain (OPI= 3). Our algorithm, which was trained on observer ratings, was able to localize pain in this case. On further analysis we found that there was a FACS coding error for this particular sample. This is an intriguing example highlighting the advantage of using automatic computational methods compared to humans.
Figure 2.3. Plots for classification and localization performance across different configurations of multiple segment representation (Section 2.6.3).
2.6.3 How does the Multiple Segment Representation effect MS-MIL

The novelty of this work lies in combining multiple segment representation with MIL. In Section 2.6.1 we have already highlighted the advantage of MIL by replacing MIL with SVM (MS-SVM). Here our aim is to empirically evaluate the benefits of the MS representation in MS-MIL. We have tried to show this by analyzing the performance of MS-MIL across different configurations of the MS representation. Here we compare different lengths of the multiple segments in Section 2.4.3, we restricted ourselves to the use of multi-scale temporal scanning windows (Scan-wind) for generating MS for this experiment. (Two approaches for generating MS are evaluated independently in the next Section.)

Two parameters are required for Scan-wind: (1) window size, and (2) overlap between two windows. The overlap was fixed to 50% of the window size in all cases and the window size parameter was swept to generate results. The parameters were selected so as to cover a broad range of window sizes starting from short windows of length 10 frames to large windows of length 100 frames. This was done keeping in mind the large variation in video lengths and temporal extent of pain signal in the dataset. We had also tried several combinations of windows sizes to generate multi-scale MS and included results for the case having best performance for both classification and localization tasks. The results are shown in Figure. 2.3, with Figure. 2.3a showing plots for classification and localization accuracy metric, and Figure. 2.3b showing plots for correlation and $F_1$ score metric. These metrics are the same as those discussed in Section. 2.5.2.

Looking at the results in Figure. 2.3, it is evident that the performance across all metrics goes up as the window size is increased from 11 to 61 frames and thereafter the performance starts to fall down. These results are quite intuitive to
Table 2.4. Evaluation across different methods for generating multiple segments. Classification accuracy is reported in %.

<table>
<thead>
<tr>
<th>Setting</th>
<th>MS-MIL</th>
<th>MS-SVM\textsubscript{max}</th>
<th>MS-SVM\textsubscript{avg}</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ncuts</td>
<td>scan-wind</td>
<td>Ncuts</td>
</tr>
<tr>
<td>short</td>
<td>78.26</td>
<td>78.16</td>
<td>78.26</td>
</tr>
<tr>
<td>medium</td>
<td>82.61</td>
<td>83.70</td>
<td>78.26</td>
</tr>
<tr>
<td>long</td>
<td>80.43</td>
<td>82.61</td>
<td>79.34</td>
</tr>
<tr>
<td>combine</td>
<td>83.70</td>
<td>83.70</td>
<td>77.17</td>
</tr>
</tbody>
</table>

interpret as features pooled over small windows will not encode sufficient temporal information, showing lower performance. While for very large window sizes, pooling tends to pack too much information in the features making them less discriminative (as discussed in Section 2.6.1). The algorithm performed consistently high across window sizes of lengths 41 to 61 frames. One possible reason for this observation could be that most subjects in the dataset present facial action related to pain within intervals of length 41 – 61 frames. Finally the result corresponding to combination of MS of length 31, 41 and 51 frames yielded the highest results across all the metrics. Although it had the same classification accuracy as segments from 41 – 61 frames, it showed significant improvement in the metrics evaluating localization task. Thus one could argue the advantage of using multi-scale MS for the pain detection task since it tries to capture all possible pain expressions in a scale independent manner.

Overall these results empirically support the advantage of the multiple segment representation in MS-MIL for the problem of pain detection. Moreover it is evident that the advantage of MS is best reaped at segments of medium length or a combination of these. It was also interesting to empirically verify our hypothesis regarding the importance of pooling over segments of the right length as discussed in Section 2.2.
2.6.4 Approaches for Generating Multiple Segments (Normalized Cuts vs Scanning Windows)

Two methods for generating the multiple segment representation were discussed in Section 2.4. The first method was Ncuts that generated segments through clustering. Since the number of frames differs across videos, we determined the number of clusters for Ncuts by fixing the minimum number of elements (frames) in a cluster. The values of other parameters were kept constant for all experiments ($\sigma_t = 100$ and $\sigma_f = 10k$). The second approach is the multi-scale temporal scanning windows. We employed the same parameters for multi-scale temporal scanning window as taken in Section 2.6.3. For both cases the parameter of interest is the length of the segment to be used.

To systematically study the effect of these approaches on performance, four scenarios were considered by varying the length of segments in our MS representation. These configurations were selected to cover a wide variety of temporal scales. They are referred to as:

1. **short**- segments of short length (11 frames)
2. **med**- segments of medium length (41 frames)
3. **long**- segments of long length (81 frames)
4. **combine**- combination of segments of length 31, 41 and 51.

We have also included results from MS-SVM$_{\text{max}}$ and MS-SVM$_{\text{avg}}$, along with results from MS-MIL in Table 2.4, for making further inference.

From results in Table 2.4 it is evident that MS-MIL has a low performance for short and long settings and high for medium and combine settings, for both Ncuts and Scan-wind. This observation is in line with the results presented in the
previous Section. We didn’t observe any clear trends for the SVM based approaches. It is also interesting to note that the performance of both MS-SVM$_{max}$ and MS-MIL is similar (78.26%) for Ncuts with the short setting. Thus it is possible that there isn’t much difference between MS-MIL and MS-SVM$_{max}$ for short segments since features pooled over short-segments are less informative. Finally MS-MIL shows a consistent performance of 83.7% for combine segments for both Ncuts and Scan-wind, highlighting a consistent benefit of multi-scale MS. Although Ncuts and Scan-wind show similar classification performance, Ncuts lags behind Scan-wind on the localization task. This is because Ncuts employs windows/segments that are sparsely located in time, compared to dense sampling in Scan-wind, and localization in the former case will only be approximate (see Section.2.4).

2.7 Visualizing the Classifier and Benefits of Bagging

Since different expressions are associated with different facial muscles, we wanted to visualize the facial regions contributing most towards pain detection. To accomplish this we selected the weights and indices of the weak-learners learned during the gradient boosting procedure (Section. 2.3.2). Since our features are based on the spatial-pyramid BoW framework [113], each of these indices represents a word that lies in a localized image patch at one of the 4 scales (see Section.2.4.2). Next we formed an intensity image by back-mapping each index to its facial patch, and then aggregating weights over all facial patches. We further converted the intensity image into a RGB image with the color encoding the magnitude of the weights. The intensity image corresponding to the MS-MIL classifier is shown in Figure. 2.5a with the color encoding shown in Figure. 2.5b. We have also shown an intensity image in Figure. 2.5c overlaid with the image of a subject to aid in
visualization of discriminative facial regions. Please note that we have shown both the overlaid and non-overlaid intensity images since the overlaid intensity image could include some extra intensity owing to the texture on subject’s face image.

We have tried to interpret the visualization in Figure 2.5c by relating regions, identified important for pain detection, to Action Units previously known to be associated with pain [91].

1. The red-most region near the lower-corner of right eye seems to be picking up levator contraction and naso-labial furrow changes associated with AU 9 and AU 10 respectively. This region also seems to capture orbital contraction movements related to AU 6.

2. The eye corner (left-eye) seems to be picking up eye squinting (AU 7 and also AU 43).

3. The chin area seems up to be picking up a chin raise related to AU 17 or mouth opening related to AU 25.

Thus it is evident that the visualization showing discriminative facial regions (learned by the algorithm) seems to correlate well with the prior knowledge about Action Units related to pain.

These visualizations have also been used for highlighting the advantage of using bagging step in MS-MIL. The bagging step works by training multiple MilBoost predictors with different initialization and bootstrapped data as discussed in Section 2.4. The final classification score for a segment is obtained by averaging scores from each predictor. To emphasize the contribution of the bagging step, we visualize the weights learned by 3 individual predictors in Fig. 2.4a, Fig. 2.4b and Fig. 2.4c, and the final average predictor obtained after averaging (bagging)
in Figure. 2.4d respectively. These visualizations have been generated using the same procedure as discussed in previous paragraph. It is evident from these visualizations that the weights learned by individual classifiers have high variance and the bagging step helps by averaging and lowering the variance in weights. It is also interesting to note that these results support the argument, posed in several works that analyzed bagging theoretically [87], that bagging can be seen as a kind regularization operation. The weight patterns also reveal the discovery that we made during our experiments that MS-MIL, being a latent variable model, is unstable with respect to initializations and prone to local-minima. And this instability is the vital component that causes bagging to work well in our case as noted in [13].

2.8 Experiments on FEEDTUM dataset

From extensive experiments it is evident that MS-MIL gives appreciable results on the UNBC-McMaster pain dataset. However it could be argued by a machine learning practitioner that the reason for good results could be over-fitting by MS-MIL for this particular setting of features and dataset. Thus we evaluated MS-MIL on a different dataset of spontaneous expressions. We compared the performance of MS-MIL with its global-features based counterpart on a different problem with different set of features. The rationale behind opting for a different problem and different set of features is to exhibit that MS-MIL can also be generalized to a different yet connected problem.

This experiment was conducted on a subset of FEEDTUM facial expression dataset [130] that consists of videos of 19 subjects (320 videos) showing six basis emotions, namely- anger, disgust, fear, happiness, sadness and surprise. The dataset exhibits natural (or spontaneous) expressions, which were elicited by showing the
Figure 2.4. Visualization of the weights learned by MS-MIL classifier. Fig. 2.4a, Fig. 2.4b and Fig. 2.4c show the weights learned by 3 individual classifier, while Fig. 2.4d shows the weights learned by final classifier obtained after bagging(Section. 2.4). Color coding is shown in Fig. 2.5b.

subjects several carefully selected video stimulus. This is different from datasets like CK+ [70], where the subjects were asked to move specific facial muscles. The rationale behind selecting this dataset is that the subjects exhibit spontaneous expressions and the videos are unsegmented, yielding no information about the onset, duration and frequency of the facial expressions. Thus AFER on this dataset poses similar challenges as were discussed in the motivation for current work (see Section .2.1).
Figure 2.5. Discriminative facial patches for pain detection as learned by our algorithm (Section 2.7). Fig. 2.5a shows an intensity image with hue of the color encoding importance of each facial region as discovered by MS-MIL. The colorbar is shown in Fig. 2.5b, with blue and red denoting lowest and highest weights respectively. Fig. 2.5c shows the same intensity image overlaid over a subject’s image for better visualization.

2.8.1 Experimental setting and Results

The experiments were conducted in leave-one-subject-out fashion. The classification was performed in 1-vs-all format and thus involved solving a different binary classification task for each of the 6 expressions. Different from BoW features, we opted for features based on the displacement of facial landmarks points [97]. 49 landmark points were obtained for each frame by using a state-of-the-art facial feature detector based on supervised gradient descent [145]. Displacement features for each frame were obtained by subtracting $x$ and $y$ coordinates of the landmark points in that frame from the landmark coordinates in the first (neutral frame) in that video. It is been shown in the expression recognition literature that this subtraction from a person-specific neutral face is vital to normalize landmark features and remove subject-dependent bias [97]. The final feature dimension of 98 is obtained by concatenating displacements of both $x$ and $y$ coordinates.
Case 1: Pain expression with multiple occurrences

Case 2: Case where PSPI index showed no pain but MS-MIL could correctly localize pain.

Figure 2.6. Example showing the performance of our algorithm for pain localization vs ground-truth frame labels (PSPI).

In order to highlight the efficacy of MS-MIL, we compared the performance of the following two implementations:

1. geom.+MS-MIL: This is essentially MS-MIL with landmark displacement features. We extracted multiple segments of length 9, 15, and 21, using the overlapping scanning window approach as discussed in Section 2.4.3. The features inside a multiple segment were obtained by averaging the landmark features of all of the frames inside that segment. This is in spirit similar to the averaging operation used for pooling BoW features (see Section 2.4.3). We also tried using operators like max instead of averaging to obtain the fixed length features and didn’t see significant change in results.

2. geom.+MilBoost: This version is similar to the global-feature based approaches as discussed in Section 2.6.1. The fixed length features that represent each video are obtained by averaging the landmark features over all the frames. Once the features are obtained, MilBoost is used as the binary classifier.
It is important to note that MilBoost functions as a generic classifier while working with training data organized as positive and negative instances (as for geom.+MilBoost). Also fixing the classifier allows us to perform a fair comparison for highlighting the performance different with (geom.+MS-MIL) and without (geom+MilBoost) multiple segments.

All the experimental settings for MilBoost have been kept same as those of MS-MIL (see Section), except the number of weak learners which is set to 60 (feature dimension is 98) since we found the performance to saturate approximately at 60 weak learners. The threshold for assigning a positive label to a video based on the probabilistic output (see Equation 2.14) was set to a standard value of 0.5. The performance metric for this experiment is mean classification accuracy over the 6 expression classes.

The results are shown in Table 2.5. MS-MIL gives a mean classification accuracy of 84.55(±0.98) compared to 81.78(±1.31) of MilBoost. Thus it is evident from the results that MS-MIL utilizing multiple segments outperforms its fixed length feature counterpart even for a expression classification problem on a different dataset and with different feature set. Such a result was expected since FEEDTUM is a spontaneous expression dataset and holds the assumption that not all frames in a video exhibit the expression of interest. Thus it is evident from this experiment that MS-MIL is capable of generalizing to other classification problems with similar assumptions.

2.9 Conclusion

This paper proposed a novel approach to the problem of detecting spontaneous expressions of pain in videos, based on multiple instance learning (MIL). We presented a novel framework called multiple-segment multiple instance learning
Table 2.5. Experiments on FEEDTUM dataset highlighting the generality of MS-MIL. Classification accuracy is reported in % for six different facial expressions. p-value for the paired t-test between the two classification accuracies is 0.0162, showing that the difference is significant.

<table>
<thead>
<tr>
<th>Algo</th>
<th>Anger</th>
<th>Disgust</th>
<th>Fear</th>
<th>Happiness</th>
<th>Sadness</th>
<th>Surprise</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>geom.+</td>
<td>79.86</td>
<td>80.95</td>
<td>87.88</td>
<td>83.98</td>
<td>77.20</td>
<td>80.73</td>
<td>81.78</td>
</tr>
<tr>
<td>MilBoost</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>±1.31</td>
</tr>
<tr>
<td>geom.+</td>
<td>84.28</td>
<td>85.41</td>
<td>85.01</td>
<td>86.36</td>
<td>83.04</td>
<td>83.18</td>
<td>84.55</td>
</tr>
<tr>
<td>MS-MIL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>±0.98</td>
</tr>
</tbody>
</table>

(MS-MIL) which incorporated with MIL a dynamic extension of concept frames, referred to as multiple segments (MS). This work targeted the joint problem of, (1) classifying the expression in a video as pain/no-pain (classification), and (2) predicting pain in each frame (localization). The problem is particularly challenging since the algorithm is trained using only sequence-level groundtruth, which provides no information regarding the presence/absence of pain for a given frame.

The paper first highlighted some limitations of previous approaches and how they motivated the design of the proposed algorithm. Next, an overview of multiple instance learning was presented, followed by the description of the proposed approach, MS-MIL. Rigorous experiments were conducted to compare the performance of MS-MIL against related algorithms on both classification and localization tasks on the UNBC Mc-Master Shoulder Pain dataset. The benefits of our algorithm were evident by having significant performance advantages compared to its counterparts across both tasks. Following this we also empirically validated the contributions of both multiple segments representation and multiple instance learning in MS-MIL independently. The results from these experiments supported our argument that MS-MIL is able to tackle the twin challenges of (1) label ambiguity, and (2) incorporating temporal information, in current problem efficiently. To highlight that our algorithm is actually learning meaningful facial
structures for pain detection, we showed the visualization for the discriminative facial patches that were learned by our algorithm. We further showed that these discriminative facial patches were related to Action Units known to be associated with Pain.

From our experiments it is evident that pain detection in videos is a challenging problem owing to the variability associated with how pain can be expressed by different subjects at different times and scenarios. The present algorithm is able to do an appreciable job of not only detecting pain, but also identifying the temporal location of pain expressions within the video clip. The most salient contribution of this work is that pain localization is achieved without any human intervention and employing only sequence level labels.

2.10 Acknowledgement

Support for this work was provided by NSF grant IIS-0905622 and NIH grant NIH R01NR013500. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. We would also like to thank Tingfan Wu, Gwen Littlewort, Mohsen Malmir, Deborah Forster and Ritwik Giri for helpful discussions.

Chapter 2, in full, is a reprint of the material as it appears in Image and Vision Computing. Sikka, Karan; Dhall, Abhinav; Bartlett, Marian., Elsevier, 2014. The dissertation author was the primary investigator and author of this paper.
Chapter 3

Exemplar Hidden Markov Models for Classification of Facial Expressions in Videos

3.1 Introduction

Automatic facial expression recognition (AFER) enables machines to understand a form of human behavior, and can be used towards building intelligent systems [23, 71, 77]. Research efforts in the last decade have led to significant improvements in AFER and have also opened new research avenues [23]. In particular, AFER research is transitioning from expression recognition in images [113, 82] to recognition in videos [108, 155]. Image based methods, sometimes referred to as static approaches, utilize visual features from a snapshot (such as the apex expression in a sequence [155, 140]) for predicting facial expressions [108, 82]. In image-based approaches, expression dynamics are not explicitly incorporated in the features or in the classifier, and instead are analyzed in the time series of the output. In addition, image-based approaches often assume temporal segmentation of facial expressions for training. Since facial expressions are dynamic events, video based approaches that incorporate dynamics earlier, in the features or in the model, may have an advantage over image-based approaches, and have been shown to
outperform their image based counterparts on several AFER problems [67, 148].

Video based approaches can be roughly categorized into space-time and sequential approaches. In space-time techniques, localized features (in space and time) such as Bag of Words (BoW) [113], fiducial point positions [97, 47], and LBPTOP [155], are first extracted across the entire video. This is followed by applying spatio-temporal pooling operations (such as average or maximum [108]) over the entire video [47] or fixed grids [155] to obtain a fixed length vector representation [108]. These fixed-length vectors are then passed to a classifier. Owing to the use of discriminative classifiers such as Support Vector Machines (SVM), we shall refer to these approaches as Discriminative space-time (Disc-ST) methods. Disc-ST methods can extend image-based features to video based methods by summarizing the per-frame features over the entire video using summary statistics [108]. This is known as pooling across the temporal dimension. Although Disc-ST methods such as LBPTOP are used frequently for AFER, we highlight two inherent issues that can result in a performance loss. These are:

1. Discriminative power: Pooling features across the entire video (or pre-defined grids) works well for pre-segmented video clips. However, the approach can lack discriminative power for unsegmented video, which can contain more than one expression as well as neutral periods. For instance the video clip shown in Figure. 3.1, is composed of angry expression segment as well as neutral expression. Since the final features summarize the entire video, the performance might degrade when some sub-segments are non-informative. For the problem of pain classification in unsegmented videos, Sikka et al. [108] showed that Disc S-T approaches yielded a lower performance than methods utilizing information from task specific sub-segments.
2. Temporal alignment: Facial expression in a video can be characterized as a dynamic event that passes through several states. A recognition pipeline would ideally match corresponding states, which requires temporal alignment between sequences [67]. A fixed feature pooling strategy ignores this correspondence, and doesn’t capture temporal relationships between these states.

Sequential approaches [18, 140, 102] present an alternate strategy for analyzing facial expressions in videos. Such models first convert a video into a sequence of observations at regular intervals and then analyze the sequence for presence of action-specific features and their dynamics. This work focuses on Hidden Markov Models (HMMs) that can describe a facial expression as a dynamic event comprising of several sub-events (apex, onset and offset) with specific temporal relationships between them [125]. Since a latent state variable is associated with each observation, HMMs naturally define a temporal segmentation of the video [125, 18]. The training routine of HMM consists of estimating class-conditional probability distributions for each class. A test video is then classified into the class that corresponds to maximum posterior probability.

Because HMMs offer a form of temporal segmentation and alignment, HMMs provide modeling advantages that may lead to better classification performance than Disc-ST methods for AFER. However HMMs, being generative models, are often weaker classifiers than discriminative models since estimating probability distributions is a harder problem than solving the classification problem directly [79]. As a result, generative HMMs generally have a lower performance compared to Disc S-T approaches [67, 140], and are seldom used despite their modeling capabilities. A possible solution towards overcoming the disadvantages of HMMs is to estimate the model parameters within a discriminative learning framework.
Approaches based on this idea, such as Hidden Conditional Random Fields, have previously been used for AFER [93]. This paper follows an alternative solution that focuses on estimating similarity kernels for probability distributions [46, 44]. Such approaches allow the possibility to measure meaningful distances between probabilistic models describing non-vectorial data.

We argue that when embedded in a discriminative classification framework based on probabilistic kernels, dynamical models such as HMMs may be effective for the facial expression problem space. An approach from the machine learning literature, generative kernels, combines the modeling advantages of HMMs with the discriminative advantages of SVMs in a principled fashion in an exemplar (or similarity) based classifier [11, 46]. Each example is first abstracted into an individual HMM, which models the spatio-temporal characteristics specific to that example. A distance metric, referred to as a probabilistic kernel [44, 46], is then used for computing distances between two Exemplar-HMMs. The probabilistic kernel can also be visualized as a distance between two data points lying on some non-linear manifold of HMM models. These distances are then used as an input to a kernel SVM. In this work we employed the Probabilistic Product Kernel (PPK) [46] since it provides a closed-form solution for HMMs and can also be interpreted as estimating distances between the temporal segments (or states) of two videos while taking into account the transition probabilities. Exemplar-HMMs with probabilistic product kernels have been shown to be effective for clustering motion capture data [46] and recognizing handwritten words [94]. Here we explore this class of models in the problem space of facial expression recognition in video. We shall refer to this approach as Exemplar-HMMs.

The performance advantage of our approach was exhibited through an evaluation on both posed and spontaneous facial expression datasets. The AFER
problems in these datasets ranged from predicting basic emotions to predicting whether a commercial presented over the internet was liked. On each of these AFER problems, our approach achieved state-of-the-art results compared to its Disc-ST counterparts and also in comparison to recently proposed AFER algorithms that exploit facial expression dynamics.

### 3.2 Related Work

This work is motivated by the idea of using model-based similarity for measuring distances between non-vectorial data such as time-series or sets of vectors [46, 11, 44, 94]. The distances are computed by (i) mapping each vector set into a probability distribution, and (ii) using a probabilistic measure to compute distances between example specific models. Availability of distances between examples allows one to define a kernel and use discriminative classifiers such as SVMs. The primary benefit underlying these approaches is the possibility of including example-specific structural information during classification. This hybrid discriminative-generative framework has shown its advantages over holistic feature based approaches on...
several problems such as handwriting recognition [94], gene classification [46], and shape recognition [11], among others.

Previous work has identified the importance of exploiting spatio-temporal structure for classifying facial expressions. In a recent paper by Liu et al. [67], the authors proposed mid-level representations, referred to as Expressionlets, that were obtained by aligning localized spatio-temporal features from an input video with a universal Gaussian Mixture Model (GMM) model. The authors argued that via localized alignment, their approach (STM-ExpLet) allows flexible spatio-temporal range among low-level features and is able to achieve improvements over Disc-ST approaches. Our approach is motivated on a similar argument as [67], however it differs on two points. Firstly, we incorporate temporal information via the model (instead of low-level features) and secondly the temporal flexibility is derived from the application of a probabilistic kernel.

3.3 Methods

3.3.1 Problem Statement

The training data from a facial expression dataset $D$ with $N$ samples is represented as $D = \{X_i, Y_i\}_{i=1}^{N}$, where $X_i$ is a facial expression video and $Y_i \in Y$ is its class label. Each video is a sequence of images $\{x_{it}\}_{t=1}^{T_i}$, where each $x_{it} \in \mathcal{R}^d$ is represented either in native pixel intensity space or feature space. The final goal of AFER is to predict the class label for an unseen test sequence. We first briefly describe the HMM framework being used for spatio-temporal modeling of facial expression sequences, followed by a description of probabilistic kernels. Thereafter we propose our method, referred to as Exemplar-HMMs, for the AFER task.
3.3.2 HMMs for AFER

HMM is a parametric model that is used to statistically describe a time-series under Markov assumption. In particular, the HMM models the joint probability of a time-series $X_t$ as a chain of observations $x_{it}$ and corresponding discrete (unobserved) hidden state $z_{it}$. For example, in Figure 3.1, the discrete hidden states are $N$ and $A$, and the observations are the individual snapshots shown at the bottom.

In the context of AFER, an HMM can be visualized as a spatio-temporal model describing a video as a chain of discrete sub-events. Each hidden state in an HMM model is essentially an abstraction for a sub-event and describes a distinct expression state such as neutral or a particular cluster in the space of facial actions (see Figure 3.1). Thus an HMM model consists of two parts (1) dynamic (temporal): describes the transitions between distinct facial states, and (2) appearance (spatial): describes the observation space that characterizes each sub-event. The observation space was modeled using multivariate Gaussian distribution with diagonal Covariance matrix. We found this modeling assumption to work well since the Mahanalobis distance used in Gaussian distribution serves as a good metric for the facial features used in our experiments and have also been shown to work well in previous AFER approaches [18].

HMM modeling consists of estimating parameters $\theta = \{\pi_k, A_k, \mu_k, \Sigma_k\}_{k=1}^{N_s}$, where $\pi_k = Pr(z_1 = k)$ is the initial probability of being in state $k$, $A_{k,j} = Pr(z_t = k|z_{t-1} = j)$ is the transition probability (stationary) of transitioning from state $j$ to state $k$, $\mu_k$ and $\Sigma_k$ are the mean vector and diagonal covariance matrix respectively of the Gaussian distribution corresponding to state $k$, and $N_s$ is the number of hidden states. The parameters for an HMM are estimated using maximum likelihood paradigm via the expectation maximization (EM) algorithm. For effectively learning
the model parameters in current setting we made use of the Bayesian formulation for HMM as described below.

**Bayesian HMM:** In contrast to learning an HMM from a set of examples, this work involved learning an HMM model from a single example. Since HMMs involve many parameters, learning parameters using only one example could result in severe over-fitting. Thus to obtain a robust solution, we employed a Bayesian solution to this problem by incorporating (conjugate) prior distributions for different parameters, which can effectively regularize the EM solution and avoid over-fitting [79, 38]. Normal Inverse-Wishart distribution was used as a prior for Gaussian parameters, and the Dirichlet distribution was used for both the parameters of initial and transition probabilities. The parameters are then estimated by using maximum a posteriori (MAP) estimation along with the EM algorithm (MAP-EM). Interested readers are referred to [79, 38] for more details. We found during our experiments that this formulation not only improved the reliability of model estimation but also improved results by allowing to learn Gaussian observation states with diagonal covariance structure.

### 3.3.3 Probabilistic Kernels

Parametric modeling approaches, such as Gaussian classifiers and logistic regression represent each object as a fixed-length feature vector followed by learning the parameters of the model. However, a fixed-length representation might not be the natural choice for objects such as variable length sequences, probability functions or high-dimensional objects [94, 46]. An alternate strategy is to design algorithms that are based on measuring the similarity between pairs of data-points, without ever requiring the explicit feature representation of each data point. Examples of similarity functions include distances or inner products between $X$ and $X'$, where
X and $X'$ lie in a space $S$. We denote $\mathcal{K}(X, X') \geq 0$ a kernel function. Learning approaches based on computing kernel functions between data points are examples of non-parametric methods and are generally referred to as kernel methods.

In the present work, we represent each datum as a probability distribution and thereafter use a kernel, referred to as probabilistic kernel, to compare distances between two distributions. For the task of AFER, we employ the Probability Product Kernel (PPK), proposed by Jebara et al., for computing distances between two distributions. PPK is computed between two distributions from the same family with parameters $\theta$ and $\theta'$ as:

$$K_{PPK}(\theta, \theta') = \int_{X_{1:T}} \Pr(X_{1:T}|\theta)^\rho \Pr(X_{1:T}|\theta')^\rho dX_{1:T}$$

(3.1)

$X_{1:T}$ refers to a sequence of length $T$. Here parameter $\rho$ controls the non-linearity of these kernels, while parameter $T$ determines the temporal extent to which two models shall be compared. These parameters are important for calculating the distance and are tuned using cross-validation for each dataset (see Section. 3.4). PPK can be interpreted as an inner-product between two probability distributions and is a positive definite kernel. Alternate probabilistic kernels, such as Fischer kernel or heat kernels, also exist. However in this work we concentrate on PPK owing to its computational feasibility and nonlinear flexibility [46]. Also in contrast to the popular kernel based on Kullback-Leibler Divergence, PPK is better able to provide a closed form solution for HMM models.

### 3.3.4 Exemplar-HMMs for AFER

Revisiting the AFER problem discussed in Section. 3.3.2, the proposed approach begins by learning an HMM model (with parameters $\theta_i$) for each example $X_i$. The next step computes a kernel matrix $\mathcal{K} \in \mathcal{R}^{N \times N}$ for the training data,
whose elements are computed as \( K(i, j) = K_{PPK}(\theta_i, \theta_j) \). The PPK kernel can be computed efficiently using factorization of HMM as mentioned in [46]. Non-linear similarities between the states of two HMMs while incorporating their temporal structure. The kernel matrix is then normalized using a standard normalization procedure:

\[
K_{\text{norm}}(i, j) = \frac{K(i, j)}{\sqrt{K(i, i)} \sqrt{K(j, j)}}
\]  

(3.2)

We then use the kernel matrix \( K_{\text{norm}} \) to train a support vector machine, using the libsvm library.\(^1\) The learned classifier can then be used to assign a classification score to a test sequence (\( X_t \)). This score is calculated based the similarity of its model \( \Pr(X|\theta_t) \) to the models corresponding to the support vectors identified from the training sequences. During our experiments we keep \( N_s \) (the number of states in the HMM) constant while learning HMM models for a dataset. The value of \( N_s \) was estimated using double-cross validation (see Section 3.4) for each dataset.

**Intuition behind PPK kernel**: Before proceeding to the experiment section, it is important to have an intuitive understanding of the model similarity estimated by the PPK kernel. The PPK distance consists of two aspects (1) static: which computes a probabilistic similarity between different state-wise Gaussian distributions of the two HMM models, and (2) dynamic: incorporates transition probabilities and calculates similarity for joint state transitions. The PPK kernel principally combines both these aspects into a recursive formulation that finally calculates the probability of all possible state evolution undertaken by the two HMM models together. The parameter \( T \) is important since as \( T \) increases, the distance is dominated by the terminal states of the two HMMs as governed by

\(^1\)SVM is extended to multiclass classification using the one-vs-all strategy.
the transition probabilities. For example, in the case of two expression sequences starting at neutral and ending at apex, a higher $T$ will result in having more contribution from the distances between the apex states.

### 3.4 Experiments

Performance was evaluated was using two sets of experiments. The first set consisted of datasets containing basic emotions such as anger and sadness, and were captured under laboratory settings. To further highlight the performance advantages, we evaluated our algorithm on more a challenging AFER problem, where the facial expressions were spontaneous, and captured under more naturalistic settings.

**PPK parameters:** The PPK consists of two hyper-parameters $\rho$ and $T$. In order to estimate the value of hyper-parameters without over-fitting on the test set, a double cross-validation (CV) protocol was employed. Double-cross validation simulates separate training, validation, and test sets, where the hyper-parameters are selected based on the validation set, and then evaluated on a separate test set. In this protocol, there are two nested cycles of CV. In the outer CV cycle, a set of test data is held out. Then an inner cycle of CV is conducted in order to select hyper-parameters. We then select the hyper-parameters that yield the highest average accuracy across all validation folds. System performance using those hyperparameters is then evaluated on the held out test set. The value of the two hyper-parameters was kept constant across a facial expression dataset.
3.4.1 Experiments on Posed and Spontaneous Basic Emotions

In this experiment a video was described as a time-series of facial landmark points [97, 47]. For every sequence, 49 landmarks points were obtained for each frame by using supervised gradient descent approach [145]. Displacement features for each frame were then obtained by subtracting $x$ and $y$ coordinates of the landmark points in that frame from the landmark coordinates in the first (neutral frame) in that video and concatenating these displacements into a single vector (dimensionality 98) [97]. A linear subspace was separately computed for each dataset by using these displacement features along with Principal Component Analysis (PCA) algorithm [79]. The features were then projected to a low dimensional subspace (of dimensionality $d_{\text{pca}}$) composed of principal components that preserved 99.5% variance. ²

CK+ Dataset: CK+ [72] is a standard AFER benchmark dataset consisting of 593 sequences from 123 subjects. These subjects were asked to perform a series of 23 facial displays, of which 327 sequences (118 subjects) were categorized into one of the seven basic emotion- anger, disgust, fear, happiness, sadness, surprise and contempt. The length of the sequences in this dataset varies from 10 to 60 frames and the facial expression transitions from onset (neutral phase) to apex phase. $d_{\text{pca}}$ in this case was 46 and the experiments were conducted using the leave-one-subject out protocol [72], where each fold consisted of data from just one subject. The prediction task was to predict the class of an unseen test sample and the performance metric being reported is average accuracy for the seven classes. The cross-validated values for number of HMM states was $N_s = 2$ (neutral and

²$d_{\text{pca}}$ for each dataset varied from 46 – 54, which means that due to PCA feature dimensionality was reduced by almost 50%.
apex) and kernel parameters were $\rho = 0.8$ and $T = 35$.

**Oulu-CASIA VIS Dataset:** The Oulu-CASIA VIS dataset is a subset of the Oulu-CASIA NIR-VIS dataset [154], in which all videos were captured under the visible (VIS) light condition. It consisted of 480 samples from 80 subjects displaying one of the six basic emotion- anger, disgust, fear, happiness, sadness and surprise. The subjects were asked to make a facial expression matching an expression sequence shown on a monitor. The length of the sequences varies from 9 to 72 frames and each video begins at neutral expression and ends at apex expression (similar to CK+). $d_{pca}$ in this case was 54. The experiments were conducted in a subject-independent format by using a 10 fold CV [67]. The evaluation task was to predict the class of an unseen test sample and the performance metric being reported is average accuracy for the six classes. The cross-validated values for number of HMM states was $N_s = 2$ (neutral and apex) and kernel parameters were $\rho = 0.9$ and $T = 30$.

**FEEDTUM:** The FEEDTUM facial expression dataset [130] consists of spontaneous facial expressions elicited by presenting the subjects with a set of carefully selected video stimuli. This is different from both CK+ and Oulu-CASIA VIS datasets, where the subjects were asked to perform specific facial movements. Moreover, in most cases the facial expressions sequences evolved from neutral to apex and then back to neutral, and the variability of the duration and timing of these phases was higher. $d_{pca}$ in this case was 52. The dataset contains 19 subjects (320 videos) showing six basis emotions- anger, disgust, fear, happiness, sadness and surprise. As above, each video was described as a time series of facial landmark points computed using supervised gradient descent. The experiments were conducted using leave-one-subject out protocol, where each fold consisted of data from just one subject. The evaluation task was to predict the class of an
unseen test sample and the performance metric being reported is average accuracy for the six classes. The cross-validated values for number of HMM states was $N_s = 3$ and kernel parameters were $\rho = 0.6$ and $T = 11$.

### 3.4.2 Experiments on Facial Action Time Series

To further evaluate the effectiveness of our approach, an additional set of experiments was conducted on another spontaneous facial expression dataset, the AM-FED dataset [77]. The AFER problem on this dataset was more challenging since the expression videos were not only unsegmented but also lacked prior information about the onset, duration and frequency of the target expression. Since the videos in this dataset involved large out-of-plane head movements, we were unable to use automatic facial landmark points as the frame-level feature representation as was done in the previous experiments. Instead we made use of the frame-level action unit (AU) annotations provided with this dataset for the time-series data. Action Units correspond roughly to movement of individual facial muscles. The focus of this paper is on modeling sequences, and not on feature extraction. The AU data provides a spontaneous biological time series, where the features are highly precise, thereby providing an alternate way to evaluate the sequence classification approach.

**AM-FED:** The AM-FED dataset consists of 242 sequences that were recorded on a web-cam while different subjects were viewing three Superbowl commercials. After watching the videos, the subjects provided self-report ratings for two questions: "Did you like the video?" and "Would you like to watch this video again?". The self-report responses could be positive (1), neutral (0) or negative (-1). Similar to the protocol used in [78], only videos where the labels are either positive or negative were included and the target was to predict these
binary self-report responses given a test video. The final dataset consisted of
103 and 170 sequences for "Watch/Not Watch again" and "Like/Does not like"
prediction tasks respectively. In line with the evaluation protocol described in
[77], the experiments were conducted in a (3 fold) leave-one-advertisement-out
protocol, where the videos corresponding to one advertisement were used for testing
and remaining for training. Average AUC score across all folds is reported as the
evaluation metric. Of the frame-level annotations provided with this dataset, we
discarded annotations for those facial actions that were either available for a few
examples or had a low inter-coder reliability. The final frame-level representation
comprised of annotations for AU 2, 4, 5, 14, 17, Unilateral left AU 12, Unilateral
right AU 12, Negative AU 12 and Unilateral left AU 14 along with smile and
expressability ratings (dimensionality was 11). In this dataset the value for each
AU was the percent of annotators indicating presence of the AU in the frame. We
thresholded all AU annotations < 50% to 0 and ≥ 50% to 1. The cross-validated
values for number of HMM states for task "Like/Does not like" was $N_s = 2$ and
kernel parameters were $\rho = 1.2$ and $T = 35$, while for task "Watch/Not Watch
again" was $N_s = 3$ and kernel parameters were $\rho = 0.6$ and $T = 5$.

3.4.3 Algorithms compared

We compared Exemplar-HMMs to three baseline algorithms. The first is
a standard Disc-ST pipeline, where the frame-level features across a video are
summarized using Mean-pooling and Max-pooling. The second baseline algorithm
is LBPTOP, which is amongst the most common Disc-ST approaches for AFER.
For implementing LBPTOP, spatio-temporal features were extracted from non-
overlapping blocks and concatenated into a single vector. The features from Disc-ST
pipelines and LBPTOP were passed as input to a SVM classifier with rbf kernel.
Table 3.1. Results from experimental evaluation.

(a) Results (mean % accuracy) for CK+ and OULU-CASIA VIS

<table>
<thead>
<tr>
<th>Method</th>
<th>CK+</th>
<th>OULU-CASIA VIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geom. + Mean-pooling</td>
<td>92.42 (±1.58)</td>
<td>70.83 (±2.84)</td>
</tr>
<tr>
<td>Geom. + Max-pooling</td>
<td>92.74 (±1.67)</td>
<td>69.76 (±1.73)</td>
</tr>
<tr>
<td>LBPTOP [155]</td>
<td>91.30 (±1.79)</td>
<td>72.08 (±2.22)</td>
</tr>
<tr>
<td>HMM</td>
<td>85.35 (±2.16)</td>
<td>63.54 (±3.10)</td>
</tr>
<tr>
<td>STM-ExpLect [67]</td>
<td>94.19 (N/A)</td>
<td>74.59 (N/A)</td>
</tr>
<tr>
<td>Exemplar-HMMs</td>
<td>94.60 (±1.37)</td>
<td>75.62 (±2.10)</td>
</tr>
</tbody>
</table>

(b) Results (mean area under the ROC curve) for AM-FED.

<table>
<thead>
<tr>
<th>Method</th>
<th>Like/Don’t Like</th>
<th>Watch-again/ Don’t Watch-again</th>
</tr>
</thead>
<tbody>
<tr>
<td>AU + Mean-pooling</td>
<td>.66</td>
<td>.87</td>
</tr>
<tr>
<td>AU + Max-pooling</td>
<td>.61</td>
<td>.89</td>
</tr>
<tr>
<td>HMM</td>
<td>.58</td>
<td>.84</td>
</tr>
<tr>
<td>Exemplar-HMMs</td>
<td>.84</td>
<td>.92</td>
</tr>
</tbody>
</table>

Table 3.2. Results (mean % accuracy) from FEEDTUM.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geom. + Mean-pooling</td>
<td>48.91 (±3.70)</td>
</tr>
<tr>
<td>Geom. + Max-pooling</td>
<td>53.87 (±2.59)</td>
</tr>
<tr>
<td>LBPTOP [155]</td>
<td>48.17 (±3.31)</td>
</tr>
<tr>
<td>HMM</td>
<td>48.23 (±3.88)</td>
</tr>
<tr>
<td>Exemplar-HMMs</td>
<td>54.14 (±3.72)</td>
</tr>
</tbody>
</table>

The third baseline algorithm is a generative HMM classifier (Section. 2.1). Same frame-level features were used in Disc-ST pipelines and HMM to facilitate a fair comparison. The parameters for HMM (number of states), LBPTOP (size of non-overlapping blocks) and rbf kernel (kernel width) were determined using the double CV procedure described in Section. 3.4. In addition, we have also provided the performance of a recent state-of-the-art algorithm [67] for making a relative comparison to our algorithm.
3.5 Results and Discussion

The results for CK+ and OULU-CASIA VIS datasets are shown in Table. 3.0a, AM-FED dataset in Table. 3.0b and FEEDTUM dataset in Table. 3.2. The results are divided into two blocks (using double lines), into Disc-ST methods and dynamic methods, where the term 'dynamic' refers to approaches explicitly incorporating temporal information inside the algorithm. Our contention that Exemplar-HMMs is able to overcome the limitations of HMMs by combining its modeling advantages with a discriminative classification paradigm is supported by the comparison to the standard HMM. The proposed approach outperforms the standard HMM on all the datasets by an appreciable margin. For example, the absolute performance improvement is approximately 10% accuracy for CK+ and OULU-CASIA, and 5% for FEEDTUM.

Our approach outperforms the baseline Disc-ST method with Max-pooling and Mean-pooling on all datasets. The performance advantage for Exemplar-HMMs is greatest for the AM-FED spontaneous expression dataset. For instance, on the Like/Don’t Like task, the AUC score for Mean-pooling and Max-pooling are .66 and .61 respectively, compared to .84 for Exemplar-HMMs, and for the Watch-again/Don’t Watch-again task, the AUC score for our approach is .92 compared to .87 and .89 for Mean-pooling and Max-pooling respectively. This increase in performance can be attributed to the ability of Exemplar-HMMs to address the key modeling issues in Disc-ST methods (1) discriminative information, and (2) temporal alignment, as discussed in Section. 3.1. We further observed that our approach outperformed LBPTOP method on all three emotion datasets. This is particularly interesting since LBPTOP utilizes image textures that may be capable of capturing more information than the geometric features [153] employed...
in our method. Moreover the (absolute) performance increase is greater for the spontaneous expression dataset FEEDTUM (5.9% hike) than the posed datasets CK+ (3.3%) and OULU-CASIA VIS (3.5%). One possible explanation is that LBPTOP doesn’t take into account the temporal structure inherent in a time-series, and this information might be important for expression recognition on spontaneous expressions such as those in FEEDTUM.

We also compared the performance of Exemplar-HMMs with a recent (2014) state-of-the-art algorithm (STM-ExpLet [67]) that explicitly adds temporal information inside the algorithm. Exemplar-HMMs achieved 75.62% accuracy on OULU-CASIA VIS dataset, which was similar to the accuracy of STM-ExpLet (74.59%). For CK+ our results were calculated using the standard leave-one-subject-out protocol as mentioned in [72], while 10 fold CV was used to calculate results in STM-ExpLet paper. Thus for making a fair comparison we ran a 10 fold CV (randomized over 10 runs) for CK+ and obtained an accuracy of 93.89% compared to 94.16% in STM-ExpLet. Although the results from our algorithm seems comparable to the state-of-the-art, it is important to note that our algorithm was based on geometric features while STM-ExpLet used 3D texture based appearance features, which are capable of capturing more information and have also previously been shown to outperform geometric features [153]. Thus the observation that the present classification paradigm is able to match the state-of-the-art results on current AFER problems despite using simpler features highlights both its performance advantages and the promising nature of the present research direction for tackling AFER. In order to extend current approach to texture based features, we have discussed our ongoing work to exploit high-dimensional features in the current pipeline in the next section.
3.6 Conclusion and Future Work

This paper builds upon the idea that facial expressions have a specific temporal structure that can be explicitly modeled using latent variable sequential models. Focusing on Hidden Markov Models, we argue that they provide certain modeling benefits over temporally holistic feature based approaches for facial expression recognition. However owing to their generative nature, HMMs typically have a lower classification performance than discriminative classifiers such as Support Vector Machines (SVMs). This paper explored an approach for combining the modeling strength of HMMs with the discriminative power of SVMs via probabilistic kernels for the task of facial expression recognition. This combination was achieved by modeling each example with an HMM, followed by computing a kernel matrix, via Probabilistic Product Kernels, that comprised the input to an SVM. By achieving state-of-the-art results on both posed and spontaneous datasets, this approach highlighted its performance advantage for video-based facial expression recognition compared to traditional HMMs, and compared to discriminative approaches based on temporally holistic features.

This preliminary work showed both the modeling and performance advantages of an approach relying on model-based similarity for AFER. Building on this line of research, our current focus is on extending both HMMs and probabilistic kernels to handle high-dimensional spatio-temporal features that can further enhance its modeling advantages. This is being accomplished by the application of the kernel trick for embedding data-points in an implicit projection space, and then extending HMMs and probabilistic measures to work with kernels. We shall also explore the possibility of using other probabilistic kernels (Fisher, KL-divergence etc.) as part of our future work.
3.7 Acknowledgment

Support for this work was provided by NIH grant NIH R01NR013500. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding agency. We would also like to thank Mohsen Malmir and Ritwik Giri for helpful discussions.

Chapter 3, in full, is a reprint of the material as it appears in Computer Vision and Pattern Recognition Workshops 2015. Sikka, Karan; Dhall, Abhinav; Bartlett, Marian., copyright IEEE, 2014. The dissertation author was the primary investigator and author of this paper.
Chapter 4

Joint Clustering and Classification for Multiple Instance Learning

4.1 Introduction

Many early approaches to visual classification were based on extracting a global descriptor, followed by learning a classification function using training labels [83, 85]. Since these methods describe an image/video as a whole e.g. GIST [83] and color histograms [85], they were often referred to as global representations. These approaches worked well for problems where the test object (or action) was visually uniform, but performed poorly when there were large variations in the object class due to factors such as occlusion, viewpoint changes or visual sub-categories [52, 22, 40]. A possible solution is to use local approaches where each image or video is represented by a set of localized visual descriptors [22, 16]. For example, as shown in Figure 4.1, a beach scene could be described by instances corresponding to a set of underlying regions. However, this setting differs from the standard supervised learning case where a unique class label is provided with each training instance. The only information available about the beach image is that at least one of the regions inside the beach scene corresponds to the target class. Since there is
**Figure 4.1.** We illustrate the inherent idea in Embedding Space based MIL approaches. A beach scene, segmented into regions, is represented as a bag of instances, where each instance is the visual descriptor of the corresponding region. A set of concepts is then used to calculate a similarity between each instance and the concept, referred to as concept-wise instance similarity (CIS). The max MIL assumption is used to embed each bag into the concept space using the CIS. Classification can then be performed in the embedding space using standard classifiers (best viewed in color).

Incomplete information regarding the class labels, this setting is often referred to as Weakly Supervised setting.

This setting is common in computer vision since complete human annotation is both costly and intensive, while labeling an image or video with a single global label is often more feasible. The data can be structured as a set of bags, each containing many instances, along with labels indicating the presence/absence of the object of interest in each bag. The learning problem in this case is referred to as Multiple Instance Learning (MIL) [3, 8]. Most of the previous MIL algorithms extended standard supervised learning algorithms to the MIL setting by assuming that the posterior probability of a bag containing a positive target class is a
maximum over the probability of each of its instances. Thus if one of the instances has the target class with high probability, then the probability of the bag containing that target is also high irrespective of other instances. We shall refer to this as the $max$ MIL assumption. This idea has been used to extend several supervised learning algorithms such as boosting [150] and logistic regression [33], to the MIL setting. Adapting the taxonomy proposed in [3], we shall refer to these algorithms as **Instance Space** (IS) based algorithms since they first define a classification function in the space of instances and extend it to the entire bag using the $max$ assumption. IS-based MIL methods work well when the positive class can be described by a single target concept. However, the assumption of a single target concept might be too restrictive in several vision problems, e.g. a scene may comprise several local regions or an action may include multiple components. In such cases a bag is composed of heterogeneous concepts, which may contribute differently towards its classification. Consider the example of classifying a scene into beach or desert. Examples from both classes contain regions corresponding to sand and cloud, however in the case of the beach scene both water and sand must occur together. Moreover, these regions correspond to intermediate concepts that differ from the target label “beach”.

To tackle the above issues, we focus on **Embedding Space** (ES) based MIL approaches [3] that embed each bag into a $K$-dimensional vector space. The procedure is illustrated in Figure 4.1, where a beach scene is represented as a bag of image regions. In this example each concept has an associated semantic meaning such as water, clouds or background. First, a similarity score between each instance and a concept is computed, which produces a **Concept-wise Instance Similarity** (CIS) for every instance. In the next step, the similarity between the bag and a concept is computed as the max CIS score, using the $max$ MIL assumption similar
to IS-based methods. The likelihood of each concept forms each dimension of an intermediate vector space, referred to as concept space. In other words, the set of concept likelihoods for the bag forms the embedding. After embedding each bag into the concept space, standard classifiers can be used to classify the overall target from the embedded representation. The concepts can take a number of forms. They could be cluster centers or dictionary atoms discovered by unsupervised methods such as k-means [22, 16, 36], or they could be concept prototypes learned by maximizing Diverse Density measure on MIL data [75, 151, 17]. The concepts induce a clustering of data instances into multiple categories as shown in Figure 4.1. This is related to dictionary learning based recognition approaches, where a dictionary is obtained using unsupervised algorithms. This dictionary is then used to encode features with the aim of improving recognition rate in the new space [22, 36].

In most previous ES-based approaches, the concepts and the classifier were obtained independently. However, isolated learning of the classifier and the set of concepts (or dictionary) may not be optimal for the final classification task since the discovered concepts, which were optimized for a different objective (such as minimization of inter-cluster distance in k-means), may induce a concept embedding that is not suited to classification. [36, 74]. We propose to tackle this problem by introducing a novel ES-based MIL method that jointly learns the set of concepts and the classifier in a MIL setting. We refer to our algorithm as Joint Clustering and Classification for Multiple Instance Learning (JC²MIL). Our work makes the following contributions:

1. Proposes a framework to estimate concepts and classifier parameters by jointly optimizing a classification loss on the MIL data.

2. Shows that the current approach is able to yield state-of-the-art results on
several MIL datasets by discovering discriminative concepts. The number of concepts are much smaller compared to the overcomplete set used in previous ES-based MIL algorithms [16].

To facilitate a fair comparison with the previous state-of-the-art ES-based MIL approaches [17, 104, 16], we use a RBF kernel based mapping function. It is also interesting to note that JC$^2$MIL follows the line of recent work on task-driven dictionary learning [74].

4.2 Related Work

The idea behind IS-based methods is to construct an instance classifier by modifying a supervised learning algorithm using the MIL assumptions [3]. A number of algorithms have been proposed, such as MILBOOST [151], MI-logistic regression [33], MI-SVM [4], MI-Forests [58]), and used to tackle visual classification problems [108, 147, 136]. IS-based formulation has also been used to propose a mixture of linear [134, 136] or non-linear classifiers [147, 75] to solve the MIL problem. Although these algorithms were capable of detecting multiple concepts, they did not assume that different concepts can contribute differently towards the label of the bag.

On the other hand, the ES-based MIL algorithms are able to incorporate the above assumption by first embedding each bag into a concept space, followed by learning a standard classifier in this space [3]. A classic example of ES-based algorithms is the Bag of Words (BoW) model [22] that maps an image/video into a histogram using an unsupervised dictionary. Popular ES-based MIL approaches are based on the idea of extracting prototype(s) by maximizing the diverse-density (DD) function [75] on MIL data. The motivation is that a point with high DD
is close to at least one instance inside a positive bag and far away from every instance in the negative bags. In this way, each prototype can also be identified as a positive concept. Chen et al. proposed DD-SVM [17] to discover several concepts through DD function and then used the corresponding concept space with SVM. An improvement was later proposed over DD-SVM called MILES [16], where the set of concepts included all the instances in the dataset, and the relevant instances were selected using sparse-SVM. A recent algorithm, called Dictionary based Multiple Instance Learning (DMIL) [104], learned a sparse reconstruction based dictionary by maximizing the DD function, and then used the sparse codes for each instance to embed a bag. Although this algorithm achieved excellent results, DMIL did not have an explicit notion of clustering owing to the use of a single mapping function for constructing the concept space.

It is also worth mentioning about Bag Space (BS) based MIL algorithms [3] that calculate similarities between bags through kernels such as MI-kernel [37], mi-Graph [157], followed by the application of a Kernel-based learner such as SVM. These methods usually assume non i.i.d relationships between instances inside a bag and require the definition of a distance function. Although these methods have good performance, they are generally not able to do any instance classification. This paper focuses only on IS-based and ES-based MIL algorithms that allow both instance and bag classification.

Our work is also related to the recent work on task-driven dictionary learning [74, 36], where dictionary learning is coupled with the final task e.g. classification. In such a setting it is possible to achieve a good recognition rate without the use of an overcomplete dictionary (as in BoW [22] or MILES [16]) since the learned dictionary is already tuned to the final task. Similar extensions have been proposed in sparse-coding [74], BoW [63]. This work extends the previous ES-based MIL
algorithms by discovering discriminative concepts that are learned simultaneously with the bag-level classifier.

4.3 Joint Clustering and Classification for Multiple Instance Learning (JC$^2$ML)

4.3.1 Model

A MIL dataset is denoted as $B = \{X_i, y_i\}_{i=1}^N$, where $X_i$ is the $i^{th}$ bag, $y_i \in \{0, 1\}$ is its binary label$^1$ and $N$ is the cardinality of the dataset. Each bag $X_i$ consists of $N_i$ instances, denoted as $X_i = \{x_{ij}\}_{j=1}^{N_i}$, where $x_{ij} \in \mathbb{R}^d$ is a visual descriptor representing an instance. The target of this work is to jointly learn (1) a set of concepts, that are used to embed each bag into a concept space, and (2) a classifier that combines the embedding to produce a classification score for each bag. Our algorithm discovers discriminative concepts by learning them simultaneously with the classifier, that requires minimization of the classification loss on the training data.

The set of concepts is denoted as $C = \{\mu_k\}_{k=1}^K$, consisting of $K$ elements $\mu_k$. The similarity between the $k^{th}$ concept and instance $x_{ij}$ is denoted as $p_{ijk}$. We purposely selected the RBF kernel, written as $p_{ijk} = \exp(-\frac{\gamma}{2} ||x_{ij} - \mu_k||^2)$, since it was used in several previous ES-based methods- MILES [16], DD-SVM [17] and DMIL [104]). The CIS for the $k^{th}$ concept are then used to derive the $k^{th}$ embedding dimension, denoted as $\phi_{ik}$, for the $i^{th}$ bag using the max MIL assumption:

$$\phi_{ik} = \max_j p_{ijk}$$

$^1$Although the algorithm is formulated for binary classification problems, it can be extended to multiclass problems by learning one-vs-all binary classifiers.
The underlying idea is similar to the Instance based MIL algorithms where the probability of a bag is defined as the maximum over the probability of each of its instances. However, instead of using a single classifier or a concept, the probability is calculated with respect to multiple concepts. The vector containing the similarity score from all the $K$ concepts for the $i^{th}$ bag is denoted as $\phi_i = \{\phi_{ik}\}_{k=1}^{K} \in \mathcal{R}^K$, which also forms the embedding of the bag in the concept space. The classification score is then obtained by a linear classifier with parameter $w$. In this work we opted for a logistic regression classifier since it is able to provide good generalization by minimizing a differentiable loss function. The classification score for the $i^{th}$ bag is converted to posterior probability, denoted as $p_i$, by using the sigmoid function as $p_i = \sigma(w^T \phi_i)$ [79].

4.3.2 Joint Optimization

We pose the problem as joint minimization of the classification loss with respect to the concepts and the classifier parameters. The classification loss includes the mean negative log-likelihood and a regularization term [79], and written as:

$$
\mathcal{L}(B) = -\frac{1}{N} \sum_i (y_i \log p_i + (1 - y_i) \log(1 - p_i)) + \frac{\lambda}{2} w^T w \quad (4.2)
$$

$$
\{\mathcal{C}^*, w^*\} = \arg\min_{\mathcal{C},w} \mathcal{L}(B) \quad (4.3)
$$

The above optimization problem is not jointly convex in both $\mathcal{C}$ and $w$, however it is convex in either while keeping the other variable fixed. Thus the minimization is performed via coordinate descent approach, where $\mathcal{L}$ is minimized alternatively with respect to both the variables. Since the original problem is non-convex, the final solution is dependent on the initialization. During our experiments we found that the initialization of the concepts with k-means (cluster centers) [25]
was able to find relevant clusters and yield good results. The alternate minimization was performed using the BFGS algorithm, which is faster compared to gradient descent [99].

The gradient of $L$ can be easily computed with respect to $w$ (logistic regression [79]). The gradient of $L$ with respect to each individual concept $\mu_k$ is expanded using the chain-rule of differentiation as:

$$
\frac{\partial L}{\partial \mu_k} = \frac{1}{N} \sum_i (p_i - y_i) \frac{\partial \phi_i}{\partial \mu_k} = \frac{1}{N} \sum_i (p_i - y_i) w_k \frac{\partial \phi_{ik}}{\partial \mu_k}
$$

(4.4)

The second expression follows since only the $k^{th}$ dimension of $\phi_i$ is dependent on $\mu_k$. $\phi_{ik}$ was computed in Equation 4.1 by taking a maximum over the per instance similarity scores ($p_{ijk}$) for the $k^{th}$ concept. However, the non-differentiability of the maximum function poses a problem in estimating the above derivative. This problem was solved by using a soft-max approximation [8]. We used the Generalized Mean (GM) approximation instead of the frequently used NOR approximation since previously GM has been shown to provide better performance compared to NOR [108, 8]. As per the GM function, the embedding $\phi_{ik}$ is defined using $p_{ijk}$ as $\phi_{ik} = \left( \frac{1}{N_i} \sum_j p_{ijk}^r \right)^{\frac{1}{r}}$, where $r$ is a parameter controlling the degree of approximation. The inner-derivative in Equation. 4.4 is then written as:

$$
\frac{\partial \phi_{ik}}{\partial \mu_k} = \frac{\phi_{ik}}{\sum_j p_{ijk}^r} \sum_j p_{ijk}^{r-1} \frac{\partial p_{ijk}}{\partial \mu_k}
$$

(4.5)

The above derivative can be easily computed to write the derivative of the
loss function with respect to $k^{th}$ concept ($\mu_k$) as:

$$\frac{\partial L}{\partial \mu_k} = -\gamma w_k \frac{1}{N} \sum_i (p_i - t_i) \phi_{ik} \sum_j p_{ijk} (\mu_k - x_{ij})$$  \hspace{1cm} (4.6)$$

Computing the above gradient is efficient since it can be written in terms of matrix products. For instance classification, we use the same procedure as introduced in MILES [16].

4.4 Experiments

To establish the empirical superiority of our proposed method over previous MIL algorithms, we tested it on five MIL datasets. The number of concepts and the RBF kernel parameter $\gamma$ were tuned by using five fold cross-validation on the training set. The regularization and Generalized Mean parameter were set to a constant value ($\lambda = 10^{-5}$, $r = 15$). We found the algorithm to converge in about 20 iterations of coordinate descent. As a standard pre-processing step, the features were processed to have a zero-mean and unit variance. The concepts were initialized with 10 repetitions (for reproducing results) of k-means [25].

The first set of experiments were performed on the three image annotation datasets- Tiger, Fox and Elephant [4]. In these datasets, an image consists of a set of segments (or blobs), each characterized by color, texture and shape descriptors. For an image lying in the positive class at least one of the blobs belongs to the target category. We used the same experimental protocol as described in [58], where average classification accuracy was reported using 5 randomly selected 10 fold cross-validation sets. The comparative performance for these 3 datasets is shown in Table 4.1.

\footnotetext{2}{The random splits were downloaded from code provided by authors in [58].}
The second set of experiments were conducted on the Corel-2000 dataset that consists of 20 object categories with 100 images per category. In order to make a fair comparison, we used the same version of dataset as in [16], where each image was segmented into a number of regions, each of which were described by a 9 dimensional low-level feature vector consisting of color moments, gradient etc. Similar to [16], 5 random splits were generated by dividing the images into two equal parts, where (for each split) one part was used for testing and other for training. Since the present formulation only addresses binary classification problems, we conducted multiclass classification by training 20 per class one-vs-all binary classifiers. During test time, an image was classified as belonging to the category with the highest classification score. The results of different algorithms are shown in Table 4.2.

We also tested our algorithm on a public breast cancer dataset (UCSB Breast Cancer) with image samples taken from 32 benign and 26 malignant breast cancer patients [50]. Each image was divided into an equal-sized $7 \times 7$ patch and visual features such as SIFT, color histograms were extracted from each patch to form a 708 dimensional vector [50]. The dimensionality was reduced to 100 by application of Principal Component Analysis [25]. The task is to classify malignant and benign cancer images and the metric being reported is the mean area under the ROC curve (AROC) for a 10-fold stratified cross-validation (Table 4.3).

### 4.5 Results and Discussion

Table 4.1 shows that our algorithm outperforms the performance of the best Instance Space based MIL algorithms on the MIL benchmark datasets by an absolute margin of 4% on Elephant, 9% on Fox and 4% on Tiger. Although DMIL seems to perform better than JC$^2$MIL on the Tiger and Elephant datasets by a
Table 4.1. Comparison of our algorithm (% accuracy) with different Instance Space based (IS) or Embedding Space (ES) based algorithms on three image annotation datasets [4]. The second column shows the type of MIL algorithm (see Section 4.1). The results were similar to DMIL in the case of the Elephant and the Tiger datasets since the standard deviation in both of these cases was around 1%.

<table>
<thead>
<tr>
<th>Method</th>
<th>type</th>
<th>Elephant</th>
<th>Fox</th>
<th>Tiger</th>
</tr>
</thead>
<tbody>
<tr>
<td>mi-SVM [4]</td>
<td>IS</td>
<td>82</td>
<td>58</td>
<td>79</td>
</tr>
<tr>
<td>MI-SVM [4]</td>
<td>IS</td>
<td>81</td>
<td>59</td>
<td>84</td>
</tr>
<tr>
<td>MILBoost-NOR [150]</td>
<td>IS</td>
<td>73</td>
<td>58</td>
<td>56</td>
</tr>
<tr>
<td>EM-DD [151]</td>
<td>IS</td>
<td>78</td>
<td>56</td>
<td>72</td>
</tr>
<tr>
<td>MIForests [58]</td>
<td>IS</td>
<td>82</td>
<td>64</td>
<td>82</td>
</tr>
<tr>
<td>MILES [16]</td>
<td>ES</td>
<td>81</td>
<td>62</td>
<td>80</td>
</tr>
<tr>
<td>DMIL [104]</td>
<td>ES</td>
<td>87</td>
<td>68</td>
<td>89</td>
</tr>
<tr>
<td>JC²MIL (Ours)</td>
<td>ES</td>
<td>86*</td>
<td>73</td>
<td>88*</td>
</tr>
</tbody>
</table>

The margin of ~ 1%, the results are statistically similar since both accuracies have a standard deviation of ~ 1%. The performance improvement on the Fox dataset is significant relative to DMIL, with a margin of 5%. This hike in performance could be explained by the presence of multiple target concepts in the Fox dataset, which is successfully captured by our algorithm. This argument is also verified by the classification accuracy on the Corel-2000 dataset (Table 4.2), since this dataset is known to have scene images with multiple target concepts, an example is shown in Figure 4.1. In this multiclass classification problem, our algorithm shows a clear performance improvement of 3% relative to the previous state-of-the-art results achieved by DMIL. This table also shows that the ES-based approaches perform much better compared to the IS-based MIL methods on the Corel-2000 dataset, e.g. MI-SVM achieves a performance of 54.6% compared to 73.2% by JC²MIL and 68.7% by MILES. This observation highlights our contention that the IS-based approaches are unable to effectively tackle problems with multiple concepts that may contribute unequally towards the classification of the bag.

Our algorithm also achieves the state-of-the-art AROC score of 0.95 on the
Table 4.2. Evaluation (multiclass % accuracy) of Instance Space and Embedding Space based MIL algorithms on the Corel-2000 dataset [17] along with 95% confidence interval.

<table>
<thead>
<tr>
<th>Method</th>
<th>Corel-2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>MI-SVM [4]</td>
<td>54.6: [53.1 63.1]</td>
</tr>
<tr>
<td>MILES [16]</td>
<td>68.7: [67.3 70.1]</td>
</tr>
<tr>
<td>DD-SVM [17]</td>
<td>67.5: [66.1 68.9]</td>
</tr>
<tr>
<td>k-means-SVM [22]</td>
<td>52.3: [51.6 52.9]</td>
</tr>
<tr>
<td>DMIL [104]</td>
<td>70.2: [68.3 72.1]</td>
</tr>
<tr>
<td>JC2MIL (Ours)</td>
<td><strong>73.2</strong>: [71.2 74.8]</td>
</tr>
</tbody>
</table>

Table 4.3. Evaluation (mean Area Under the ROC curve) on the UCSB Breast Cancer dataset [50].

<table>
<thead>
<tr>
<th>Method</th>
<th>UCSB Breast Cancer</th>
</tr>
</thead>
<tbody>
<tr>
<td>MILBoost-NOR [150]</td>
<td>0.83</td>
</tr>
<tr>
<td>MI-SVM [4]</td>
<td>0.90</td>
</tr>
<tr>
<td>MILES [16]</td>
<td>0.74</td>
</tr>
<tr>
<td>JC2MIL (Ours)</td>
<td><strong>0.95</strong></td>
</tr>
</tbody>
</table>

UCSB Breast Cancer dataset. The results on this dataset are interesting since in all of the previous datasets (except Tiger) the performance of MILES was greater than or equal to MI-SVM and MILBOOST. This could be the case since MILES might be overfitting as a result of using a large number of concepts and unable to select relevant concepts using sparse-SVM. In this scenario our algorithm not only outperforms MILES but also perform better than the state-of-the-art IS-based algorithms.

4.5.1 Advantages of Discovering Discriminative Concepts

As discussed in Section 4.1, the primary advantage of discovering discriminative concepts through joint training is that the discovered concepts are already tuned to the final task leading to a performance improvement. This allows to achieve a good recognition rate by using a small number of concepts compared
to the methods that separately learn the classifier and the concepts [22, 16, 104]. The overcompletenes is required in unsupervised dictionary based methods since it relaxes the classification problem by inducing a high dimensional embedding space where the data can be easily separated. On the other hand by incorporating label information during concept discovery, our algorithm is able to induce a low dimensional embedding space that is highly discriminative. To quantitatively highlight this point, we compared the performance of JC²MIL with a variant employing the same set of concepts, discovered via k-means, that were used to initialize our algorithm. A fair comparison was made by using the same algorithmic design (classifier and similarity kernel) as JC²MIL. We shall refer to this variant as “k-means + LR (BoW)” since in principle it is similar to the Bag of Words model.

Figure 4.2 shows the performance of both the algorithms as a function of the dictionary size for the Corel-2000 dataset. This result supports our contention that the task-specific concepts discovered by our algorithm are able to yield a significant performance improvement over k-means + LR (BoW). Moreover, our algorithm obtains the state-of-the-art results (73.2%) with a small number of concepts (equal to 20) and the performance saturates for higher number of concepts. On the contrary the performance of the BoW variant increases with the size of concepts and reaches to a maximum of 70.1% (at a concept size of 3000), which is still 3% points below the performance of JC²MIL. We have only shown results till a concept size of 50 for clarity of exposition.

4.6 Conclusion and Future Directions

This paper proposed a novel Embedding Space (ES) based Multiple Instance Learning (MIL) approach for visual classification problems. Unlike Instance-space (IS) based MIL approaches such as MILBOOST [150], the proposed method models
Figure 4.2. This graph highlights the advantage of learning discriminative concepts simultaneously with the classifier (see Section 4.5.1). The plot shows the performance of our algorithm and its variation using unsupervised concepts (k-means + LR (BoW)), on Corel-2000 dataset, as a function of the number of concepts. The performance of JC²MIL reaches a maximum of 73.2% at a concept size of 20 and saturates for a higher number of concepts. On the other hand, the performance of the BoW variant reaches to a maximum of 70.1% at a concept size of 3000 and saturates thereafter.

the presence of multiple intermediate concepts that may contribute unequally towards predicting an object or an action. Arguing the advantages of learning concepts in a task-driven fashion, we proposed a novel approach for jointly learning the set of concepts and classifier parameters in a MIL setting. The proposed solution addresses an inherent issue in previous ES based methods where the concepts and the classifier were tuned independently, leading to concepts that may not be optimal for classification. We refer to our algorithm as Joint Clustering and Classification for Multiple Instance Learning (JC²MIL) since the set of discovered concepts can be related to semantic clusters in the instance space. The performance advantages
of JC²MIL were shown by reporting state-of-the-art results on several challenging MIL datasets. We further showed the advantages of discovering discriminating concepts in JC²MIL compared to algorithms using unsupervised concepts.

We also observed that our algorithm outperformed unsupervised dictionary based ES methods by discovering a (relatively) small number of concepts. This allows our algorithm to be easily kernelized, making it possible to successfully cluster and classify fine-grained categories with high accuracy [36]. A possible research avenue that emerges with this work is towards using a more generic model for learning concepts (such as sparse coding based dictionaries [36, 117, 74]) in the proposed joint framework.

4.7 Acknowledgment

Support for this work was provided by NIH grant NIH R01NR013500. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding agency. We would like to thank Dr. Zhuowen Tu, Mohsen Malmir and Dr. Abhinav Dhall for helpful discussions. The authors in particular thank David Mateos-Núñez for helping with proofreading.

Chapter 4, in full, is a reprint of the material as it appears in British Machine and Vision Conference 2015. Sikka, Karan; Giri, Ritwik; Bartlett, Marian., BMVA Press, 2015. The dissertation author was the primary investigator and author of this paper.
Chapter 5

LOMo: Latent Ordinal Models for Facial Analysis in Videos

5.1 Introduction

Facial analysis is an important area of computer vision. The representative problems include face (identity) recognition [156], identity based face pair matching [42], age estimation [1], kinship verification [69], emotion prediction [29], [49], among others. Facial analysis finds important and relevant real world applications such as human computer interaction, personal robotics, and patient care in hospitals [108, 71, 128, 23]. While we work with videos of faces, i.e. we assume that face detection has been done reliably, we note that the problem is pretty challenging due to variations in human faces, articulations, lighting conditions, poses, video artifacts such as blur etc. Moreover, we work in a weakly supervised setting, where only video level annotations are available and there are no annotations for individual video frames.

In weakly supervised setting, Multiple Instance Learning (MIL) [4] methods are one of the popular approaches and have been applied to the task of facial video analysis [108, 96, 143] with video level, and not frame level, annotations. However, the main drawbacks of most of such approaches are that (i) they use the maximum
The model scores sub-events (color coded) in a particular order.

The final score is the sum of (i) the average sub-event scores, and (ii) the cost of the sequence in which they appear.

Different ordering of sub-events carry different costs.

**Figure 5.1.** Illustration of the proposed approach.

scoring vector to make the prediction [4], and (ii) the temporal/ordinal information is always lost completely. While, in the recent work by Li and Vasconcelos [61], MIL framework has been extended to consider multiple top scoring vectors, the temporal order is still not incorporated. In the present paper we propose a novel method that (i) works with weakly supervised data, (ii) mines out the prototypical and discriminative set of vectors required for the task, and (iii) learns constraints on the temporal order of such vectors. We show how modelling multiple vectors instead of the maximum one, while simultaneously considering their ordering, leads to improvements in performance.

The proposed model belongs to the family of models with structured latent variables e.g. Deformable Part Models (DPM) [30] and Hidden Conditional Random Fields (HCRF) [139]. In DPM, Felzenszwalb et al. [30] constrain the location of the parts (latent variables) to be around fixed anchor points with penalty for deviation...
while Wang and Mori [139] impose a tree structure on the human parts (latent variables) in their HCRF based formulation. In contrast, we are not interested in constraining our latent variables based on fixed anchors [30] or distance (or correlation) among themselves [139, 95], but are only interested in modeling the order in which they appear. Thus, the model is stronger than models without any structure while being weaker that models with more strict structure [30, 139].

The current model is also reminiscent of Actom Sequence Model (ASM) of Gaidon et al. [34], where a temporally ordered sequence of sub-events are used to perform action recognition in videos. However, ASM requires annotation of such sub-events in the videos; the proposed model aims to find such sub-events automatically. While ASM places absolute temporal localization constraints on the sub-events, the proposed model only cares about the order in which such sub-events occur. One advantage of doing so is the flexibility of sharing appearances for two sub-events, especially when they are automatically mined. As an example, the facial expression may start, as well as end, with a neutral face. In such case, if the sub-event (neutral face) is tied to a temporal location we will need two redundant (in appearance) sub-events i.e. one at the beginning and one at the end. While, here such sub-events will merge to a single appearance model, with the symmetry encoded with similar cost for the two ordering of such sub-event, keeping the rest same.

In summary, we make the following contributions. (i) We propose a novel (loosely) structured latent variable model, which we call Latent Ordinal Model (LOMo). It mines prototypical sub-events and learns a prior, in the form of a cost function, on the ordering of such sub-events automatically with weakly supervised data. (ii) We propose a max-margin hinge loss minimization objective, to learn the model and design an efficient stochastic gradient descent based learning algorithm.
(iii) We validate the model on four challenging datasets of expression recognition [70, 154], clinical pain prediction [71] and intent prediction (in dyadic conversations) [103]. We show that the method consistently outperforms temporal pooling and MIL based competitive baselines. In combination with complementary features, we report state-of-the-art results on these datasets with the proposed model.

5.2 Related works

Early approaches for facial expression recognition used apex (maximum expression) frames [113, 82, 23] or pre-segmented clips, and thus were strongly supervised. Also, they were often evaluated on posed video datasets [70].

To encode the faces into numerical vectors, many successful features were proposed e.g. Gabor [66] and Local Binary Patterns (LBP) [82], fiducial points based descriptors [153]. They handled videos by either aggregating features over all frames, using average or max-pooling [56, 107], or extending features to be spatio-temporal e.g. 3D Gabor [144] and LBPTOP [155]. Facial Action Units, represent movement of facial muscle(s) [23], were automatically detected and used as high level features for video prediction [23, 65].

Noting that temporal dynamics are important for expressions [23], the recent focus has been more on algorithms capturing dynamics e.g. Hidden Markov Model (HMM) [18, 64] and Hidden Conditional Random Fields (HCRF) [15, 76, 93] have been used for predicting expressions. Chang et al. [15] proposed a HCRF based model that included a partially observed hidden state at the apex frame, to learn a more interpretable model where hidden states had specific meaning. The models based on HCRF are also similar to latent structural SVMs [139, 114], where the structure is defined as a linear chain over the frames. Other discriminative methods were proposed based on Dynamic Bayesian Networks [152] or hybrids of HMM and
SVM [125]. Lorincz et al. [68] explored time-series kernels e.g. based on Dynamic Time Warping (DTW) for comparing expressions. Another model used probabilistic kernels for classifying exemplar HMM models [107].

Nguyen et al. [80] proposed a latent SVM based algorithm for classifying and localizing events in a time-series. They later proposed a fully supervised structured SVM for predicting Action Unit segments in video sequences [114]. Our algorithm differs from [80], while they use simple MIL, we detect multiple prototypical segments and further learn their temporal ordering. MIL based algorithm has also been used for predicting pain [108]. In recent works, MIL has been used with HMM [143] and also to learn embedding for multiple concepts [96] for predicting facial expressions. Rudovic et al. [95] proposed a CRF based model that accounted for ordinal relationships between expression intensities. Our work differs from this work in handling weakly labeled data and modeling the ordinal sequence between sub-events (see §5.1).

We also note the excellent performances reached by recurrent neural networks on video classification tasks e.g. Karpathy et al. [51] and the reference within. While such, neural networks based, methods lead to impressive results, they require a large amount of data to train. In the tasks we are interested in, collecting large amounts of data is costly and has practical and ethical challenges e.g. clinical pain prediction [71, 141]. While networks trained on large datasets for identity verification have been recently made public [84], we found empirically that they do not generalize effectively to the tasks we are interested in (§5.4).
5.3 Approach

We now describe our proposed Latent Ordinal Model (LOMo) in detail. We denote the video as a sequence of $N$ frames\(^1\) represented as a matrix $X = \begin{bmatrix} x_1, x_2, \ldots, x_N \end{bmatrix}$ with $x_f \in \mathbb{R}^d$ being the feature vector for frame $f$. We work in a weakly supervised binary classification setting, where we are given a training set

$$X = \{(X, y)\} \subset \mathbb{R}^{d \times N} \times \{-1, +1\}$$

containing videos annotated with the presence ($y = +1$) or absence ($y = -1$) of a class in $X$, without any annotations for specific columns of $X$ i.e. $x_f \forall f \in [1, N]$. While we present our model for the case of face videos annotated with absence or presence of an expression, we note that it is a general multi-dimensional vector sequence classification model.

The model is a collection of discriminative templates (cf. SVM hyperplane parameters) and a cost function associated with the sequence of templates. The templates capture the appearances of different sub-events e.g. neutral, onset or offset phase of an expression [114], while the cost function captures the likelihood of the occurrence of the sub-events in different temporal orders. The parts and the cost function are all automatically and jointly learned, from the training data. Hence, the sub-events are not constrained to be either similar or distinct and are not fixed to represent certain expected states. They are mined from the data and could potentially be a combination of the sub-events generally used to describe expressions.

\(^1\)We assume, for brevity, all videos have the same number of frames, extension to different number of frames is immediate
Figure 5.2. SGD based learning for LOMo

Formally, the model is given by

\[ \Theta = \left( \{ w_i \}_{i=1}^M, \{ c_j \}_{j=1}^{M!} \right), w_i \in \mathbb{R}^d, c_j \in \mathbb{R} \]  
(5.2)

with \( i = 1, \ldots, M \) indexing over the \( M \) sub-event templates and \( j = 1, \ldots, M! \) indexing over the different temporal orders in which these templates can occur.

The cost function depends only on the ordering in which the sub-events occur in the current video, and hence is a look-up table (simple array, \( c = [c_1, \ldots, c_{M!}] \)) with size equal to the number of permutations of the number of sub-events \( M \).

The reason and use of this will become more clear in \$5.3.1\$ when we describe the scoring function.

We learn the model \( \Theta \) with a regularized max-margin hinge loss minimization, given by

\[ \frac{\lambda}{2} \sum_{i=1}^{M} ||w_i||^2 + \frac{1}{|\mathcal{X}|} \sum_{X \in \mathcal{X}} [1 - y_i s_\Theta(X)]_+ \]  
(5.3)
where \([a]_+ = \max(a, 0) \quad \forall a \in \mathbb{R}\). \(s_\Theta(X)\) is our scoring function which uses the templates and the cost function to assign a confidence score to the example \(X\). The decision boundary is given by \(s_\Theta(X) = 0\).

5.3.1 **Scoring function**

Deviating from a linear SVM classifier, which has a single parameter vector, our model has multiple such vectors which act at different temporal positions. We propose to score a video \(X\), with model \(\Theta\), as

\[
\begin{align*}
    s_\Theta(X) &= \max_k \frac{1}{M} \sum_{i=1}^{M} w_i^T x_{k_i} + c_\sigma(k) \\
    \text{s.t.} \quad O(k) &\leq \beta
\end{align*}
\]

(5.4a)

(5.4b)

where, \(k = [k_1, \ldots, k_M] \in \mathbb{N}^M\) are the \(M\) latent variables, and \(\sigma : \mathbb{N}^M \to \mathbb{N}\) maps \(k = (k_1, \ldots, k_M)\) to an index, with lexicographical ordering e.g. with \(M = 4\) and without loss of generality \(k_1 < k_2 < k_3 < k_4\), \(\sigma(k_1, k_2, k_3, k_4) = 1\), \(\sigma(k_1, k_2, k_4, k_3) = 2\), \(\sigma(k_1, k_3, k_2, k_4) = 3\) and so on. The latent variables take the values of the frames on which the corresponding sub-event templates in the model gives maximal response while being penalized by the cost function for the sequence of occurrence of the sub-events. \(O(k)\) is an overlap function, with \(\beta\) being a threshold, to ensure that multiple \(w_i\)'s do not select close by frames.

Intuitively, we capture the idea that each expression or pain sequence is composed of a small number of prototypical appearances e.g. onset and offset phase for smile, brow lower and cheek raise for pain, or a combination thereof. Each of the \(w_i\) captures such a prototypical appearance, albeit (i) they are learned in a discriminative framework and (ii) are mined automatically, again with a discriminative objective. The cost component \(c\) effectively learns the order in
which such appearances should occur. It is expected to support the likely order of sub-events while penalizing the unlikely ones. Even if a negative example gives reasonable detections of such prototypical appearances, the order of such false positive detections is expected to be incorrect and it is expected to be penalized by the order dependent cost. We later validate such intuitions with qualitative results in §5.4.3.

5.3.2 Learning

We propose to learn the model using a stochastic gradient descent (SGD) based algorithm with analytically calculable sub-gradients. The algorithm, summarized in Alg. 5.2, randomly samples the training set and does stochastic updates based on the current example. Due to its stochastic nature, the algorithm is quite fast and is usable in online settings where the data is not entirely available in advance and arrives with time.

We solve the scoring optimization with an approximate algorithm. We obtain the best scoring frame $x_{k_{i}}$ for $w_{i}$ and remove $w_{i}$ from the model and $x_{f-t}, \ldots, x_{f+t}$ frames from the video; and repeat steps $M$ times so that every $w_{i}$ has a corresponding $x_{k_{i}}$. $t$ is a hyperparameter to ensure temporal coverage by the model – it stops multiple $w_{i}$’s from choosing (temporally) close frames. Once the $k = k_{1}, \ldots, k_{M}$ are chosen we add $c_{\sigma(k)}$ to their average template score.

5.4 Experimental Results

We empirically evaluated the proposed approach on four challenging, publicly available, facial behavior datsets, of emotions, clinical pain and non-verbal behavior, in a weakly supervised setting i.e. without frame level annotations. The four datasets ranged from both posed (recorded in lab setting) to spontaneous
expressions (recorded in realistic settings). We now briefly describe the datasets with experimental protocols used and the performance measures reported.

In the following, we first describe the datasets and their respective protocols and performance measures. We then give quantitative comparisons with our own implementation of competitive existing methods. We then present some qualitative results highlighting the choice of subevents and their orders by the method. Finally, we compare the proposed method with state-of-the-art methods on the datasets used.

**CK+ [70]** is a benchmark dataset for expression recognition, with 327 videos from 118 participants posing for seven basic emotions – anger, sadness, disgust, contempt, happy, surprise and fear. We use a standard subject independent 10 fold cross-validation and report mean of average class accuracies over the 10 folds. It has annotation for the apex frame and thus also allows fully supervised training and testing.

**Oulu-CASIA VIS [154]** is another challenging benchmark for basic emotion classification. We used the subset of expressions that were recorded under the visible light condition. There are 480 sequences (from 80 subjects) and six classes (as CK+ except contempt). It has a higher variability due to differences among subjects. We report average accuracy across all classes and use subject independent folds provided by the dataset creators.

**UNBC McMaster Shoulder Pain [71]** is used to evaluate clinical pain prediction. It consists of real world videos of subjects with pain while performing guided movements of their affected and unaffected arm in a clinical interview. The videos are rated for pain intensity (0 to 5) by trained experts. Following [143], we

---

2 [http://www.consortium.ri.cmu.edu/ckagree/](http://www.consortium.ri.cmu.edu/ckagree/)
4 [http://www.pitt.edu/~emotion/um-spread.htm](http://www.pitt.edu/~emotion/um-spread.htm)
labeled videos as ‘pain’ for intensity above three and ‘no pain’ for intensity zero, and discarded the rest. This resulted in 149 videos from 25 subjects with 57 positive and 92 negative samples. Following [143] we do a standard leave-one-subject out cross-validation and report classification rate at ROC-EER.

LILiR\(^5\) [103] is a dataset of non-verbal behavior such as agreeing, thinking, in natural social conversations. It contains 527 videos of 8 subjects involved in dyadic conversations. The videos are annotated for 4 displayed non-verbal behavior signals-agreeing, questioning, thinking and understanding, by multiple annotators. We generated positive and negative examples by thresholding the scores with a lower and higher value and discarding those in between. We then generated ten folds at random and report average Area under ROC – we will make our cross-validation folds public. This differs from Sheerman et al. [103], who used a very small subset of only 50 video samples that were annotated with the highest and the lowest scores.

### 5.4.1 Implementation Details and Baselines

We now give the details of the features used, followed by the details of the baselines and the parameter settings for the model learning algorithms (proposed and our implementations of the baselines).

**Features.** For our experiments, we computed four types of facial descriptors. We extracted 49 facial landmark points and head-pose information using supervised gradient descent\(^6\) [145] and used them for aligning faces. The first set of descriptors were SIFT-based features, which we computed by extracting SIFT features around facial landmarks and thereafter concatenating them [145, 23]. We aligned the faces into 128 × 128 pixel and extracted SIFT features (using open source vlfeat library

\(^5\)http://www.ee.surrey.ac.uk/Projects/LILiR/twotalk_corpus/
\(^6\)http://www.humansensing.cs.cmu.edu/intraface/download.html
in a fixed window of size 12 pixels. The SIFT features were normalized to unit \( \ell_2 \) norm. We chose location of 16 landmark points around eyes (4), brows (4), nose (2) and mouth (6) for extracting the features. Since SIFT features are known to contain redundant information \[52\], we used Principal Component Analysis to reduce their dimensionality to 24. To each of these frame-level features, we added coarse temporal information by appending the descriptors from next 5 consecutive frames, leading to a dimensionality of 1920. The second features that we used were geometric features \[153, 23\], that are known to contain shape or location information of permanent facial features (e.g. eyes, nose). We extracted them from each frame by subtracting \( x \) and \( y \) coordinates of the landmark points of that frame from the first frame (assumed to be neutral) of the video and concatenating them into a single vector (98 dimensions). We also computed LBP features\(^7\) (with radius 1 and neighborhood 8) that represent texture information in an image as a histogram. We added spatial information to the LBP features by dividing the aligned faces into a \( 9 \times 9 \) regular grid and concatenating the histograms (4779 dimensions) \[113, 57\]. We also considered Convolution Neural Network (CNN) features by using publicly available models of Parkhi et al. \[84\] that was trained on a large dataset for face recognition. We used the network output from the last fully connected layer. However, we found that these performed lower than other features e.g. on Oulu and CK+ datasets they performed about 10% absolute lower than LBP features. We suspected that they are not adapted to tasks other than identity discrimination and did not use them further.

**Baselines.** We report results with 4 baseline approaches. For first two baselines we used average (or mean) and max temporal pooling \[107\] over per-frame facial features along with SVM. Temporal pooling is often used along with spatio-temporal

\(^7\)http://www.cse.oulu.fi/CMV/Downloads/LBPMatlab
Table 5.1. Comparison of LOMo with Baseline methods on 4 facial behavior prediction datasets using SIFT based facial features (see §5.4.1).

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Task</th>
<th>Full Sup.</th>
<th>Mean Pool</th>
<th>Max Pool</th>
<th>MIL</th>
<th>LOMo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cohn-Kanade+</td>
<td>Emotion</td>
<td>91.9</td>
<td>86.0</td>
<td>87.5</td>
<td>90.8</td>
<td>92.0</td>
</tr>
<tr>
<td>Oulu-CASIA VIS</td>
<td>Emotion</td>
<td>75.0</td>
<td>68.3</td>
<td>69.0</td>
<td>69.8</td>
<td>74.0</td>
</tr>
<tr>
<td>UNBC McMaster</td>
<td>Pain</td>
<td>–</td>
<td>67.4</td>
<td>81.5</td>
<td>85.9</td>
<td>74.0</td>
</tr>
<tr>
<td>LILiR</td>
<td>Agree</td>
<td>–</td>
<td>84.7</td>
<td>85.5</td>
<td>77.7</td>
<td>79.4</td>
</tr>
<tr>
<td></td>
<td>Question</td>
<td>–</td>
<td>86.2</td>
<td>84.3</td>
<td>80.7</td>
<td>86.6</td>
</tr>
<tr>
<td></td>
<td>Thinking</td>
<td>–</td>
<td>93.6</td>
<td>88.9</td>
<td>93.8</td>
<td>94.8</td>
</tr>
<tr>
<td></td>
<td>Understand</td>
<td>–</td>
<td>79.4</td>
<td>79.2</td>
<td>78.9</td>
<td>80.3</td>
</tr>
</tbody>
</table>

features such as Bag of Words [56, 108], LBP [155] in video event classification, as it yields vectorial representation for each video by summarizing variable length frame features. We selected Multiple Instance Learning based on latent SVM [4] as the third baseline algorithm. We also computed the performance of the fully supervised algorithms for cases with known location of the frame that contains the expression. For making a fair comparison, we used the same implementation for SVM, MIL and LOMo.

Parameters. We fix $M = 1$ and $c_\sigma = 0$ in the current implementation, for obtaining SVM baseline results with a single vector input, and report best results across both learning rate and number of iterations. For both MIL ($M = 1$) and LOMo, which take a sequence of vectors as input, we set the learning rate to $\eta = 0.05$ and for MIL we set $c_\sigma = 0$. We fix the regularization parameter $\lambda = 10^{-5}$ for all experiments. We do multiclass classification using one-vs-all strategy. For ensuring temporal coverage (see §5.3.2), we set the search space for finding the next sub-event to exclude $t = 5$ and 50 neighboring frames from the previously detected sub-events’ locations for datasets with fewer frames per video (i.e. CK+, Oulu-CASIA VIS and LILiR datasets) and UNBC McMaster dataset, respectively. For our final implementation, we combined LOMo models learned on multiple features using late fusion i.e. we averaged the scores.
5.4.2 Quantitative Results

The performances of the proposed approach, along with those of the baseline methods, are shown in Table 5.1. In this comparison, we used SIFT-based facial features for all datasets. Since head nod information is important for identifying non-verbal behavior such as agreeing, we also appended head-pose information (yaw, pitch and roll) to the SIFT-based features for the LILiR dataset.

We see performance improvements with proposed LOMo, in comparison to baseline methods, on 6 out of 7 prediction tasks. In comparison to MIL, we observe that LOMo outperforms the former method on all tasks. The improvements are $1.2\%, 4.2\%$ and $1.1\%$ absolute, on CK+, Oulu-CASIA VIS and UNBC McMaster datasets, respectively. This improvement can be explained by the modeling advantages of LOMo, where it not only discovers multiple discriminative sub-events but also learns their ordinal arrangement. For the LILiR dataset, we see improvements in particular on the ‘Questioning’ ($5.9\%$ absolute) and ‘Agreeing’ ($1.7\%$ absolute), where temporal information is useful for recognition. In comparison to temporal pooling based approaches, LOMo outperforms both mean and max pooling on 6 out of 7 tasks. This is not surprising since temporal pooling operations are known to add noise to discriminative segments of a video by adding information from non-informative segments [107]. Moreover, they discard any temporal ordering, which is often important for analyzing facial activity [108].

On both facial expression tasks, i.e. emotion (CK+ and Oulu-CASIA VIS) and pain prediction (UNBC McMaster), methods can be arranged in increasing order of performance as mean-pooling, max-pooling, MIL, LOMo. A similar trend between temporal pooling and weakly supervised methods has also been reported by previous studies on video classification [108, 34]. We again stress that
LOMo performs better than the existing weakly supervised methods, which are the preferred choice for these tasks. In particular, we observed the difference to be higher between temporal pooling and weakly supervised methods on the UNBC McMaster dataset, 67.4% for mean-pooling, 81.5% for max-pooling, 85.9% for MIL and 87.0% for LOMo. This is because the subjects exhibit both head movements and non-verbal behavior unrelated to pain, and thus focusing on the discriminative segment, cf. using a global description, leads to performance gain. However, we didn’t notice a similar trend on the LILiR dataset – the differences are smaller or reversed e.g. for ‘Understanding’ mean-pooling is marginally better than MIL (79.4% vs. 78.9%), while LOMo is better than both (80.3%). This could be because most conversation videos are pre-segmented and predicting non-verbal behavior relying on a single prototypical segment might be difficult e.g. ‘Understanding’ includes both upward and downward head nod, which cannot be captured well by detecting a single event. In such cases we see LOMo beats MIL by temporal modeling of multiple events.

![Figure 5.3](image)

**Figure 5.3.** Detection of multiple discriminative sub-events, discovered by LOMo, on a video sequence from the UNBC McMaster Pain dataset. The number below the timeline shows the relative location (in percentile of total number of frames).
5.4.3 Qualitative Results

Fig. 5.4 shows the detections of our approach, with model trained for ‘happy’ expression, on two sequences from the Oulu-CASIA VIS dataset. The model was trained with three sub-events. As seen in Fig. 5.4, the three events seem to correspond to the expected semantic events i.e. neutral, low-intensity and apex, in that order, for the positive example (left), while for the negative example (right) the events are incorrectly detected and in the wrong order as well. Further, the final scores assigned to the negative example is $-2.87$ owing to low detection scores as well as penalization due to incorrect temporal order. The cost learned, by the model, for the ordering $(3, 1, 2)$ was $-0.6$ which is much lower than 0.9 for the correct order of $(1, 2, 3)$. This result highlights the modeling strength of LOMo, where it learns both multiple sub-events and a prior on their temporal order.

Fig. 5.3 shows detections on an example sequence from the UNBC McMaster dataset where subjects could show multiple expressions of pain [108, 96]. The results show that our approach is able to detect such multiple expressions of pain as sub-events.

Thus, we conclude that qualitatively our model supports our intuition, that not only the correct sub-events but their correct temporal order is critical for high performance in such tasks.

5.4.4 Comparison with State-of-the-Art

In this section we compare our approach with several existing approaches on the three facial expression datasets (CK+, Oulu-CASIA VIS and UNBC McMaster). Tab. 5.2 shows our results along with many competing methods on these datasets. To obtain the best performance from the model, we exploited the complementarity of different facial features by combining LOMo models learned on three facial
Figure 5.4. Detections made by LOMo trained ($M = 3$) for classifying ‘happy’ expression on two expression sequences from Oulu-CASIA VIS dataset. LOMo assigns a negative score to the sad expression (on the right) owing to negative detections for each sub-event and also negative cost of their ordering (see §5.3.1). The number below the timeline shows the relative location (in percentile of total number of frames).
Table 5.2. Comparison of the proposed approach with several state-of-the-art algorithms on three datasets.

<table>
<thead>
<tr>
<th>CK+ dataset [70]</th>
<th>Oulu-CASIA VIS dataset [154]</th>
</tr>
</thead>
<tbody>
<tr>
<td>3DSIFT [100]</td>
<td>HOG3D [54]</td>
</tr>
<tr>
<td>LBPTOP [155]</td>
<td>LBPTOP [155]</td>
</tr>
<tr>
<td>HOG3D [54]</td>
<td>STM-ExpLet [67]</td>
</tr>
<tr>
<td>Ex-HMMs [107]</td>
<td>Atlases [41]</td>
</tr>
<tr>
<td>STM-ExpLet [67]</td>
<td>Ex-HMMs [107]</td>
</tr>
<tr>
<td>LOMo (proposed)</td>
<td>LOMo (proposed)</td>
</tr>
<tr>
<td>81.4</td>
<td>70.6</td>
</tr>
<tr>
<td>89.0</td>
<td>72.1</td>
</tr>
<tr>
<td>91.4</td>
<td>74.6</td>
</tr>
<tr>
<td>93.9</td>
<td>75.5</td>
</tr>
<tr>
<td>94.2</td>
<td>75.6</td>
</tr>
<tr>
<td>95.1</td>
<td>82.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>UNBC McMaster dataset [71]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ashraf et al. [73]</td>
</tr>
<tr>
<td>Lucey et al. [73]</td>
</tr>
<tr>
<td>MS-MIL [108]</td>
</tr>
<tr>
<td>MIL-HMM [143]</td>
</tr>
<tr>
<td>RMC-MIL [96]</td>
</tr>
<tr>
<td>LOMo (proposed)</td>
</tr>
<tr>
<td>68.3</td>
</tr>
<tr>
<td>81.0</td>
</tr>
<tr>
<td>83.7</td>
</tr>
<tr>
<td>85.2</td>
</tr>
<tr>
<td>85.7</td>
</tr>
<tr>
<td>87.0</td>
</tr>
</tbody>
</table>

descriptors – SIFT based, geometric and LBP (see §5.4.1). We used late fusion for combination by averaging the prediction scores from each model. With this setup, we achieve state-of-the-art results on the three datasets. We now discuss some representative works.

Several initial methods worked with pooling the spatio-temporal information in the videos e.g. (i) LBPTOP [155] – Local Binary Patterns in three planes (XY and time), (ii) HOG3D [54] – spatio-temporal gradients, and (iii) 3D SIFT [100]. We report results from Liu et al. [67], who used a similar experimental protocol. These were initial works and we see that their performances are far from current method e.g. compared to 81.2% for the proposed LOMo, HOG3D obtains 70.6% and LBPTOP obtains 72.1% on the Oulu-CASIA VIS dataset.

Approaches modeling temporal information include Exemplar-HMMs [107], STM-ExpLet [67], MS-MIL [128]. While Sikka et al. (Exemplar-HMM) [107] compute distances between exemplar HMM models, Liu et al. (STM-ExpLet) [67] learns a flexible spatio-temporal model by aligning local spatio-temporal features in
an expression video with a universal Gaussian Mixture Model. LOMo outperforms such methods on both emotion classification tasks e.g. on Oulu-CASIA VIS dataset, LOMo achieves a performance improvement of 7.5% and 6.5% absolute relative to STM-ExpLet and Exemplar-HMMs respectively. Sikka et al. [108] first extracted multiple temporal segments and then used MIL based on boosting MIL [128]. Chongliang et al. [143] extended this approach to include temporal information by adapting HMM to MIL. We also note the performance in comparison to both MIL based approaches (MS-MIL [108] and MIL-HMM [143]) on the pain dataset. Both the methods report very competitive performances of 83.7% and 85.2% on UNBC McMaster dataset compared to 87.0% obtained by the proposed LOMo. Since having a large amount of data is difficult for many facial analysis tasks, e.g. clinical pain prediction, our results also show that combining, simple but complementary, features with a competitive model leads to higher results.

5.5 Conclusion

We proposed a (loosely) structured latent variable model that discovers prototypical and discriminative sub-events and learn a prior on the order in which they occur in the video. We learn the model with a regularized max-margin hinge loss minimization which we optimize with an efficient stochastic gradient descent based solver. We evaluated our model on four challenging datasets of expression recognition, clinical pain prediction and intent prediction is dyadic conversations. We provide experimental results that show that the proposed model consistently improves over other competitive baselines based on spatio-temporal pooling and Multiple Instance Learning. Further in combination with complementary features, the model achieves state-of-the-art results on the above datasets. We also showed qualitative results demonstrating the improved modeling capabilities of the proposed
method. The model is a general ordered sequence prediction model and we hope to extend it to other sequence prediction tasks.

5.6 Appendix

In this section we present some more results and comments. A supplementary video summarizing this work can be viewed at https://youtu.be/k-FDUxnIifa8.

5.6.1 Effect of Parameters

We study the effect of varying model parameters $\lambda$ and the number of PCA dimensions on the classification performance. We selected Oulu-CASIA VIS and UNBC McMaster datasets and plotted classification accuracies versus different values of the parameters. We can observe from the plots for parameter $\lambda$ in Fig. 5.5 that (i) the results are not very sensitive to $\lambda$, and (ii) LOMo shows consistent improvement over baseline methods for different $\lambda$. Fig. 5.5 also shows performance by varying PCA dimensions and we see that the results for LOMo do not vary significantly with this value as well. It is also possible to obtain better results with LOMo than those reported in this paper by selecting parameters using cross-validation.

5.6.2 Intuitive Example for Understanding the Scoring Function

In order to better understand the scoring function discussed in S5.3.1, we use a simple of example of scoring a video versus scoring the same video with shuffled events. The scoring function has (i) appearance templates scores and (ii) ordering cost. It will score each shuffled video equally with the appearance templates as the appearances of the sub-events were not changed. The ordering costs learned
using LOMo will positively score combinations of (N, O, A) and (N, A, O), thus imposing a loose temporal structure and allowing variations; and it will penalize combinations (A, O, N), (O, A, N) and (A, N, O) as these combinations were found unlikely for the smiling while training. Such ordering cost will negatively score expressions that don’t belong to the target class but managed to get decent scores from the appearance templates (false positives). If we shuffle the order for example shown in Fig. 3a to events (3, 1, 2) instead of (1, 2, 3), then its score decreases to $-0.06$, as learned ordering costs were $0.9$ for $(1, 2, 3)$ and $-0.6$ for $(3, 1, 2)$, and the total appearance score was $0.54$. This property also adds robustness to our algorithm in discriminating between visually similar expressions (e.g. happy and fear) by using the temporal ordering cost.

5.6.3 Visualization of Detected Events

For better understanding the model, we show the frames corresponding to each latent sub-event as identified by LOMo across different subjects. Ideally each sub-event should correspond to a facial state and thus have a common structure.
Semantic coherence between detected events across samples.

Event 1 (detected across multiple videos)

Event 2 (detected across multiple videos)

Event 3 (detected across multiple videos)

Figure 5.6. Frames corresponding to latent sub-events as identified by our algorithm on different subjects. This figure shows results for LOMo trained for classifying ‘happy’ expression and tested on new samples belonging to the ‘happy’ class.

across different subjects. As shown in Fig. 5.6, we see a common semantic pattern across detected events where event 1 seems to be similar to neutral, event 2 to onset and event 3 to apex. Although we have only shown results for LOMo trained to classify ‘happy’ expression, we observed similar trend across other classes.

5.6.4 Additional Quantitative Results

In addition to the results shown in Fig. 5.4, we have also shown results for LOMo trained to classify ‘disgust’ expression on another subject in Fig. 5.7. We have shown results for samples belonging to ‘disgust’ class and ‘sad’ class due to higher confusions between the two classes. In Fig. 5.8, we have shown results from our algorithm on another example from the UNBC McMaster dataset.
5.7 Acknowledgement

Support for this work was provided by NIH grant NIH R01NR013500. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding agency.

Chapter 5, in part, is a reprint of the material as it appears in Computer Vision and Pattern Recognition Conference 2016. Sikka, Karan; Sharma, Gaurav; Bartlett, Marian., copyright IEEE, 2016. The dissertation author was the primary investigator and author of this paper.
Figure 5.7. Detections made by LOMo trained \((M = 3)\) for classifying ‘disgust’ expression on two expression sequences from Oulu-CASIA VIS dataset. LOMo assigns a negative score to the sad expression (on the bottom) owing to negative detections for each sub-event and also negative cost of their ordering (see §5.3.1). The number below the timeline shows the relative location (in percentile of total number of frames).
Figure 5.8. Detection of multiple discriminative sub-events, discovered by LOMo, on a video sequence from the UNBC McMaster Pain dataset. The number below the timeline shows the relative location (in percentile of total number of frames).
Chapter 6

Conclusion and Future Work

In this dissertation, we targeted the problem of facial behavior prediction in videos by improving upon two particular modeling challenges that were unaddressed by previous methods. In particular most previous state-of-the-art methods are based on global temporal pooling, which assumes the presence of a single uniform action across the entire sequence and moreover squashes all the information across the temporal dimension by pooling. The first modeling challenge was related to training models with weak labels that provide a global video-level label for the entire video. This is often the case with real-world datasets where it is easier to obtain coarse labels in comparison to fine-grained labels such as precise timestamps of facial behavior or even their muscle activations. The second modeling aspect was related to incorporating temporal dynamics of facial expressions during predictions. Temporal information is important since facial expressions are dynamic events and may be composed of sub-events or phases such as neutral, onset and offset or their combinations. We proposed 4 statistical algorithms based on Latent Weakly Supervised Learning Models (LWSLM) that tried to addresses both these modeling challenges and showed improvements on current state-of-the-art facial behavior classification.

In Ch. 2 we discussed an algorithm to tackle these issues by combining
Multiple Instance Learning (MIL) framework with Multiple Segment Representation (MSMIL). The MIL framework modeled weak labels while Multiple Segment representation handled the issue of temporal dynamics by representing a video as a bag containing video segments at multiple temporal scales and timepoints. MSMIL algorithm jointly classified and localized the target expression in a sequence despite training with only video-level labels. We evaluated this algorithm on the challenging problem of pain expression classification on the UNBC McMaster Shoulder Pain dataset and showed significant improvement relative to prior art (+2.7% absolute) and relative to global temporal pooling (+15% versus mean pooling and +2.2% versus max pooling). We also evaluated it on the localization task of predicting per-frame pain labels while only training with weak video-level labels. We observed improvements against weakly supervised per-frame pain prediction algorithm (+2.1% absolute) while being comparable to fully supervised per-frame prediction algorithm that didn’t use any Multiple Segments. We concluded the chapter by showing some visualizations of the per-frame predictions made by MSMIL and also the discriminative facial regions it was able to learn for predicting pain expressions.

MSMIL being based on the MIL framework assumed independence between temporal segments and only learned a single discriminative event, and thus didn’t model any temporal structure. In the next work discussed in Ch. 3 we addressed this concern by explicitly modeling temporal evolution using latent sequential models. We learned an exemplar Hidden Markov Model (EHMM) for each video sample to model both the appearances of sub-events and their temporal structure as first order transition matrices. We then used Probabalistic Kernels to calculate distances between EHMM models and trained a Kernel Support Vector Machine (SVM) classifier on these distances. The Probabalistic Kernel has an advantage over global temporal pooling since it principally compares both static (appearance) and
temporal components of two sequences by calculating the probability of all possible state evolutions undertaken by the two HMM models together up to a time-step. We empirically showed improvement over relevant baselines—HMMs trained for each class (+12% on Oulu-CASIA VIS dataset) and global temporal pooling (+4% versus mean pooling on Oulu-CASIA VIS dataset). This was the first step towards explicitly addressing temporal dynamics, however this model was limited by its modeling capacity owing to the assumption of Gaussian distribution for HMM observations.

We also discussed an alternative approach in Ch. 5 to improve MIL algorithm that only models a single discriminative instance by extending it to model multiple concepts. This approach is based on Embedding based MIL algorithms that first use a concept space to embed a bag into concept space and then learn a classifier over this space. Compared to previous methods, that learned concepts and classifier in disjoint steps, we propose a new approach, referred to as Joint Clustering and Classification for MIL (JC^2MIL), that jointly learned the latent space of multiple concepts and the classifier. We showed consistent improvements relative to standard MIL approaches and previous Embedding Space based MIL approaches on standard MIL datasets on scene and object classification (+3% improvement on Corel-2000 dataset relative to prior art). We also discussed the advantages of jointly learning the latent concept space and the classifier instead of learning them disjointly as in the later setting the latent space may not be completely aligned with the final classification task.

Both JC^2MIL and EHMM proposed to extend standard MIL algorithms to model richer latent structures by either modeling multiple concepts or temporal structure or both. However, both of these algorithms were limited in their modeling capabilities. For example JC^2MIL didn’t model any temporal aspect, while EHMM
was limited in the dimensionality of the underlying feature representation. In the final Chapter (Ch. 5) we proposed a unified framework that generalized the MIL algorithm and combined the strength of these two approaches, namely (i) weakly supervised learning, (ii) multiple concepts (or sub-events) and (iii) temporal ordering. This approach, referred to as Latent Ordinal Models for Facial Analysis in Videos (LOMo), is based on Latent SVM formulation that jointly learns to model each sequence as a set of (discovered) sub-events with a cost associated with their temporal ordering. Building upon Discriminative Part Based Models that impose stronger penalties on the location of latent parts, our approach only ‘loosely’ models the temporal ordering between these sub-events. We proposed a max-margin hinge loss minimization objective, to learn the model and design an efficient stochastic gradient descent based learning algorithm. We evaluated the algorithm on 4 challenging human facial behavior classification datasets and reported improvement over both global temporal pooling based baselines (+5.7% versus mean pooling and +4% versus MIL on Oulu-CASIA VIS dataset). In combination with complementary features we also reported state-of-the-art results on Oulu-CASIA VIS, McMaster and CK+ datasets. We also showed some visual examples of the learned latent space of sub-events that seem to correspond to facial states such as neutral, onset and apex. Since the LOMo model unified the modeling improvements discussed in previous Chapters in a single framework, it is possible to compare the advantages of these complementary improvements step by step. On Oulu-CASIA VIS dataset we can see the performance improvement of including both multiple latent events and temporal ordering by observing LOMo’s performance of 74.0% versus 69.8% of standard MIL. When training the model without any temporal ordering performance of LOMo falls down by 1% (with multiple sub-events). While the difference between LOMo trained with a single
concept and 3 concepts and not using any temporal ordering is 4.5%, clearly showing
the advantages of using both the ideas introduced in this work.

In conclusion we showed the advantages of modeling both weak labels
and temporal ordering using approaches based on LWSLM on both posed and
spontaneous human facial behavior video datasets. We also showed that superior
performance is achieved by an algorithm that tackled both the modeling challenges
by unifying joint learning of latent structures- multiple concepts and temporal
ordering, in a Weakly Supervised Learning framework.

6.0.1 Future Work

With the recent progress in Deep Learning, the first possible direction is to
extend LOMo to a Deep architecture that not only learns latent information but also
low-level feature. Different concepts in the LOMo framework could share features
in the first few layers of the network but then concept specific features are learned
in higher layers. A similar idea has also been used in Spatial Transformer networks
where different parts of the network learn to focus on specific discriminative parts
of a bird using multiple Spatial Transformers for fine-grained bird classification
[45]. We have also recently seen the use of Attention network which, similar to
Weakly Supervised Learning, learns to focus on specific parts of an images [146] and
videos [127] for classification. In order to train such Deep Networks we could either
combine losses from multiple datasets (as done in [115]) or learn models on a but
unconstrained related problem of human action classification.

Recently Bilen et al. [12] proposed the use of dynamic images as a new video
representation, where a single image summarizes the entire sequence. This single
dynamic image is generated by applying rank pooling via ranking SVM over the
frames of sequences. We could use the LOMo framework proposed in this work to
perform selective pooling over only a few representative frames of a video. Such a pooling operation will be robust to parts of the video that may not correspond to the target class. A similar observation was shown by Wang et al. [138] who used supervised classifier for identifying non-action frames in a video followed by rank pooling.


[77] Daniel McDuff, Rana El Kaliouby, Thibaud Senechal, Mohammed Amr, Jeffrey F Cohn, and Rosalind Picard. Affectiva-mit facial expression dataset (am-fed): Naturalistic and spontaneous facial expressions collected” in-the-wild”. In CVPRW, 2013.


