Structured Video Content Analysis: Learning Spatio-Temporal and Multimodal Structures

by

Yale Song

B.Sc. Computer Science and Engineering,
Hanyang University (2008)

S.M. Electrical Engineering and Computer Science,
Massachusetts Institute of Technology (2010)

Submitted to the
Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2014

Signature redacted

Author .....
Yale Song
Department of Electrical Engineering and Computer Science
May 12, 2014

Signature redacted

Certified by .....
Randall Davis
Professor of Computer Science and Engineering
Thesis Supervisor

Signature redacted

Accepted by .....
Leslie A. Kolodziejski
Chair, Department Committee on Graduate Students

Structured Video Content Analysis:
Learning Spatio-Temporal and Multimodal Structures

by

Yale Song

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

Abstract

Video data exhibits a variety of structures: pixels exhibit spatial structure, e.g., the same class of objects share certain shapes and/or colors in image; sequences of frames exhibit temporal structure, e.g., dynamic events such as jumping and running have a certain chronological order of frame occurrence; and when combined with audio and text, there is multimodal structure, e.g., human behavioral data shows correlation between audio (speech) and visual information (gesture). Identifying, formulating, and learning these structured patterns is a fundamental task in video content analysis.

This thesis tackles two challenging problems in video content analysis – human action recognition and behavior understanding – and presents novel algorithms to solve each: one algorithm performs sequence classification by learning spatio-temporal structure of human action; another performs data fusion by learning multimodal structure of human behavior.

The first algorithm, hierarchical sequence summarization, is a probabilistic graphical model that learns spatio-temporal structure of human action in a fine-to-coarse manner. It constructs a hierarchical representation of video by iteratively summarizing the video sequence, and uses the representation to learn spatio-temporal structure of human action, classifying sequences into action categories. We developed an efficient learning method to train our model, and show that its complexity grows only sublinearly with the depth of the hierarchy.

The second algorithm focuses on data fusion – the task of combining information from multiple modalities in an effective way. Our approach is motivated by the observation that human behavioral data is modality-wise sparse, i.e., information from just a few modalities contain most information needed at any given time. We perform data fusion using structured sparsity, representing a multimodal signal as a sparse combination of multimodal basis vectors embedded in a hierarchical tree structure, learned directly from the data. The key novelty is in a mixed-norm formulation of regularized matrix factorization via structured sparsity.

We show the effectiveness of our algorithms on two real-world application scenarios: recognizing aircraft handling signals used by the US Navy, and predicting people's impression about the personality of public figures from their multimodal behavior. We describe the whole procedure of the recognition pipeline, from the signal acquisition to processing, to the interpretation of the processed signals using our algorithms. Experimental results show that our algorithms outperform state-of-the-art methods on human action recognition and behavior understanding.

Thesis Supervisor: Randall Davis
Title: Professor of Computer Science and Engineering

This work was supported in part by contract N00014-09-1-0625 from the Office of Naval Research and by grant IIS-1018055 from the National Science Foundation.

Acknowledgments

"Getting an Education from MIT is like taking a drink from a Fire Hose."

Jerome Wiesner, MIT President from 1971 to 1980

This past six years at MIT has been full of excitement and learning. Not only have I learned how to do research and enjoy doing it, I have gained valuable life experience and perspective from faculty members, fellow students, and friends. I cannot thank enough everyone who has made my PhD journey such a pleasant and stimulating one.

First and foremost, I would like to express my deepest gratitude and appreciation to my academic father Professor Randall Davis for his patient guidance and invaluable advice throughout my doctoral study. He has taught me to how become an independent researcher, how to formulate research questions and explore brave new ideas, and how important it is to continually ask "why?" during the entire process of research. Probably the single most valuable lesson I have learned from him is how to provide intuitive explanations of difficult and complex concepts. I cannot imagine what my PhD journey would have been like without him. Thank you, Randy.

I feel incredibly fortunate to have met Professor Louis-Philippe Morency at a conference in Santa Barbara in 2010. Since then, he has been an extraordinary mentor and an even better friend (remember that dancing night, LP!). His feedback and suggestions have played an instrumental role in improving my research ability and productivity. I have extremely enjoyed collaborating with him, and look forward to continuing our collaboration and friendship.

I also would like to thank my thesis committee, Professor William T. Freeman and Dr. John W. Fisher III, for providing constructive comments and encouragement. Our discussions were intellectually stimulating and challenging, yet fun and rewarding.

Many thanks to the members of the Multimodal Understanding Group: Aaron Adler, Sonya Cates, Chih-Yu Chao, Tom Ouyang, Andrew Correa, Ying Yin, Jeremy Scott, Dan-

ica Chang, Kristen Felch, James Oleinik, Lakshman Sankar, Kaichen Ma, and William Souillard-Mandar. I only wish we had even more time to enjoy and have fun!

To the members of the Korean Graduate Student Association at MIT and Harvard, thank you all for making my life in Boston/Cambridge so much more enjoyable and memorable. Special thanks to the K-CSAIL'ers Byoungkwon An, Taeg Sang Cho, Eunsuk Kang, Deokhwan Kim, Juho Kim, Taesoo Kim, Yongwook Bryce Kim, Jae Wook Lee, Joseph Lim, and Sejoon Lim; the 82'ers Sinbae Kim, Yeonju Alice Lee, Changwook Min, Bumjin Namkoong, and Sungjun Park; the korgrad07'ers Jinyoung Baek, Hyowon Gweon, Jongsup Hong, SeungHyuck Hong, Jeong Hwan Jeon, ShinYoung Kang, Braden Kim, Jae Chul Kim, Jae Kyung Kim, Sungmin Kim, Jae Hoon Lee, and Jong-Mi Lee; and my roommate Jongwoo Lee.

To the members of the Humans and Automation Lab: Professor Missy Cummings, Mark Ashdown, Yves Boussemart, Brian Mekdeci and Tuco, and Jason Ryan. I tremendously enjoyed working with you and sharing an office together. In particular, I cannot thank enough Carl Nehme who has helped me to start my graduate student life at MIT.

I would like to acknowledge my musician friends Aaron Ellerbee, Woojung Han, Jia Yi Har, Justin Liang, Daniel Park, Rob Weiss, and Eric Winokur. Our weekly jam at SidPac was the driving force behind my work. I hope our path cross again and have a jam together.

On top of all this, I thank my parents to whom I am eternally grateful. Especially mom, I admire your tough spirit. You are the role model of my life, and I love you more than words ever could tell. I also thank my brother who has been a best friend of mine and has had my back no matter what.

Last but certainly not least, I thank my dearest fiancée Jennifer Eugine Rhee for filling my life with joy and happiness. Jenn, it is because of you that I was able to make it this far, and I dedicate this thesis to you. I am truly excited for our wedding in Seoul on May 24, 2014, and I look forward to spending the rest of my life with you, caring, loving, and always being there for you!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Video content analysis is an umbrella term that encompasses detection, tracking, recognition, and understanding of objects and their behaviors in video. It has a wide variety of applications including action recognition [1], sentiment analysis [129], video summarization [75], and visual surveillance [47]. With an increasing amount of video available both online and offline, processing all of that video content manually has become more expensive and extremely time consuming. This in turn has dramatically increased the need for systems capable of automatically analyzing video with far less human intervention.

We believe that one key to doing this is taking advantage of structures in video: pixels exhibit *spatial* structure, e.g., the same class of objects share certain shapes and/or colors in an image; frames have *temporal* structure, e.g., dynamic events such as jumping and running have a certain chronological order of frame appearance; and when combined with audio, there is *multimodal* structure, e.g., human multimodal signals show correlation between audio (speech) and visual information (facial expression and body gesture) [83]. A fundamental task here is therefore identifying, formulating, and learning structured patterns in different types of video contents.

This thesis tackles two challenging problems in video content analysis: human action recognition and behavior understanding. The main focus is in the development of algorithms

that exploit structures for video content analysis. In particular, we present two novel machine learning algorithms: one algorithm performs sequence classification by learning spatio-temporal structure of human action; another performs data fusion by learning multimodal structure of human behavior.

Human action recognition aims to classify body movements into a set of known categorizes. We developed a novel machine learning algorithm for learning spatio-temporal structures in human actions. We formulate action recognition as sequence classification and develop a latent variable discriminative graphical model to tackle the problem. Our approach constructs a hierarchical representation of video by iteratively "summarizing" the contents in a fine-to-coarse manner, and uses the representation to learn spatio-temporal structure of human action. We evaluate the algorithm on a task of recognizing a visual vocabulary aircraft handling signals used by the US Navy.

Human behavior understanding, on the other hand, aims to understand subtle affective states of humans based on multimodal signals. We developed an algorithm that formulates data fusion as a mixed-norm regularized matrix factorization with structure sparsity. Our algorithm combines signals from multiple modalities and transforms it into a sparse representation, preserving the most informative multimodal structure in a succinct manner. The key idea is to construct a dictionary of basis vectors embedded in a hierarchy of multimodal subspaces induced by the superset/subset relations among the subspaces. We evaluate the algorithm on the task of predicting people's impression about the personality of public figures from their multimodal behaviors displayed in short video clips.

This chapter gives an overview of the thesis and highlights key contributions. Some of the material presented in this thesis has been appeared in previous publications [100, 99, 101, 104]. Section 1.4 gives a full list of our publications and briefly explains contributions in each of them in the context of this thesis.

1.1 Video Content Analysis: Two Examples

Over the years, the amount of video has been substantially increased both online and offline. According to YouTube statistics[1], 100 hours of video are uploaded to YouTube every minute, and 6 billion hours of video are watched by one billion unique users every month. Consumer products with high quality video cameras have become widespread, with Apple selling 150 million iPhones and 71 million iPads worldwide in 2013 alone[2], making it easy to produce and share personal video. Surveillance cameras are becoming widespread for public safety and security, with an estimated 30 million surveillance cameras now deployed in the United States recording 4 billion hours of footage every week.[3]

There are just as many types of applications as the amount of video available. One of the challenges with a large amount of data is information retrieval: video indexing and retrieval is crucial to the user experience in online video sites [98]. Video summarization provides an easy way to review lengthy video quickly, useful for searching a short footage and generating a preview of video [75]. Visual surveillance systems help detect abnormal objects and behaviors, analyze traffic flow and congestion, and identify people from video [47].

The large amount of video makes manual processing of video contents prohibitive. The current video indexing technology relies heavily on text (e.g., meta data such as title, description, user tags, etc.), but text information is often sparse, making it desirable to utilize the content of video. Also, the manpower required to constantly monitor the surveillance footage is extremely costly, and thus many video feeds are left unmonitored [60]. An automated visual surveillance system that works reliably would help dramatically improve public safety and security.

Below we introduce two practical applications of video content analysis – human action recognition and behavior understanding – that will be used as motivating examples throughout the thesis.

[1]http://www.youtube.com/yt/press/statistics.html. Retrieved 05/12/2014
[2]http://mashable.com/2013/10/28/apple-150-million-iphones/. Retrived 05/12/2014.
[3]http://www.popularmechanics.com/technology/military/4236865. Retrieved 05/12/2014

Figure 1-1: The US Navy plans to deploy drones to aircraft carriers by 2019. Shown above is the Northrop Grumman X-47B designed for carrier-based operations. Our ultimate goal is to enable drones to understand the same visual vocabulary of aircraft handling signals currently used for communication between pilots and aircraft marshallers.

1.1.1 Action Recognition: Aircraft Handling Signals

The US Navy plans to deploy drones (e.g., the X-47B shown in Figure 1-1) to aircraft carriers by 2019.[4] A question arises as to how to enable natural communication between aircraft marshallers and the drones. One solution is to provide remote control devices to the marshallers; but this requires special training, increasing the burden on the marshallers, and means handling manned and unmanned vehicles differently.

The Naval Air Training and Operating Procedures Standardization (NATOPS) manual standardizes a visual vocabulary of aircraft handling signals for communication between marshallers and pilots on US Navy aircraft carriers. Our ultimate goal is to make drones able to understand the same set of visual signals currently used for communication between marshallers and human pilots. Such technology would need to be as robust as human pilots, but it would allow more seamless integration of the drones to the carrier deck environment, minimizing changes to the existing (already challenging) system.

To study this problem, we collected a dataset of 24 aircraft handling signals currently

[4]http://en.wikipedia.org/wiki/Northrop_Grumman_X-47B. Retrieved 05/12/2014

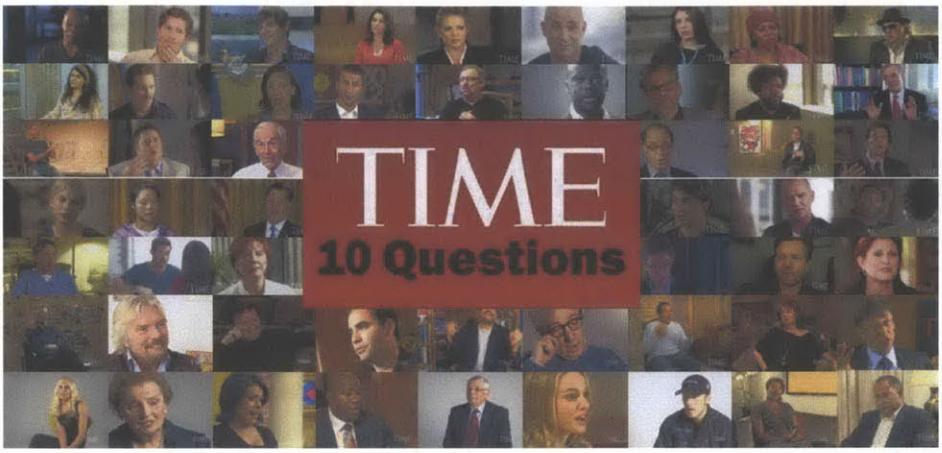


Figure 1-2: The Time10Q dataset collected from YouTube. The goal is to predict people's impression on the personality of public figures based on their multimodal behavior recorded in a short video clip.

used by the US Navy. We describe this dataset in Chapter 2 and recognition procedure in Chapter 3.

1.1.2 Behavior Understanding: Personality Impression

Imagine an AI system that understands human behavior and provides feedback on conversational skills based on behavior recorded in video [46], that harvests public opinions on consumer products based on product review video [120], and that predicts the effectiveness of TV commercials by analyzing facial responses of the audience [72].

Automatic human behavior understanding from video content is an essential component in these systems [129]. In this thesis, we tackle the problem of recognizing personality impression. How someone's personality is perceived by others tremendously affects interpersonal communication, e.g., when making friends [51], in a job interview [126], and political campaigns [34]. Our goal is to develop a system that predicts people's impression of someone's personality, based on their multimodal behavior recorded in a short video clip.

To study this problem, we collected a set of 866 video clips containing interviews with 149

public figures with a variety of professions (from actors to religious leaders; see Figure 1-2). Based on a well-developed theory in psychology called the Big Five personality traits [25, 71], we crowd-sourced people's impression about the personality of the public figures, collecting 8,660 responses. Details are given in Chapter 4 and Chapter 5.

1.2 Learning Structures in Video

We believe that taking advantage of structure in video is one key to the success of automatic video content analysis. In this thesis, we focus on two forms of structures in video, spatio-temporal and multimodal, described below.

1.2.1 Spatio-Temporal Structure of Human Action

When analyzing video contents we are often interested in how certain objects move in space and time. Action recognition is one such scenario where the goal is to discriminate different classes of human actions based on body movement [1]. Central to this problem is learning spatio-temporal structure from video, that is, how spatially coherent groups of pixels (corresponding to body parts) change their location over a period of time. In this thesis, we focus on learning spatio-temporal structure in the context of sequence classification, categorizing each sequence (not each frame) into one of the known categories.

Human actions are often composed of sub-actions, with each sub-action again similarly decomposable. This recursive decomposition creates a hierarchical structure of an action with multiple layers of abstractions, such as the one shown in Figure 1-3. Imagine a video showing someone cooking a steak. It may contain several coarse-level sub-actions (e.g., preparation, grilling, and serving) and each sub-action may again contain finer-level sub-actions (e.g., chopping up unions, seasoning the meat, and heating the grill). The sub-actions at each level represent actions at different spatio-temporal granularities, providing a description of an action with different levels of abstraction.

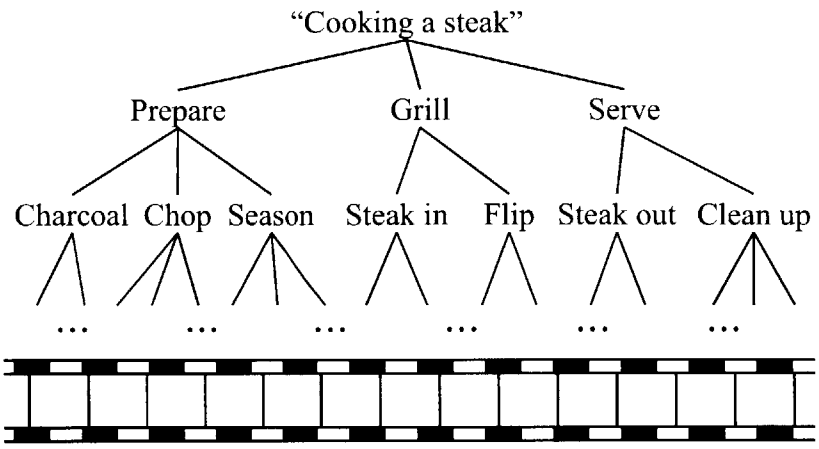


Figure 1-3: Hierarchical decomposition of a fictitious action category "cooking a steak". In some action recognition tasks, we are given a single categorical label describing the overall content of the action sequence, while mid-level sub-action labels are typically unknown.

We show that the hierarchical structure of human actions provides a means of improving recognition performance beyond those of linear-chain graphical models (e.g., Hidden Conditional Random Fields (HCRF) [86]). Note that the labels for sub-actions are not known a priori, nor does the system know the corresponding segmentation information. However, we hope to discover them in an unsupervised fashion using latent variables. The latent variables in a linear-chain model represent hidden states of an action, which can be interpreted as sub-action labels. An algorithm that builds a hierarchy in a bottom-up fashion, from the original sequence to progressively coarser-grained representation, may reveal the hierarchical structure of an action. We describe the algorithm in Chapter 3.

1.2.2 Multimodal Structure of Human Behavior

Human behavior is inherently multimodal: we express our intents and thoughts via speech, gesture, and facial expressions. It also involves a complex interplay of multiple modalities, e.g., neither speech nor facial expression alone entirely conveys the true intention behind sarcastic expressions (e.g., "that was great" said with an uninterested facial expression); the sarcasm is conveyed only through multimodal signal as a whole.

To get a machine to understand natural human behavior from video, we need algorithms able to sense, learn, and infer from multiple modalities. Central to this problem is data fusion: how should we combine information from multiple modalities in such a way that it makes the best use of the richness of information available?

In this thesis, we perform data fusion by learning the dependence/independence structures across modalities, capturing the patterns that are shared across modalities and patterns that are private to each modality. In Chapter 5, we describe an algorithm that factorizes a multimodal signal space into modality-shared and modality-private subspaces, and learns a hierarchical structure among the subspaces induced by superset/subset relations.

1.3 Thesis Outline

We organize the thesis largely into two parts: (i) learning spatio-temporal structure of action recognition, and (ii) learning multimodal structure of human behavior. The first part includes Chapter 2 and Chapter 3, the second part includes Chapter 4, Chapter 5, and Chapter 6. Below we give an overview of each chapter.

Chapter 2. Action Recognition: Aircraft Handling Signals

We developed an action recognition system able to recognize aircraft handling signals used by the US Navy. The system starts by tracking upper body postures and hand shapes simultaneously. Upper body postures are reconstructed in 3D space using a kinematic skeleton model with a particle filter [50], combining both static and dynamic description of body motion as the input feature to make tracking robust to self-occlusion. The reconstructed body postures guide searching for hands. Hand shapes are classified into one of four canonical hand shapes (hands opened/closed, thumb up/down) using a Support Vector Machine (SVM) [113]. Finally, the extracted body and hand features are combined and used as the input feature for action recognition.

In our earlier work [101], we posed the task as an online sequence labeling and segmentation problem. A Latent-Dynamic Conditional Random Field (LDCRF) [77] is used with a temporal sliding window to perform the task continuously. We augmented this with a novel technique called multi-layered filtering, which performs filtering both on the input layer and the prediction layer. Filtering on the input layer allows capturing long-range temporal dependencies and reducing input signal noise; filtering on the prediction layer allows taking weighted votes of multiple overlapping prediction results as well as reducing estimation noise. On the task of recognizing a set of 24 aircraft handling signals online from continuous video input stream, our system achieved an accuracy of 75.37%.

We have explored various other approaches to recognize the aircraft handling signals, including the hierarchical sequence summarization approach [104] we present in this thesis. We briefly introduce the approach below and detail it in Chapter 3.

Chapter 3. Learning Spatio-Temporal Structure of Action Recognition

We present a hierarchical sequence summarization approach for action recognition that constructs a hierarchical representation of video by iteratively "summarizing" the contents in a bottom-up fashion, and uses it to learn spatio-temporal structure from each of the summary representations. Each layer in the hierarchy is a temporally coarser-grained summary of the sequence from the preceding layer. Intuitively, as the hierarchy builds, we learn ever more abstract spatio-temporal structure.

Our approach constructs the hierarchical representation by alternating two steps: sequence learning and sequence summarization.

The goal of the sequence learning step is to learn spatio-temporal patterns in each layer in the hierarchy. We learn the patterns using an HCRF [86] with a modification on the feature function to accommodate the hierarchical nature of our approach: each super observation is a group of several feature vectors, rather than a single one. We define the super observation feature function to incorporate a set of non-linear gate functions, as used

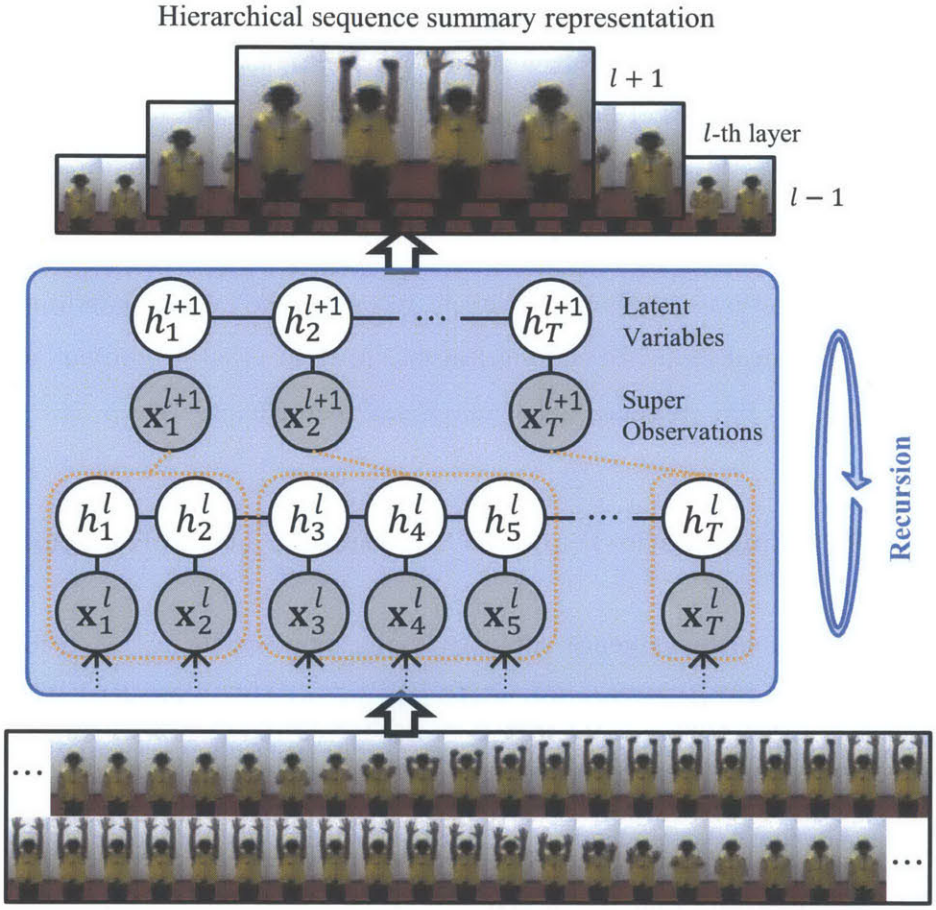


Figure 1-4: We build a hierarchical representation of action sequence and use it to learn spatio-temporal pattern from multiple layers of summary representations. Gray nodes represent super observations, white nodes represent latent variables. Superscripts indicate layer index, subscripts indicate time index.

in neural networks, to learn an abstract feature representation of super observations. The set of gate functions creates an additional layer between latent variables and observations, and has a similar effect to that of the neural network: it learns an abstract representation of super observations, providing more discriminative information for learning spatio-temporal structure of human action.

The goal of the sequence summarization step is to obtain a summary representation of the sequence at a coarser-grained time granularity. We group observations *adaptively*, based on the similarity of hidden states of the observations. We work with hidden states because they are optimized to maximize class discrimination and thus provide more semantically meaningful information. Sequence summarization can be seen as a variable grouping problem with a piecewise connectivity constraint. We use the graph-based variable grouping algorithm by Felzenszwalb *et al.* [37], with a modification on the similarity metric to include latent variables.

We formulate our model, Hierarchical Sequence Summarization (HSS), as the product of the conditional probability distributions computed at each layer. We optimize the model by performing incremental optimization [44], where, at each layer we solve for only the necessary part of the solution while fixing all the others, and iterate the optimization process, incrementing layers.

We evaluated the performance of our HSS model on the NATOPS dataset as well as two other human action datasets, ArmGesture [86] and Canal9 [114]. The results show that our approach outperforms all the state-of-the-art results on the ArmGesture and Canal9 datasets. Notably, our approach achieves a near-perfect accuracy on the ArmGesture dataset (99.59%); on Canal9 dataset we achieve 75.57% accuracy. For the NATOPS dataset, our approach achieved an accuracy of 85.00%, significantly outperforming various previous results using an early-fusion: HMM (from [102], 76.67%), HCRF (from [102], 76.00%), and HCNF (78.33%).

While our HSS approach has shown to work well in action recognition, specifically with

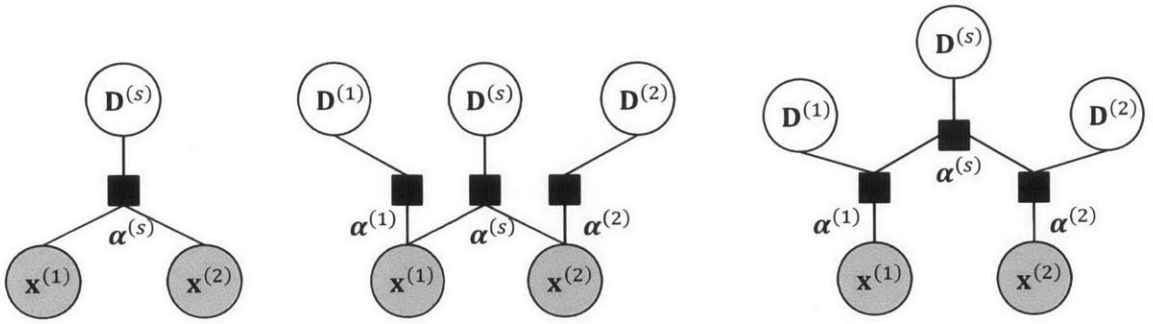
single modality (visual), experimental results show that there is still room for improvement in several directions. The HSS model takes the early-fusion approach for data fusion – feature vectors from different modalities are concatenated to produce one large feature vector – and thus discards structures *among* the modalities, e.g., correlation and interaction. One direction of improvement is therefore leveraging multimodal information, described below.

Chapter 4. Behavior Understanding: Personality Impression

We shift our attention from unimodal (visual) to multimodal aspects of video content analysis, focusing on the task of understanding natural human behaviors recorded in video data. In particular, we describe the problem of personality impression recognition from human behavior. Based on a well-developed theory in psychology called the Big Five personality traits [25, 71], we collected a dataset from YouTube, which we call the Time10Q dataset. The goal is to predict *people's impression* about the personality public figures from their multimodal behavior displayed in short video clips; in other words, instead of predicting the true personality of someone, we focus on predicting how people would perceive the personality of public figures.

Our system uses information from multiple modalities, including body motions, facial expressions, and both verbal and non-verbal speech, to understand the kind of behavior that gives particular impression to people about one's personality. The body motion features are extracted from densely sampled trajectories [116] by computing various local image descriptors including the Histogram of Oriented Gradients (HOG), the Histogram of Optical Flow (HOF), and the Motion Boundary Histogram (MBH). The face features are extracted by detecting and registering the face region to a fixed sized image patch and computing the Pyramid of HOG features [14].

The speech features are extracted both from the verbal and non-verbal perspectives. The verbal speech features are extracted from automatically transcribed captions of video using Latent Dirichlet Allocation (LDA) [11], and the non-verbal speech features are extracted



(a) Shared Subspace (b) Shared-Private Subspace (c) Hierarchical Shared-Private Subspace

Figure 1-5: Factor graph representations of three two-modality subspace models: (a) shared subspace model, (b) shared-private subspace model [54], (c) our hierarchical shared-private subspace model. An input signal from two modalities $[\mathbf{x}^{(1)}; \mathbf{x}^{(2)}]$ is represented in terms of basis vectors from a shared subspace $\mathbf{D}^{(s)}$ and basis vectors from a private subspace $[\mathbf{D}^{(1)}; \mathbf{D}^{(2)}]$ via the coefficient term $\boldsymbol{\alpha}^{(\cdot)}$.

from the acoustic channel of video by computing various prosody features, including the fundamental frequency (F0), the noise-to-harmonic ratio, and the loudness contour.

Information from the four modalities are combined using a novel data fusion algorithm based on structured sparsity, briefly described below and detailed in Chapter 5.

Chapter 5. Learning Multimodal Structure of Human Behavior

Recently, there has been a surge of interest in learning a multimodal dictionary by exploiting group sparsity in multimodal signals. In particular, it has been shown that factorizing a multimodal signal space into parts corresponding to an individual modality and parts that are shared across multiple modalities leads to improvements in multimodal signal understanding [74, 54] (Figure 1-5 (b)). We call such factorized spaces the *multimodal subspaces*, and the two kinds of subspaces modality-*private* and modality-*shared* subspaces, respectively. Intuitively, modality-private subspaces account for the patterns within each modality that are independent of other modalities, while modality-shared subspaces account for the patterns that are dependent on other modalities.

We present a novel data fusion approach by learning multimodal subspaces with a hierarchical structure. We observe that multimodal subspaces have a superset/subset relationship that induces a hierarchical structure of the sort shown in Figure 1-5 (c): a subspace $\mathbf{D}^{(s)}$ is a superset of two subspaces defined over the corresponding modalities $[\mathbf{D}^{(1)}; \mathbf{D}^{(2)}]$, thus $\mathbf{D}^{(s)}$ can be seen as a parent of $[\mathbf{D}^{(1)}; \mathbf{D}^{(2)}]$.

Our intuition is that leveraging this hierarchical structure will enable the multimodal subspaces to capture the dependence/independence relationships across modalities accurately. From the hierarchical structure we can use the hierarchical sparsity rule [53]: a subspace $\mathbf{D}^{(i)}$ will participate in reconstructing an input signal $\mathbf{x}$ (i.e., the corresponding weight term $\boldsymbol{\alpha}^{(\cdot)}$ is non-zero), only if all of its parent subspaces $\mathbf{D}^{(j)}$ are participating as well, where j's are the indices of the parent of the i-th node. The sparsity constraint ensures that only a few paths (from the root to the leaves) are participating in signal reconstruction. We show that this effectively allows the sparse representation to select the most important subset of modalities that best represent the given signal.

We show that it is possible to learn global and local patterns of multimodal data by constructing a multimodal dictionary using the hierarchical sparsity constraint. Two characteristics make this possible: (i) the range of modalities that each subspace covers, and (ii) the frequency of each subspace participating in signal reconstruction. High-level subspaces span over a wider range of modalities than low-level subspaces, and are active more frequently than low-level subspaces in signal construction. These two characteristics encourage high-level subspaces to capture the global patterns shared across multiple modalities, and low-level subspaces to capture the local details narrowed down to specific modalities. For example, a multimodal signal representing "laughing out loud" will be reconstructed as a combination of a high-level subspace "highly aroused" and low-level subspaces "the appearance of mouth" and "the pitch of the voice."

We evaluate our hierarchical multimodal subspace learning approach on the Time10Q dataset. In particular, in order to study the benefit of using structured sparsity for data fusion, we compare our approach to two other sparsity approaches shown in Figure 1-5.

We show that our approach achieves a mean F1 score 0.66, outperforming LASSO [112] (0.64) and the factorized subspace approach by Jia *et al.* [54] (0.65).

Chapter 5. Learning Correlation and Interaction Across Modalities

This chapter describes a framework for learning correlation and interaction across modalities for multimodal sentiment analysis. Our framework is based on Canonical Correlation Analysis [43] (CCA) and Hidden Conditional Random Fields [86] (HCRFs): CCA is used to find a projection of multimodal signal that maximizes correlation across modalities, while a multi-chain structured HCRF is used to learn interaction across modalities. The multi-chain structured HCRF incorporates disjoint sets of latent variables, one set per modality, to jointly learn both modality-shared and modality-private substructures in the data. We evaluated our approach on sentiment analysis (agreement-disagreement classification) from non-verbal audio-visual cues based on the Canal 9 dataset [114]. Experimental results show that CCA makes capturing non-linear hidden dynamics easier, while a multi-chain HCRF helps learning interaction across modalities.

1.4 List of Publications

Some of the material presented in Chapter 2 and Chapter 3 has appeared earlier in an journal article [101] and in conference proceedings [100, 99, 104]; the work in Chapter 4 and Chapter 5 have not been published; and the work in Chapter 6 has appeared in conference proceedings [102, 103]. Below we provide a full list of previous publications, situating them in the context of this thesis work.

Learning Spatio-Temporal Structure of Action Recognition

[100] Yale Song, David Demirdjian, Randall Davis: Tracking body and hands for gesture recognition: NATOPS aircraft handling signals database. In *Proceedings of the IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG) 2011*: 500-506.

- This paper described the NATOPS dataset, including a data collection procedure and algorithms to track 3D upper body postures and hand shapes. The dataset is described in Chapter 2 and used for experiments in Chapter 3.

[99] Yale Song, David Demirdjian, Randall Davis: Multi-signal gesture recognition using temporal smoothing hidden conditional random fields. In *Proceedings of the IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG) 2011*: 388-393

- This paper presented a temporally smoothed HCRF for learning long-range temporal dependence structure in sequential data. It extends the work by Wang *et al.* [118], who captured long-range temporal dependency by concatenating neighboring observations into each observation, increasing the feature dimension. Our work uses a Gaussian kernel to incorporate neighboring observations into each observation, which offers the advantage of not increasing the feature dimension. Experimental results showed that our approach significantly outperforms the approach by Wang *et al.* [118].

[101] Yale Song, David Demirdjian, Randall Davis: Continuous body and hand gesture recognition for natural human-computer interaction. In *ACM Transactions on Interactive Intelligent Systems (2012)*: Volume 2(1), Article 5.

- This paper presented online recognition of the NATOPS aircraft handling signals, classifying each frame into one of action categories. We developed a multi-layered

filtering technique that performs filtering both on the observation layer and the prediction layer. Filtering on the observation layer allows capturing long-range temporal dependencies and reducing signal noise; filtering on the prediction layer allows taking weighted votes of multiple overlapping prediction results as well as reducing estimation noise. We showed that our approach achieves a recognition accuracy of 75.37%.

[104] Yale Song, Louis-Philippe Morency, Randall Davis. Action Recognition by Hierarchical Sequence Summarization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2013*: 3562-3569

- This paper presented hierarchical sequence summarization approach to action recognition, which we describe in Chapter 3. Motivated by the observation that human action data contains information at various temporal resolutions, our approach constructs multiple layers of discriminative feature representations at different temporal granularities. We build up a hierarchy dynamically and recursively by alternating sequence learning and sequence summarization. For sequence learning we use HCRFs [86] to learn spatio-temporal structure; for sequence summarization we group observations that have similar semantic meaning in the latent space. For each layer we learn an abstract feature representation through neural network. This procedure is repeated to obtain a hierarchical sequence summary representation. We developed an efficient learning method to train our model and showed that its complexity grows sublinearly with the size of the hierarchy. Experimental results showed the effectiveness of our approach, achieving the best published results on the ArmGesture and Canal9 datasets.

Learning Multimodal Structure of Human Behavior

[102] Yale Song, Louis-Philippe Morency, Randall Davis: Multi-view latent variable discriminative models for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2012*: 2120-2127

- This paper presented multi-view latent variable discriminative models, an extension to HCRFs [86] and LDCRFs [77] for learning with multimodal data. Our model learns modality-shared and modality-private sub-structures of multimodal data by using a multi-chain structured latent variable model. We showed that our model outperforms both HCRFs and LDCRFs on action recognition datasets with multiple information channels (e.g., the body motion and hand shape from the NATOPS).

[103] Yale Song, Louis-Philippe Morency, Randall Davis: Multimodal human behavior analysis: learning correlation and interaction across modalities. In *Proceedings of the International Conference on Multimodal Interaction (ICMI) 2012*: 27-30

- We extended our multi-view latent variable discriminative model [102] by incorporating Kernel Canonical Correlation Analysis (KCCA) [43] to learn the correlation across modalities, which we describe in Chapter 6. Our approach uses a non-linear kernel to map multimodal data to a high-dimensional feature space and finds a new projection of the data that maximizes the correlation across modalities. The transformed data is then used as an input to our multi-view model [102]. We evaluated our approach on a sentiment analysis task (agreement-disagreement recognition) and showed that our approach outperforms HMM, CRF, HCRF, and our earlier work [102].

[106] Yale Song, Louis-Philippe Morency, Randall Davis: Learning a sparse codebook of facial and body microexpressions for emotion recognition. In *Proceedings of the International Conference on Multimodal Interaction (ICMI) 2013*: 237-244

- This paper presented an approach to obtaining a compact representation of facial and body micro-expressions using sparse coding. Local space-time features are extracted over the face and body region for a very short time period, e.g., few milliseconds. A dictionary of microexpressions is learned from the data and used to encode the features in a sparse manner. This allows us to obtain a representation that captures the most salient motion patterns of the face and body at a micro-temporal scale.

Experiments performed on the AVEC 2012 dataset [93] showed that our approach achieves the best published performance on the expectation dimension based solely on visual features.

Other Contributions

[107] Yale Song, Zhen Wen, Ching-Yung Lin, Randall Davis: One-Class Conditional Random Fields for Sequential Anomaly Detection. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI) 2013*: 1685-1691

- Sequential anomaly detection is a challenging problem due to the one-class nature of the data (i.e., data is collected from only one class) and the temporal dependence in sequential data. This paper presented One-Class Conditional Random Fields (OCCRF) for sequential anomaly detection that learn from a one-class dataset and capture the temporal dependence structure, in an unsupervised fashion. We proposed a hinge loss in a regularized risk minimization framework that maximizes the margin between each sequence being classified as "normal" and "abnormal." This allows our model to accept most (but not all) of the training data as normal, yet keeps the solution space tight. Experimental results on a number of real-world datasets showed our model outperforming several baselines. We also reported an exploratory study on detecting abnormal organizational behavior in enterprise social networks.

[105] Yale Song, Louis-Philippe Morency, Randall Davis: Distribution-sensitive learning for imbalanced datasets. In *Proceedings of the IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG) 2013*: 1-6

- Many real-world face and gesture datasets are by nature imbalanced across classes. Conventional statistical learning models (e.g., SVM, HMM, CRF), however, are sensitive to imbalanced datasets. In this paper we showed how an imbalanced dataset

affects the performance of a standard learning algorithm, and proposed a distribution-sensitive prior to deal with the imbalanced data problem. This prior analyzes the training dataset before learning a model, and puts more weight on the samples from underrepresented classes, allowing all samples in the dataset to have a balanced impact in the learning process. We reported on two empirical studies regarding learning with imbalanced data, using two publicly available recent gesture datasets, the Microsoft Research Cambridge-12 (MSRC-12) [38] and our NATOPS dataset. Experimental results showed that learning from balanced data is important, and that the distribution-sensitive prior improves performance with imbalanced datasets.

Chapter 2

Understanding Human Actions: Aircraft Handling Signals

This chapter introduces the task of recognizing aircraft handling signals from video data. We outline the background of the project, describe the data collection procedure, and explain how we estimate body pose and hand shapes from stereo video data.

2.1 NATOPS Aircraft Handling Signals

This work is a part of a multiple lab-wide project in which a team of MIT faculty members and students is working to develop a next-generation aircraft carrier deck environment where manned and unmanned vehicles (drones) co-exist.

US Navy plans to deploy drones to aircraft carriers by 2019 (e.g., the X-47B, Figure 2-1). A question arises as to how to enable natural communication between aircraft marshallers and drones. One solution is to provide remote control devices to the marshallers; but this requires special training, increasing the burden on the marshallers, and means handling manned and unmanned vehicles differently. A more desirable solution is to enable drones to

Figure 2-1: The US Navy plans to deploy drones to aircraft carriers by 2019. Shown above is the Northrop Grumman X-47B designed for carrier-based operations. Our ultimate goal is to enable drones to understand the same visual vocabulary of aircraft handling signals currently used for communication between pilots and aircraft marshallers.

understand the same visual vocabulary of aircraft handling signals used for communication between marshallers and human pilots. Such technology would need to be as reliable as communicating with human pilots, but would allow more seamless integration of the drones to the carrier deck environment, minimizing changes to the existing (already quite challenging) system.

The Naval Air Training and Operating Procedures Standardization (NATOPS) manual standardizes general flight and operating procedures for US Navy aircraft. One chapter describes aircraft handling signals on a carrier deck environment, a vocabulary of visual signals used for communication between marshallers and pilots for on-the-deck control of aircrafts, e.g., taxing and fueling. Due to the extraordinary noise created by jet engines (reaching 140 dB), this form of visual communication has proven to be effective on an aircraft carrier deck environment.

To study the feasibility of automatic recognition of aircraft handling signals, we collected the NATOPS dataset. It contains 24 aircraft handling signals from the NATOPS manual that the US Navy marshallers most routinely use to communicate with the pilots (Figure 2.1).

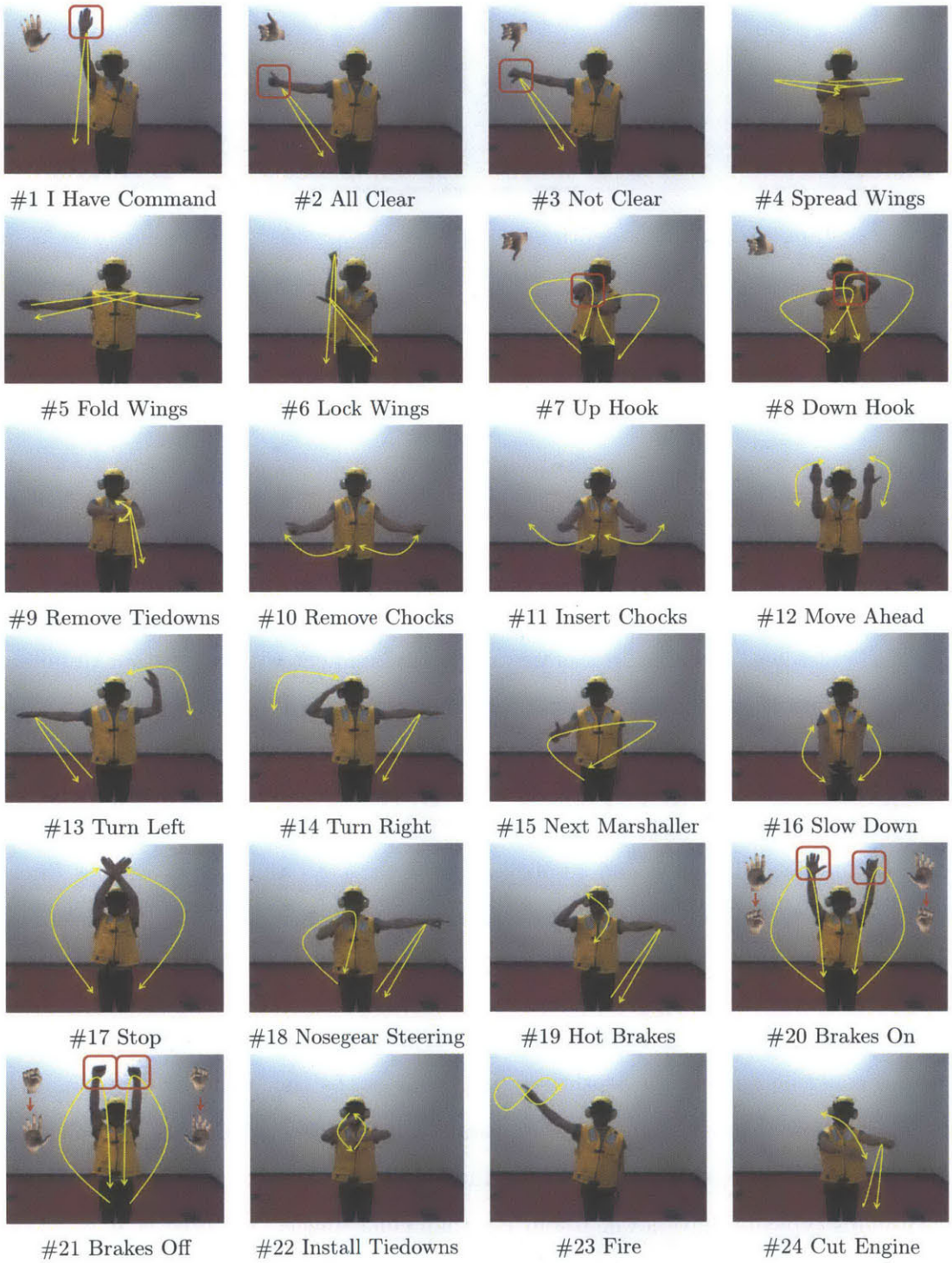


Figure 2-2: Twenty-four NATOPS aircraft handling signals. Body movements are illustrated in yellow arrows, and hand poses are illustrated with synthesized images of hands. Red rectangles indicate hand poses are important in distinguishing the actions with its corresponding similar action pair.

Figure 2-3: Input image (left), depth map (middle), and mask image (right). The "T-pose" shown in the figures is used for body tracking initialization.

2.2 Data Collection

A Bumblebee2 stereo camera from Point Grey Research Inc. was used to record video data, producing 320 x 240 pixel resolution images at 20 FPS. Each of our 20 subjects repeated each of 24 actions 20 times, resulting in 400 sequences for each action class (9,600 sequences in total). The sequence length varied between 1 and 5 seconds, with an average of 2.34 seconds. Video was recorded in a closed room environment with a constant illumination, and with positions of cameras and subjects fixed throughout.

While recording video, we produce depth maps using a manufacture-provided SDK[1] compute mask images using in real-time (see Figure 2-3). Depth maps allow us to reconstruct body postures in 3D space and resolve some of the pose ambiguities arising from self-occlusion; mask images allow us to concentrate on the person of interest and ignore the background, optimizing the use of available computational resources.

We obtain mask images by performing background subtraction. Ideally, background subtraction could be done using depth information alone, by the "depth-cut" method: Filter out pixels whose distance is further from camera than a foreground object, assuming there is no object in between the camera and the subject. However, as shown in Figure 2-3, depth maps typically have lower resolution than color images, meaning that mask images produced from the depth maps would be equally low resolution. This motivates our approach of performing background subtraction using a codebook approach [58], then refining

[1] http://www.ptgrey.com

the result with the depth-cut method.

The codebook approach works by learning a per-pixel background model from a history of background images sampled over a period of time, then segmenting out the "outlier" pixels in new images as foreground. Since this approach uses RGB images, it produces high resolution (per-pixel accuracy) mask images. One weakness of the codebook approach is, however, its sensitivity to shadows, arising because the codebook defines a foreground object as any set of pixels whose color values are noticeably different from the previously learned background model. To remedy this, after input images are background subtracted using the codebook approach, we refine the result using the depth-cut method described above, which helps remove shadows created by a foreground object.

2.3 Feature Extraction

Any pattern recognition task involves a feature extraction step to obtain a vector representation of the data. Two popular feature extraction techniques in human action recognition are body skeleton tracking [95] and low-level visual descriptor tracking such as space-time interest points [65]. The former provides a compact description of body motion (a vector with fewer than 100 dimensions, indicating 3D coordinates of body joints); the latter typically produces a much higher dimensional feature vector (a few hundred to thousands), but is more robust to viewpoint changes and partial body occlusions, and hence more suitable for action recognition "in the wild" [66].

Since the NATOPS dataset has been collected in a controlled lab environment, we can obtain a compact feature representation by skeleton tracking. Using RGBD data collected from a stereo camera, we estimate upper body pose (3D coordinates of left/right elbow and wrist) as well as hand shapes (hand open, hand close, thumb up, thumb down), producing a 20-dimensional vector.

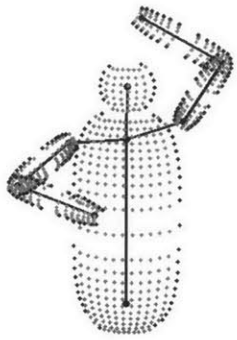


Figure 2-4: Skeleton model of the human upper-body model. The model includes 6 body parts (trunk, head, upper/lower arms for both sides) and 9 joints (chest, head, navel, left/right shoulder, elbow, and wrist).

2.3.1 3D Body Pose Estimation

The goal here is to reconstruct 3D upper-body postures from input images. We formulate this as a sequential Bayesian filtering problem, i.e., having observed a sequence of images $\mathbf{Z}_t = \{\mathbf{z}_1, \cdots, \mathbf{z}_t\}$ and knowing the prior state density $p(\mathbf{x}_t)$, make a prediction about a posterior state density $p(\mathbf{x}_t \mid \mathbf{Z}_t)$, where $\mathbf{x}_t = (x_{1,t} \cdots x_{k,t})$ is a k-dimensional vector representing the body posture at the t-th frame.

Skeleton Upper Body Model

We construct a skeleton upper-body model in 3D space, using a kinematic chain and a volumetric model described by super-ellipsoids [5] (see Figure 2-4). The model includes 6 body parts (trunk, head, upper and lower arms for both sides) and 9 joints (chest, head, navel, left/right shoulder, elbow, and wrist). The shoulder is modeled as a 3D ball-and-socket joint, and the elbow is modeled as a 1D revolute joint, resulting in 8 model parameters in total. Coordinates of each joint are obtained by solving the forward kinematics problem, following the Denavit-Hartenberg convention [28], a compact way of representing n-link kinematic structures. We prevent the model from generating anatomically implausible body postures by constraining joint angles to known physiological limits [78].

We improve on our basic model of the human upper-body by building a more precise model of the shoulder, while still not increasing the dimensionality of the model parameter vector that we estimate. To capture arm movement more accurately, after a body model is generated, the shoulder model is refined analytically using the relative positions of other body joints. In particular, we compute the angle φ between the chest-to-shoulder line and the chest-to-elbow line, and update the chest-to-shoulder angle θ^{CS} as

$$\theta^{CS'} = \begin{cases} \theta^{CS} + \frac{\varphi}{\theta^{CS}_{MAX}} & \text{if elbow is higher than shoulder} \\ \theta^{CS} - \frac{\varphi}{\theta^{CS}_{MIN}} & \text{otherwise} \end{cases} \tag{2.1}$$

where θ^{CS}_{min} and θ^{CS}_{max} are minimum and maximum joint angle limits for chest-to-shoulder joints [78]. Figure 2-4 illustrates our skeleton body model, rendered after the chest-to-shoulder angles θ^{CS} are adjusted (note the left/right chest-to-shoulder angles are different). This simplified model mimics shoulder movement in only one-dimension, up and down, but works quite well if the subject is facing the camera, as is commonly true for human-computer interaction.

With these settings, an upper-body posture is parameterized as $\mathbf{x} = (G\ R)^T$ where G is a 6-dimensional global translation and rotation vector, and R is an 8-dimensional joint angle vector (3 for shoulder and 1 for elbow, for each arm). In practice, once the parameters are initialized, we fix all but (x, z) translation elements of G, making $\mathbf{x}$ a 10-dimensional vector.

Particle Filter Estimation

Human body movements can be highly unpredictable, so an inference that assumes its random variables form a single Gaussian distribution can fall into a local minima or completely loose track. A particle filter [50] is particularly well suited to this type of task for its ability to maintain multiple hypotheses during inference, discarding less likely hypotheses only slowly. It represents the posterior state density $p(\mathbf{x}_t \mid \mathbf{Z}_t)$ as a multi-

modal non-Gaussian distribution, which is approximated by a set of N weighted particles: $\left\{\left(\mathbf{s}_t^{(1)}, \pi_t^{(1)}\right), \cdots, \left(\mathbf{s}_t^{(N)}, \pi_t^{(N)}\right)\right\}$. Each sample $\mathbf{s}_t$ represents a pose configuration, and the weights $\pi_t^{(n)}$ are obtained by computing the likelihood $p(\mathbf{z}_t \mid \mathbf{x}_t = \mathbf{s}_t^{(n)})$, and normalized so that $\sum_{n=1}^{N} \pi_t^{(n)} = 1$.

The joint angle dynamic model is constructed as a Gaussian process: $\mathbf{x}_t = \mathbf{x}_{t-1} + e, \quad e \sim \mathcal{N}(0, \sigma^2)$. Once N particles are generated, we obtain the estimation result by calculating the Bayesian Least Squares (BLS) estimate:

$$\mathbb{E}\left[f(\mathbf{x}_t)\right] = \sum_{n=1}^{N} \pi_t^{(n)} f(\mathbf{s}_t^{(n)}). \tag{2.2}$$

Iterative methods need a good initialization. We initialize our skeleton body model at the first frame: The initial body posture configurations (i.e., joint angles and limb lengths) are obtained by having the subject assume a static "T-pose" (shown in Figure 2-3), and fitting the model to the image with exhaustive search. This typically requires no more than 0.3 seconds (on an Intel Xeon CPU 2.4 GHz machine with 4 GBs of RAM).

Designing Likelihood Function

The likelihood function $p(\mathbf{z}_t \mid \mathbf{x}_t = \mathbf{s}_t^{(n)})$ measures the goodness-of-fit of an observation $\mathbf{z}_t$ given a sample $\mathbf{s}_t^{(n)}$. We define it as an inverse of an exponentiated fitting error $\varepsilon\left(\mathbf{z}_t, \mathbf{s}_t^{(n)}\right)$:

$$p(\mathbf{z}_t \mid \mathbf{x}_t = \mathbf{s}_t^{(n)}) = \frac{1}{\exp\left\{\varepsilon\left(\mathbf{z}_t, \mathbf{s}_t^{(n)}\right)\right\}}. \tag{2.3}$$

The fitting error $\varepsilon\left(\mathbf{z}_t, \mathbf{s}_t^{(n)}\right)$ is a weighted sum of three error terms computed by comparing features extracted from the skeleton body model to the corresponding features extracted from input images. The three features include a 3D visible-surface point cloud, a 3D contour point cloud, and a motion history image (MHI) [13]. The first two features capture discrepancies in static poses; the third captures discrepancies in the dynamics of motion.

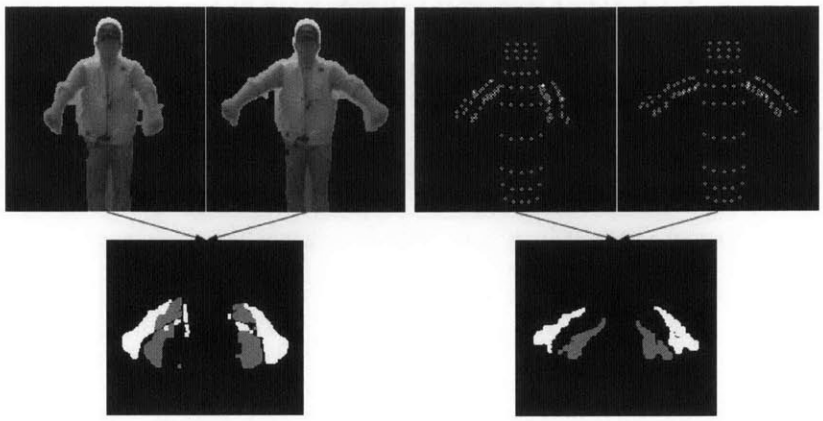


Figure 2-5: Motion history images of the observation (left) and the estimated model (right). White pixel values indicate an object has appeared in the pixel; gray pixel values indicate there was an object in the pixel but it has moved; black pixel values indicate there has been no change in the pixel.

We chose the weights for each error term empirically.

The first two features, 3D visible-surface and contour point clouds, are used frequently in body motion tracking (e.g., [29]) for their ability to evaluate how well the generated body posture fits the actual pose observed in image. We measure the fitting error by computing the sum-of-squared Euclidean distance errors between the point cloud of the model and the point cloud of the input image (i.e., the 3D data supplied by the image pre-processing step described above).

The third feature, an MHI, is an image where each pixel value is a function of the recency of motion in a sequence of images (see Figure 2-5). This often provides useful information about dynamics of motion, as it indicates where and how the motion has occurred. We define an MHI-based error term to measure discrepancies in the dynamics of motion.

An MHI is computed from I_{t-1} and I_t, two time-consecutive 8-bit unsigned integer images whose pixel values span from 0 to 255. For the skeleton body model, I_t is obtained by rendering the model generated by a sample $\mathbf{s}_t^{(n)}$ (i.e., rendering an image of what body posture $\mathbf{s}_t^{(n)}$ would look), and I_{t-1} is obtained by rendering $\mathbb{E}\left[f(\mathbf{x}_{t-1})\right]$, the model generated by the estimation result from the previous step (Equation 2.2). For the input images, I_t

is obtained by converting an RGB input image to YCrCb color space and extracting the brightness channel[2], and this is stored to be used as I_{t-1} for the next time step. Then an MHI is computed as

$$I_{MHI} = \text{thresh}(I_{t-1} - I_t, 0, 127) + \text{thresh}(I_t - I_{t-1}, 0, 255) \tag{2.4}$$

where $\text{thresh}(I, \alpha, \beta)$ is a binary threshold operator that sets each pixel value to β if $I(x, y) > \alpha$, and zero otherwise. The first term captures pixels that were occupied at the previous time step but not in the current time step. The second term captures pixels that are newly occupied in the current time step. We chose the values 0, 127, and 255 to indicate the time information of those pixels: 0 means there has been no change in the pixel, regardless of whether or not there was an object; 127 means there was an object in the pixel but it has moved; while 255 means an object has appeared in the pixel. This allows us to construct an image that concentrates on the moved regions only (e.g., arms), while ignoring the unmoved parts (e.g., trunk, background). The computed MHI images are visualized in Figure 2-5.

Given the MHIs of the skeleton body model and the observation, one can define various error measures. In this work, we define an MHI error as

$$\varepsilon_{MHI} = \mathbb{C}\text{ount}\left[\,\text{thresh}(I', 127, 255)\,\right] \tag{2.5}$$

where

$$I' = \text{abs}\left(I_{MHI}(\mathbf{z}_t, \mathbf{z}_{t-1}) - I_{MHI}(\mathbf{s}_t^{(n)}, \mathbb{E}\left[f(\mathbf{x}_{t-1})\right])\right) \tag{2.6}$$

This error function first subtracts an MHI of the model $I_{MHI}(\mathbf{s}_t^{(n)}, \mathbb{E}[f(\mathbf{x}_{t-1})])$ from an MHI of the observation $I_{MHI}(\mathbf{z}_t, \mathbf{z}_{t-1})$, and computes an absolute-valued image of it (Equation 2.6). Then it applies the binary threshold operator with the cutoff value and result value (127 and 255, respective), and counts non-zero pixels with $\mathbb{C}\text{ount}\left[\cdot\right]$ (Equation 2.5).

[2]Empirically, most of the variation in images is better represented along the brightness axis, not the color axis [18].

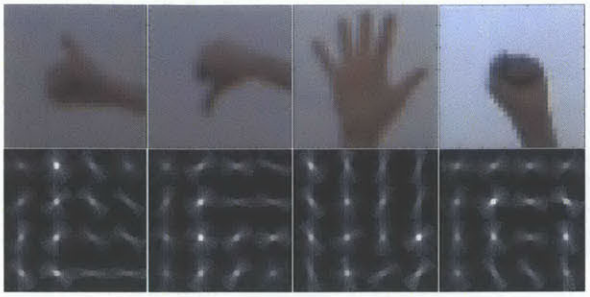


Figure 2-6: Four canonical hand shapes defined in the NATOPS dataset (thumb up and down, palm open and close), and visualization of their HOG features. HOG features are computed with an image size of 32 x 32 pixels, cell size of 4 x 4 pixels, and block size of 2 x 2 cells (8 x 8 pixels), with 9 orientation bins. This results in 16 blocks in total. Bright spots in the visualization indicate places in the image that have sharp gradients at a particular orientation; the orientation of the spot indicates orientation of the gradients.

We set the cutoff value to 127 to penalize the conditions in which two MHIs do not match at the current time-step, independent of the situation at the previous time-step.[3]

2.3.2 Hand Shape Classification

The goal of hand shape classification is to categorize hand shape into one of four canonical shapes: thumb up and down, palm open and close. These are often used in hand signals, particularly on the NATOPS actions used in this work (see Figure 2-6).

Search region

Because it is time-consuming to search for hands in an entire image, we use the information about wrist positions computed in body posture estimation to constrain the search for hands in the image. We create a small search region around each of the estimated wrist positions, slightly larger than the average size of an actual hand image, and search for a hand shape in that region using a sliding window. Estimated wrist positions are of

[3] As mentioned, our error measure in Equation 2.5 concentrates on errors at the current time-step only. However, note that Equation 2.4 also offers information on the errors at the previous time-step as well.

Figure 2-7: Search regions around estimated wrist positions (black rectangles) and clustering of multiple classification results. Our search region was 56 x 56 pixels (outer rectangles); the sliding window was 32 x 32 pixels (inner rectangles). Inner rectangles indicate clustered results (blue/red: palm open/close), and small circles are individual classification results (best viewed in color).

course not always accurate, so a search region might not contain a hand. We compensate for this by including information on hand location from the previous step's hand shape classification result. If a hand is found at time $t-1$, for time t we center the search region at the geometric mean of the estimated wrist position and the hand position at time $t-1$. Our search region was 56 x 56 pixels; the sliding window was 32 x 32 pixels (see Figure 2-7).

HOG features

HOG features [39, 27] are image descriptors based on dense and overlapping encoding of image regions. The central assumption of the method is that the appearance of an object is rather well characterized by locally collected distributions of intensity gradients or edge orientations, even without having the knowledge about the corresponding gradient or edge positions that are globally collected over the image.

HOG features are computed by dividing an image window into a grid of small regions (cells), then producing a histogram of the gradients in each cell. To make the features less sensitive to illumination and shadowing effects, the image window is again divided into a grid of larger regions (blocks), and all the cell histograms within a block are accumulated for normalization. The histograms over the normalized blocks are referred to as HOG

features. We used a cell size of 4x4 pixels, block size of 2x2 cells (8x8 pixels), window size of 32x32 pixels, with 9 orientation bins. Figure 2-6 shows a visualization of the computed HOG features.

Multi-class SVM classifier

To classify the hand shapes from HOG features, we trained a multi-class SVM classifier [113] using LIBSVM [20], with 5 classes (i.e., the four canonical hand poses plus "no hand"). We trained a multi-class SVM with an RBF kernel following the one-against-one method, performing a grid search and 10-fold cross validation for parameter selection.

Training dataset

To train an SVM classifier, a training dataset was collected from the NATOPS dataset, choosing the recorded video clips of the first 10 subjects (out of 20). Positive samples (the four hand poses) were collected by manually selecting 32 x 32 pixel images that contained hands and labeling them; negative samples ("no hand") were collected automatically after collecting positive samples, by choosing two random foreground locations and cropping the same-sized images. We applied affine transformations to the positive samples, to make the classifier more robust to scaling and rotational variations, and to increase and balance the number of samples across hand shape classes. After applying the transformations, the size of each class was balanced at about 12,000 samples.

Clustering

Each time a sliding window moves to a new search region, the HOG features are computed, and the SVM classifier examines them, returning a vector of $k + 1$ probability estimates (k hand classes plus one negative class; $k = 4$ in our current experiments). We thus get multiple classification results per search region, with one from each sliding window

position. To get a single classification result per search region, we cluster all positive classification results (i.e., classified into one of the k positive classes) within the region, averaging positions and probability estimates of the results (see Figure 2-7).

2.3.3 Output Features

From the body and hand tracking described above, we get a 12-dimensional body feature vector and an 8-dimensional hand feature vector. The body feature vector includes 3D joint velocities for left/right elbows and wrists. To obtain this, we first generate a model with the estimated joint angles and fixed-length limbs, so that all generated models have the same set of limb lengths across subjects. This reduces cross-subject variances resulting from different limb lengths. Then we log coordinates of the joints relative to the chest, and take the first order derivatives.

The hand feature includes probability estimates of the four predefined hand poses for left/right hand, dropping the fifth class "no hand" (because as with any set of probabilities that sum to one, N-1 values are enough).

In the next chapter, we describe an algorithm for recognizing actions based on these output features.

Chapter 3

Learning Spatio-Temporal Structure of Human Action

This chapter presents a novel probabilistic graphical model for action recognition that learns spatio-temporal structure in a fine-to-coarse manner. The main idea behind this work is hierarchical sequence summarization: our algorithm constructs a hierarchical representation of a video sequence by iteratively "summarizing" the contents in a bottom-up fashion, and uses it to learn spatio-temporal structure from each of the summary representations. Each layer in the hierarchy is a coarser-grained summary of the sequence from the preceding layer. Intuitively, as the hierarchy builds, we learn ever more abstract spatio-temporal structure of human action.

Technically, the main problem we want to solve is sequence classification: categorize each sequence (not each frame) into one of the known categories. Our idea of learning hierarchical spatio-temporal structure from multiple layers resembles that of *ensemble learning*: we combine multiple prediction results, collected from multiple layers, to obtain better classification performance than could be done from a single layer. It also resembles *deep learning*: our model consists of multiple layers of non-linear operations to obtain ever more abstract feature representations.

(a) Northop Grumman X-47B (b) NATOPS dataset

Figure 3-1: A subset of NATOPS dataset [100]. Body joint trajectories are illustrated in yellow arrows, hand poses are illustrated with synthesized images of hands. Red rectangles indicate hand poses are important in distinguishing the action with its corresponding similar action pair. Action categories shown are: #1 All Clear, #2 Not Clear, #3 Remove Chocks, #4 Insert Chocks, #5 Brakes On, #6 Breaks Off.

3.1 Task and Motivation

Our goal is to build a system that understands human actions recorded in a video format. In particular, we consider the case where the whole video sequence is associated with a single categorical label describing the overall content of the video, and the lengths of sequences can be different from each other. The input to our system is thus a video sequence of varying length, the output is a single categorical label.

Figure 3-1 (b) shows six action categories from the NATOPS dataset used in our experiments. Note that each action instance has a single categorical label, and the length of each action varies (e.g., on average, the "Not Clear" action is 1.8 seconds long, while the "Remove Chocks" action is 2.7 seconds long). Regardless of the sequence length, we want our system to categorize video clips using the NATOPS vocabulary.

3.1.1 Linear-Chain Graphical Models

Our task of assigning a single categorical label to a video sequence can be cast as sequence classification [123]. One popular approach to this problem is to use a linear-chain graphical model, such as Hidden Markov Models (HMMs) [87] and Hidden Conditional Random Fields (HCRFs) [86]. These models assume there exists a set of "states" describing each action, and associate a latent variable with each observation (i.e., each frame), learning state transitions of an action sequence using a set of latent variables. Each latent variable is a discrete multinomial random variable and represents which state an action is in at any given moment. For example, if we train a two-state model with an "arm waving left-to-right" action sequence, one state may correspond to "arm moving left" and the other may correspond to "arm moving right"; once the model is trained, a sequence of latent states will alternate between the two states as an arm waves from left to right.

Linear-chain models are widely popular mainly because of their computational efficiency: an exact inference is feasible in linear-chain models using the forward-backward algorithm [87] or the belief propagation algorithm [84], and the computational complexity is linear in the length of the sequence. One limitation in linear-chain models, however, comes from the first-order Markov assumption that considers only local pairwise dependencies among observations; in other words, it cares about body movement between only two consecutive frames at a time. Therefore long-term dependencies are discarded, limiting its capability of learning spatio-temporal patterns that do not follow the first-order Markov property. Adding a long-term dependency term will solve this problem, but then the computational complexity will grow exponentially with the sequence length.

3.1.2 Hierarchical Spatio-Temporal Structure

We observe that, although the whole action sequence is associated with a single categorical label, an action is often composed of several sub-actions, with each sub-action again decomposable as a series of sub-actions. This recursive decomposition creates a hierarchical

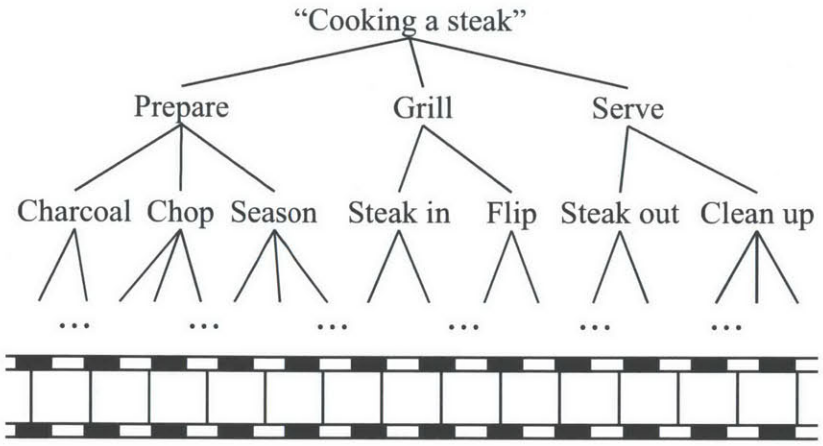


Figure 3-2: Hierarchical decomposition of a fictitious action category "cooking a steak". In many real-world action recognition tasks, we are given a single categorical label describing the overall content of the action sequence; mid-level sub-action labels are typically unknown.

Figure 3-3: A visualization of the "Brakes On" action sequence from the NATOPS dataset [100]. For the purpose of visualization we down-sampled the original sequence by a factor of two.

structure of an action with multiple levels of abstractions. For example, imagine a video showing someone cooking a steak (see Figure 3-2). It may contain several coarse-level sub-actions (e.g., preparation, grilling, and serving) and each sub-action may again contain fine-level sub-actions (e.g., chopping up unions, seasoning the meat, and heating.the grill). The sub-actions at each level represent actions at different spatio-temporal granularities, providing an intuitive description of an action with different levels of abstraction.

The "cooking a steak" example was chosen to illustrate our point because it has a clear hierarchical structure with different levels of abstractions (from coarse to fine). Although they may not be as clear cut as this example, many real-world actions share a similar characteristic that their action sequence can be decomposed into a hierarchy of sub-actions. Take as an example the "Brakes On" signal shown in Figure 3-3. This action is composed

of sub-actions – "arms up", "hands open", and "arms down" – with each sub-action again decomposable as a series of finer-grained body movements.

The goal of work is to exploit this kind of hierarchical structure in human actions. Note that the labels for sub-actions are not known a priori, nor does the system know the corresponding segmentation information, e.g., which sub-sequence of video corresponds to "hands open"? However, we hope to discover them in an unsupervised fashion using latent variables. Remember that latent variables in a linear chain model represent hidden states of an action, which can be interpreted as sub-action labels. An algorithm that builds a hierarchy in a bottom-up fashion, from the original sequence to progressively coarser-grained representation, may reveal the hierarchical structure of an action.

3.2 Hierarchical Sequence Summarization

We learn spatio-temporal structure of human action data using hierarchical sequence summarization. We construct a hierarchical representation of a video sequence by iteratively summarizing the contents in a bottom-up fashion, and use the hierarchical representation to learn spatio-temporal structure from each of the summary representations. Each layer in the hierarchy is a temporally coarser-grained summary of the sequence from the preceding layer. Intuitively, as the hierarchy builds, we learn ever more abstract spatio-temporal structure.

Our approach builds the hierarchical representation by alternating two steps: sequence learning (the L-step) and sequence summarization (the S-step). We start by defining our notation, then describe the sequence learning step and the sequence summarization step. We then formally define our model and explain an efficient optimization procedure.

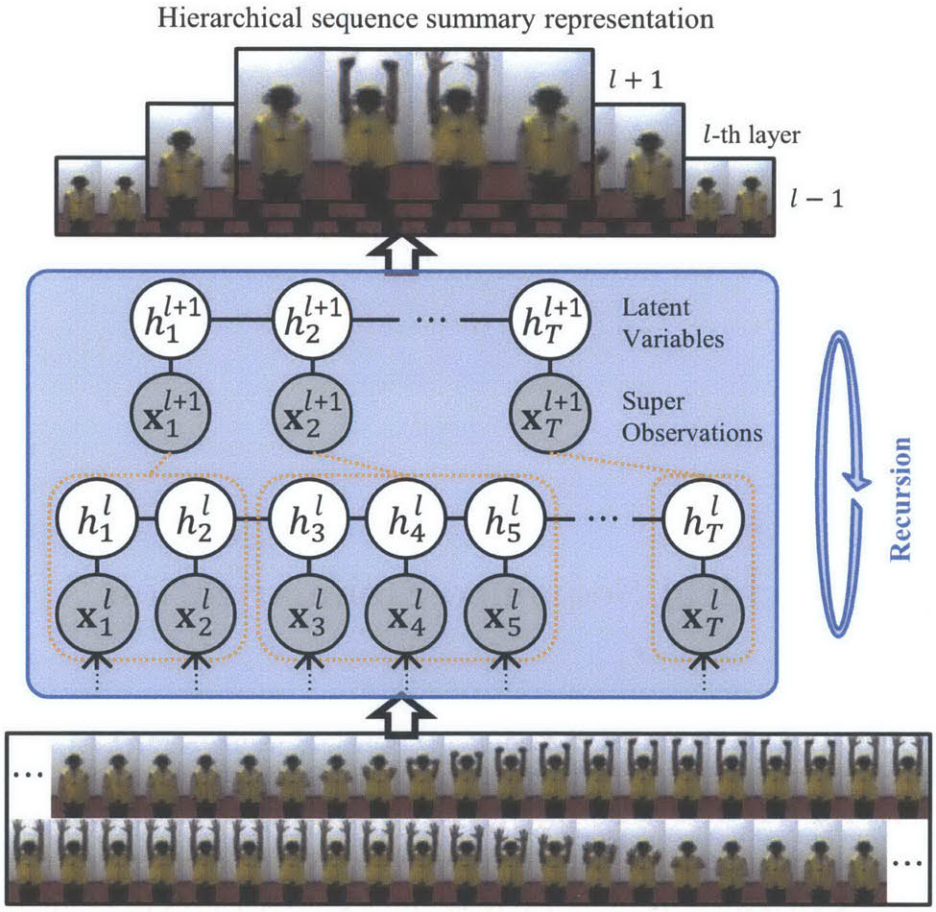


Figure 3-4: We build a hierarchical representation of action sequence and use it to learn spatio-temporal pattern from multiple layers of summary representations.

3.2.1 Notation

Input to our model is a sequence of frames of length T $\mathbf{x} = [\mathbf{x}_1; \cdots ; \mathbf{x}_T]$ (the length can vary across sequences); each observation $\mathbf{x}_t \in \mathbb{R}^D$ is of dimension D and can be any type of action feature (e.g., body pose configuration [86], bag-of-words representation of HOG/HOF [66], etc.). Each sequence is labeled from a finite alphabet set, $y \in \mathcal{Y}$.

We denote a sequence summary at the l-th layer in the hierarchy by $\mathbf{x}^l = [\mathbf{x}_1^l; \cdots ; \mathbf{x}_T^l]$. A *super* observation $\mathbf{x}_t^l$ is a group of observations from the preceding layer, and we define $c(\mathbf{x}_t^l)$ as a reference operator of $\mathbf{x}_t^l$ that returns the group of observations; for $l = 1$ we set $c(\mathbf{x}_t^l) = \mathbf{x}_t$.

Because our model is defined recursively, most procedures at each layer can be formulated without specifying the layer index. In what follows, we omit l whenever it is clear from the context; we also omit it for the original sequence, i.e., l=1.

3.2.2 L-Step: Sequence Learning

We use HCRFs [86] to capture hidden dynamics in each layer in the hierarchy. Using a set of latent variables $\mathbf{h} \in \mathcal{H}$, the conditional probability distribution is defined as

$$p(y|\mathbf{x}; \mathbf{w}) = \frac{1}{Z(\mathbf{x}; \mathbf{w})} \sum_{\mathbf{h}} \exp F(y, \mathbf{h}, \mathbf{x}; \mathbf{w}) \tag{3.1}$$

where $\mathbf{w}$ is a model parameter vector, $F(\cdot)$ is a generic feature function, and $Z(\mathbf{x}; \mathbf{w}) = \sum_{y', \mathbf{h}} \exp F(y', \mathbf{h}, \mathbf{x}; \mathbf{w})$ is a normalization term.

Feature Function: We define the feature function as

$$F(y, \mathbf{h}, \mathbf{x}; \mathbf{w}) = \sum_t f^1(\mathbf{h}, \mathbf{x}, t; \mathbf{w}) + \sum_t f^2(y, \mathbf{h}, t; \mathbf{w}) + \sum_t f^3(y, \mathbf{h}, t, t{+}1; \mathbf{w})$$

Our definition of feature function is different from that of [86] in order to accommodate the

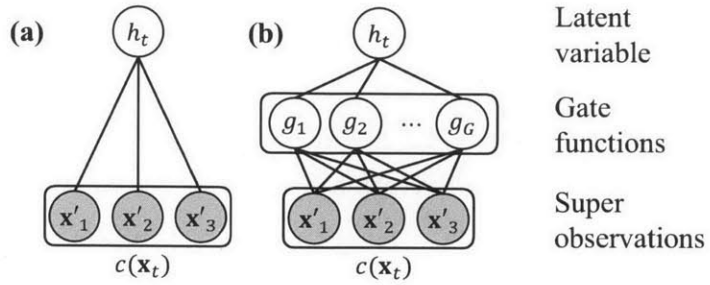


Figure 3-5: **Illustration of our super observation feature function.** (a) Observation feature function similar to Quattoni *et al.* [86], (b) our approach uses an additional set of gate functions to learn an abstract feature representation of super observations.

hierarchical nature of our approach. Specifically, we define the *super observation* feature function that is different from [86].

Let $\mathbb{1}[\cdot]$ be an indicator function, and $y' \in \mathcal{Y}$ and $(h', h'') \in \mathcal{H}$ be the assignments to the label and latent variables, respectively. The second and the third terms in Equation 3.2 are the same as those defined in [86], i.e., the *label* feature function $f^2(\cdot) = w_{y,h}\mathbb{1}[y = y']\mathbb{1}[h_t = h']$ and the *transition* feature function $f^3(\cdot) = w_{y,h,h}\mathbb{1}[y = y']\mathbb{1}[h_t = h']\mathbb{1}[h_{t+1} = h'']$.

Our *super observation* feature function (the first term of Equation 3.2) incorporates a set of non-linear gate functions G, as used in neural networks, to *learn* an abstract feature representation of super observations (see Figure 3-5 (b)). Let $\psi_g(\mathbf{x}, t; \mathbf{w})$ be a function that computes an average of gated output values from each observation contained in a super observation $\mathbf{x}' \in c(\mathbf{x}_t)$,

$$\psi_g(\mathbf{x}, t; \mathbf{w}) = \frac{1}{|c(\mathbf{x}_t)|} \sum_{\mathbf{x}' \in c(\mathbf{x}_t)} g\left(\sum_d w_{g,d} x'_d\right) \tag{3.2}$$

We adopt the logistic function as our gate function, $g(z) = 1/(1 + \exp(-z))$, which has been shown to perform well in the representation learning literature [8]. We define our *super observation* feature function as

$$f^1(\mathbf{h}, \mathbf{x}, t; \mathbf{w}) = \mathbb{1}[h_t = h'] \sum_{g \in G} w_{g,h} \psi_g(\mathbf{x}, t; \mathbf{w}). \tag{3.3}$$

where each $g \in G$ has the same form. The set of gate functions G creates an additional layer between latent variables and observations, and has a similar effect to that of the neural network. That is, this feature function learns an abstract representation of super observations, and thus provides more discriminative information for capturing complex spatio-temporal patterns in human activity data.

To see the effectiveness of the gate functions, consider another definition of the observation feature function, one without the gate functions (see Figure 3-5 (a)),

$$f^1(\mathbf{h}, \mathbf{x}, t; \mathbf{w}) = \frac{1}{|c(\mathbf{x}_t)|} \mathbb{1}[h_t = h'] \sum_{\mathbf{x}'} \sum_{d} w_{h,d} x'_d \tag{3.4}$$

This does not have the automatic feature learning step, and simply represents the feature as an average of the linear combinations of features x'_d and weights $w_{h,d}$. As evidenced by the deep learning literature [8, 68], and consistent with our experimental result, the step of non-linear feature learning leads to a more discriminative representation.

Complexity Analysis: Our model parameter vector is $\mathbf{w} = [w_{g,h}; w_{g,d}; w_{y,h}; w_{y,h,h}]$ and has the dimension of $GH + GD + YH + YHH$, with the number of gate functions G, the number of latent states H, the feature dimension D, and the number of class labels Y. Given a chain-structured sequence $\mathbf{x}$ of length T, we can solve the inference problem at $O(YTH^2)$ using the belief propagation algorithm [84].

3.2.3 S-Step: Sequence Summarization

There are many ways to summarize $\mathbf{x}^l$ to obtain a temporally coarser-grained sequence summary $\mathbf{x}^{l+1}$. One simple approach is to group observations from $\mathbf{x}^l$ at a *fixed* time interval, e.g., collapse every two consecutive observations and obtain a sequence with half the length of $\mathbf{x}^l$. However, as we show in our experiments, this approach may fail to preserve important local information and result in over-grouping and over-smoothing.

We therefore summarize $\mathbf{x}^l$ by grouping observations at an *adaptive* interval, based on

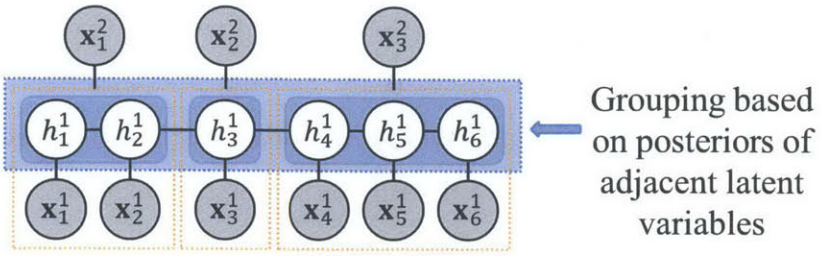


Figure 3-6: **Illustration of sequence summarization.** We generate a sequence summary by grouping neighboring observations that have similar semantic labeling in the latent space.

Algorithm 1: Sequence Summarization Procedure

Input: A weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$
Output: Variable grouping $\mathcal{C} = \{c_1, \cdots, c_T\}$
$\mathcal{C} \leftarrow \mathcal{V}$, $c_t = c(\mathbf{x}_t^{l+1}) = \{\mathbf{x}_t^l\}, \forall t$;
$\mathcal{O} \leftarrow$ sort_ascend$(\mathcal{E}, \mathcal{W})$, $\mathcal{O} = \{o_1, \cdots, o_{T-1}\}$;
for $q = 1 \cdots |\mathcal{O}|$ **do**
  $(s, t) \leftarrow o_q$;
  if $c_s \neq c_t \wedge w_{st} \leq MInt(c_s, c_t)$ **then**
    $\mathcal{C} \leftarrow$ merge(c_s, c_t) ;

how similar the semantic labeling of observations are in the latent space. We work in the latent space because it has learned to maximize class discrimination and thus provides more semantically meaningful information. Said slightly differently, the similarity of latent variables is a measure of the similarity of the corrsponding observations, but in a space more likely to discriminative appropriately.

Sequence summarization can be seen as a variable grouping problem with a piecewise connectivity constraint. We use the well-established graph-based variable grouping algorithm by Felzenszwalb *et al.* [37], with a modification on the similarity metric. The algorithm has the desirable property that it preserves detail in low-variance groups while ignoring detail in high-variance groups, producing a grouping of variables that is globally coherent. The pseudocode of the algorithm is given in Algorithm 1.

The Algorithm: Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ be a weighed graph at the l-th layer, where $\mathcal{V}$ is a

set of nodes (latent variables), $\mathcal{E}$ is a set of edges induced by a linear chain, and $\mathcal{W}$ is a set of edge weights defined as the similarity between two nodes. The algorithm produces a set of super observations $\mathcal{C} = \{c(\mathbf{x}_1^{l+1}), \cdots, c(\mathbf{x}_T^{l+1})\}$.

The algorithm merges $c(\mathbf{x}_s^{l+1})$ and $c(\mathbf{x}_t^{l+1})$ if the difference between the groups is smaller than the *minimum internal difference* within the groups. Let the *internal difference* of a group c be $Int(c) = \max_{(s,t)\in \text{MST}(c,\mathcal{E}_c)} w_{st}$, i.e., the largest weight in the minimum spanning tree of the group c with the corresponding edge set $\mathcal{E}_c$. The *minimum internal difference* between two groups c_s and c_t is defined as $MInt(c_s, c_t) = \min\left(Int(c_s) + \tau(c_s), Int(c_t) + \tau(c_t)\right)$ where $\tau(c_s) = \tau/|c_s|$ is a threshold function; it controls the degree to which the difference between two groups must be greater than their internal differences in order for there to be evidence of a boundary between them.

Similarity Metric: We define the similarity between two nodes (i.e., the weight w_{st}) as

$$w_{st} = \sum_{y,h'} |p(h_s{=}h' \mid y, \mathbf{x}; \mathbf{w}) - p(h_t{=}h' \mid y, \mathbf{x}; \mathbf{w})| \tag{3.5}$$

that is, it is the sum of absolute differences of the posterior probabilities between the two corresponding latent variables, marginalized over the class label. [1]

Complexity Analysis: As shown by Felzenszwalb [37], this sequence summarization algorithm runs quite efficiently in $O(T \log T)$ with the sequence length T.

[1] Other metrics can also be defined in the latent space. We experimented with different weight functions, but the performance difference was not significant. We chose this definition because it performed well across different datasets and is computationally simple.

3.2.4 Model Definition

We formulate our model, Hierarchical Sequence Summarization (HSS), as the conditional probability distribution

$$p(y|\mathbf{x};\mathbf{w}) \propto p(y|\mathbf{x}^1,\cdots,\mathbf{x}^{\mathcal{L}};\mathbf{w}) \propto \prod_{l=1}^{\mathcal{L}} p(y|\mathbf{x}^l;\mathbf{w}^l) \tag{3.6}$$

where $p(y|\mathbf{x}^l;\mathbf{w}^l)$ is obtained using Equation 3.1. Note the layer-specific model parameter vector $\mathbf{w}^l$, $\mathbf{w} = [\mathbf{w}^1;\cdots;\mathbf{w}^{\mathcal{L}}]$.

The first derivation comes from our reformulation of $p(y|\mathbf{x};\mathbf{w})$ using hierarchical sequence summaries, the second comes from the way we construct the sequence summaries. To see this, recall that we obtain a sequence summary $\mathbf{x}^{l+1}$ given the posterior of latent variables $p(\mathbf{h}^l|y,\mathbf{x}^l;\mathbf{w}^l)$, and the posterior is computed based on the parameter vector $\mathbf{w}^l$; this implies that $\mathbf{x}^{l+1}$ is conditionally independent of $\mathbf{x}^l$ given $\mathbf{w}^l$. To make our model tractable, we assume that parameter vectors from different layers $\mathbf{w}^l$ are independent of each other. As a result, we can express the second term as the product of $p(y|\mathbf{x}^l;\mathbf{w}^l)$.

3.2.5 Optimization

Given $\mathcal{D} = \left\{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in \mathbb{R}^{D\times T_i}, y_i \in \mathcal{Y}\right\}_{i=1}^{N}$ as a training dataset, the standard way to find the optimal solution $\mathbf{w}^*$ is to define an objective function as

$$\min_{\mathbf{w}} L(\mathbf{w}) = \mathcal{R}(\mathbf{w}) - \sum_{i=1}^{N} \log p(y_i|\mathbf{x}_i;\mathbf{w}) \tag{3.7}$$

with a regularization term $\mathcal{R}(\mathbf{w}) = \frac{1}{2\sigma^2}\|\mathbf{w}\|^2$, i.e., the log of a Gaussian prior with variance σ^2, $p(\mathbf{w}) \sim \exp(\frac{1}{2\sigma^2}\|\mathbf{w}\|^2)$, then solve it using gradient descent [80].

Unfortunately, because of the hierarchical nature of our approach, the objective function needs to be changed. In our approach only the original sequence $\mathbf{x}^1$ is available at the

outset; to generate a sequence summary $\mathbf{x}^{l+1}$ we need the posterior $p(\mathbf{h}^l|y, \mathbf{x}^l; \mathbf{w}^l)$, and the quality of the posterior relies on an estimate of the solution $\mathbf{w}^l$ obtained so far.

We therefore perform incremental optimization [44], where, at each layer l, we solve for only the necessary part of the solution while fixing all the others, and iterate the optimization process, incrementing l. At each layer l of the incremental optimization, we solve

$$\min_{\mathbf{w}^l} L(\mathbf{w}^l) = \mathcal{R}(\mathbf{w}^l) - \sum_{i=1}^{N} \log p(y_i|\mathbf{x}_i^l; \mathbf{w}^l) \tag{3.8}$$

This layer-specific optimization problems is solved using gradient descent with a standard quasi-newton method, L-BFGS [80], chosen because of its empirical success in the literature [86].

The partial derivative of the second term in Equation 3.8 with respect to the parameter $\mathbf{w}^l$, for a training sample $(\mathbf{x}_i, y_i)$, is computed as

$$\frac{\partial \log p(y_i|\mathbf{x}_i^l; \mathbf{w}^l)}{\partial \mathbf{w}^l} = \sum_{\mathbf{h}^l} p(\mathbf{h}^l|y_i, \mathbf{x}_i^l; \mathbf{w}^l) \frac{\partial F(\cdot)}{\partial \mathbf{w}^l} - \sum_{y', \mathbf{h}^l} p(y', \mathbf{h}^l|\mathbf{x}_i^l; \mathbf{w}^l) \frac{\partial F(\cdot)}{\partial \mathbf{w}^l} \tag{3.9}$$

Specific forms of the partial derivatives $\frac{\partial F(\cdot)}{\partial \mathbf{w}^l}$ with respect to $w_{y,h}^l$ and $w_{y,h,h}^l$ are the same as those in [86], $\frac{\partial f^2(\cdot)}{\partial w_{y,h}^l} = \sum_t \mathbb{1}[y = y']\mathbb{1}[h_t^l = h']$ and $\frac{\partial f^3(\cdot)}{\partial w_{y,h,h}^l} = \sum_t \mathbb{1}[y = y']\mathbb{1}[h_t^l = h']\mathbb{1}[h_{t+1}^l = h'']$. For $w_{g,h}^l$ and $w_{g,d}^l$, they are $\frac{\partial f^1(\cdot)}{\partial w_{g,h}^l} = \sum_t \mathbb{1}[h_t^l = h']\psi_g(\mathbf{x}^l, t; \mathbf{w}^l)$ and $\frac{\partial f^1(\cdot)}{\partial w_{g,d}^l} = \sum_t w_{g,h}^l \frac{1}{|c(\mathbf{x}^l,t)|} \sum_{\mathbf{x}'} g(w_{g,d}^l x_d')(1 - g(w_{g,d}^l x_d'))$, respectively.

Training and Testing: Algorithm 2 and 3 show training and testing procedures, respectively. The training procedure involves, for each l, solving for $\mathbf{w}^{*l}$ and generating a sequence summary $\mathbf{x}^{l+1}$ for each sample in the dataset. The testing procedure involves adding up $\log p(y|\mathbf{x}^l; \mathbf{w}^{*l})$ computed from each layer and finding the optimal sequence label y with the highest probability.

Note that if the summary produces the same sequence (i.e., $\mathbf{x}_i^{l+1}$ is equal to $\mathbf{x}_i^l$), we stop further grouping the sample $\mathbf{x}_i$, both in training and testing procedures. As a result, $\mathbf{x}^{l+1}$

Algorithm 2: Training Procedure

Input: Training dataset $\mathcal{D}$
Output: Optimal solution $\mathbf{w}^*$
for $l = 1 \cdots \mathcal{L}$ **do**
  $\mathbf{w}^{*l} \leftarrow \arg\min_{\mathbf{w}^l} L(\mathbf{w}^l)$; // Equation 3.8
  foreach $\mathbf{x}_i \in \mathcal{D}$ **do**
    $\mathbf{x}_i^{l+1} \leftarrow \text{summarize}(\mathbf{x}_i^l, \mathbf{w}^{*l})$; // Algorithm 1

Algorithm 3: Testing Procedure

Input: Test sequence $\mathbf{x}$, optimal solution $\mathbf{w}^*$
Output: Sequence label y^*
Initialize $p(y|\mathbf{x}; \mathbf{w}^*)$ to zero;
for $l = 1 \cdots \mathcal{L}$ **do**
  $\log p(y|\mathbf{x}; \mathbf{w}^*) \mathrel{+}= \log p(y|\mathbf{x}^l; \mathbf{w}^{*l})$;
  $\mathbf{x}^{l+1} \leftarrow \text{summarize}(\mathbf{x}^l, \mathbf{w}^{*l})$; // Algorithm 1
$y_* \leftarrow \arg\max_y \log p(y|\mathbf{x}; \mathbf{w}^*)$

is always shorter than $\mathbf{x}^l$.

Complexity Analysis: Because of this incremental optimization, the complexity grows only sublinearly with the number of layers considered. To see this, recall that solving an inference problem given a sequence takes $O(YTH^2)$ and the sequence summarization takes $O(T \log T)$. With $\mathcal{L}$ layers considered, the complexity is $O(\mathcal{L}YTH^2 + \mathcal{L}T \log T)$; here, T is a strictly decreasing function of the layer variable (because the length of $\mathbf{x}^{l+1}$ is always shorter than $\mathbf{x}^l$), and thus the complexity of our model increases sublinearly with the number of layers used.

3.3 Experiments

We evaluated the performance of our HSS model on three human activity datasets with different tasks, using different types of input features.

ArmGesture [86]: The task in this dataset is to recognize various arm gestures based on upper body joint configuration. It contains 724 sequences from 6 action categories with an

average of 25 frames per sequence. Each frame is represented as a 20D feature vector: 2D joint angles and 3D coordinates for left/right shoulders and elbows.

Canal9 [114][2]: The task is to recognize agreement and disagreement during a political debate based on nonverbal audio-visual cues. It contains 145 sequences, with an average of 96 frames per sequence. Each frame is represented as a 10D feature vector: 2D prosodic features (F0 and energy) and 8D canonical body gestures, where the presence/absence of 8 gesture categories in each frame was manually annotated with binary values.

NATOPS [100][3]: The task is to recognize aircraft handling signals based on upper body joint configuration and hand shapes. It contains 2,400 sequences from 6 action categories, with an average of 44 frames per sequence. Each frame is represented as a 20D feature vector: 3D joint velocities for left/right elbows and wrists, and the probability estimates of four canonical hand gestures for each hand, encoded as 8D feature vector.

3.3.1 Methodology

We followed experimental protocols used in published work on each dataset: For the ArmGesture and Canal9 datasets we performed 5-fold cross-validation, for the NATOPS datasets we performed hold-out testing, using the samples from the first 5 subjects for testing, the second 5 subjects for validation, with the rest for training.

We varied the number of latent states $H \in \{4, 8, 12\}$ and the number of gate functions $G \in \{4, 8, 12\}$, and set the number of layers $\mathcal{L} = 4$; for simplicity we set H and G to be the same across layers. The threshold constant in sequence summarization was varied $\tau \in \{0.1, 0.5, 1.0\}$ (see Algorithm 1). The L_2 regularization scale term σ was varied $\sigma = \{10^k | k \in \{1, 2, 3\}\}$.

[2]The original dataset [114] contains over 43 hours of recording; to facilitate comparison with previous results we used the subset of the dataset described in [17].

[3]The original dataset [100] contains 9,600 sequences from 24 action categories; we used the subset of the dataset used in [102].

Model	Mean Accuracy
HMM (from [86])	84.83%
CRF (from [86])	86.03%
MM-HCRF (from [102])	93.79%
Quattoni *et al.* [86]	93.81%
Shyr *et al.* [96]	95.30%
Song *et al.* [102]	97.65%
HCNF	97.79%
Our HSS Model	**99.59%**

Table 3.1: Experimental results from the ArmGesture dataset.

Model	Mean Accuracy
SVM (from [17])	51.89%
HMM (from [17])	52.29%
Bousmalis *et al.* [17]	64.22%
Song *et al.* [103]	71.99%
HCNF	73.35%
Our HSS Model	**75.56%**

Table 3.2: Experimental results from the Canal9 dataset.

Since the objective function (Equation 3.8) is non-convex, we trained each model twice with different random initializations. The optimal configuration of all the hyper-parameters we used were chosen based on the highest classification accuracy on the validation dataset.

3.3.2 Results

Table 3.1 and Table 3.2 shows experimental results on the ArmGesture and Canal9 datasets, respectively. For each dataset we include previous results reported in the literature;, as well as the result obtained by us using Conditional Neural Fields [85] with latent variables (HCNF). As can be seen, our approach outperforms all the state-of-the-art results on the ArmGesture and Canal9 datasets. Notably, our approach achieves a near-perfect accuracy on the ArmGesture dataset (99.59%).

For the NATOPS dataset, the state-of-the-art result is 87.00% in our ealrier work [102].

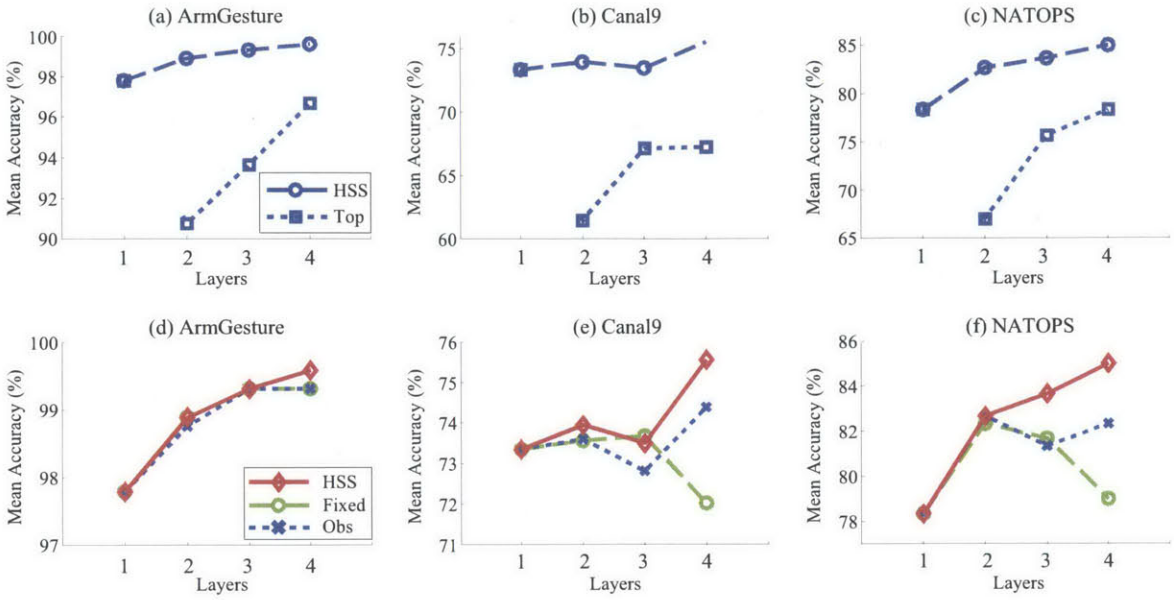


Figure 3-7: **Detailed analysis results**. The top row (a)-(c) shows experimental results comparing hierarchical (HSS) and single optimal (top) representation approaches, the bottom row (d)-(f) shows the results on three different sequence summarization approaches.

Our earlier approach used a multi-view HCRF to jointly learn view-shared and view-specific hidden dynamics, where the two views are defined as upper body joint configuration and hand shape information. Even without considering the multi-view nature of the dataset (we perform an early-fusion of the two views), our approach achieved a comparable accuracy of 85.00%. This is still a significant improvement over various previous results using an early-fusion: HMM (from [102], 76.67%), HCRF (from [102], 76.00%), and HCNF (78.33%).

For detailed analysis we evaluated whether our hierarchical representation is indeed advantageous over a single representation, and how our sequence summarization in the latent space differs from the other approaches.

1) Hierarchical vs. single optimal representation: While the performances shown above show significant improvements over previous sequence learning models, they do not prove the advantage of learning from hierarchical sequence summary representation, as opposed to learning from only the optimal layer inside the hierarchy (if any). To this end,

we compared our approach to the single (top) layer approach by computing during the testing procedure $p(y|\mathbf{x};\mathbf{w}) = p(y|\mathbf{x}^{\mathcal{L}};\mathbf{w}^{\mathcal{L}})$, varying $\mathcal{L} = \{2,3,4\}$; the training procedure was the same as Algorithm 2 (otherwise the obtained sequence summary is not optimal).

Figures 3-7 (a)-(c) show the mean classification accuracy as a function of $\mathcal{L}$, the number of layers, on all three datasets. Our "HSS" approach always outperformed the "Top" approach. Paired t-tests showed that the differences were statistically significant in all three datasets (p <.001). This shows that there is no single representation that is as discriminative as the hierarchical representation.

2) Different sequence summarization algorithms: Our sequence summarization produces groups of temporally neighboring observations that have similar semantic meaning in the latent space. We compare this to two different approaches: One approach simply collapses every l consecutive observations and obtain a sequence of length T/l at each layer l ("Fixed" in Figure 3-7). Another approach produces groups of observations that are similar in the feature space, with a similarity metric defined as $w_{st} = |\mathbf{x}_s - \mathbf{x}_t|$ and with the threshold range $\tau = \{1,5,10\}$ ("Obs" in Figure 3-7).

As can be seen in Figures 3-7 (d)-(f), our approach outperforms the two other approaches on the Canal9 and NATOPS datasets; on the ArmGesture dataset, performance saturates towards near perfect accuracy. The Fixed approach collapses observations as long as there is more than one, even if they contain discriminative information individually, which may cause over-grouping. Our result supports this hypothesis, showing that the performance started to decrease after $\mathcal{L} > 3$ on the Canal9 and NATOPS datasets.

The Obs approach groups observations using input features, not the corresponding posteriors $p(\mathbf{h}|y,\mathbf{x};\mathbf{w})$ in the latent space. There are a number of difficulties when dealing with input features directly, e.g., different scales, range of values, etc, which makes the approach sensitive to the selected feature space. Our approach, on the other hand, uses latent variables that are defined in the scale [0:1] and contains discriminative information learned via mathematical optimization. We can therefore expect that, as can be seen in

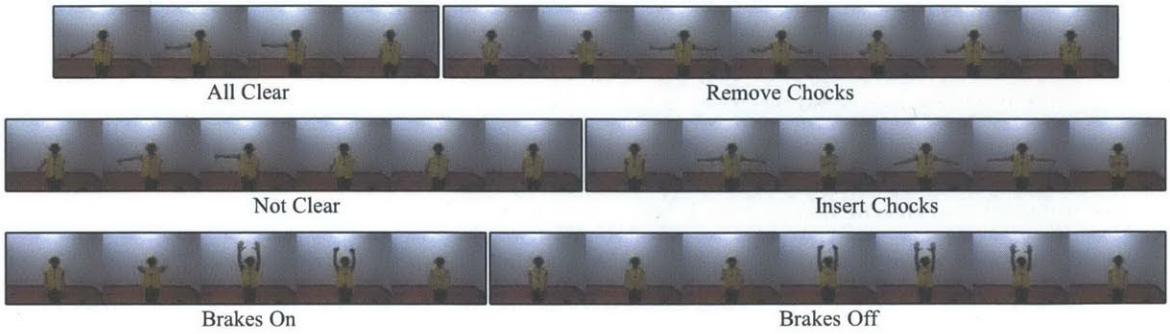


Figure 3-8: **Inferred sequence summaries on the NATOPS dataset**. Each super observation represents key transitions of each action class. For the purpose of visualization we selected the middle frame from each super observation at the 4-th layer.

our results, our approach is more robust to the selection of the scale/range as well as the threshold parameter τ, resulting in overall better performance.

3.4 Related Work

Learning from a hierarchical feature representation has been a recurring theme in action recognition [79, 109, 61, 117, 68]. One approach detects spatio-temporal interest points (STIP) [65] at local video volumes, constructs a bag-of-words representation of HOG/HOF features extracted around STIPs, and learns an SVM classifier to categorize actions [66]. This has been used to construct a hierarchical feature representation that is more discriminative and context-rich than "flat" representations [109, 117, 61]. Sun *et al.* [109] defined three levels of context hierarchy with SIFT-based trajectories, while Wang *et al.* [117] learned interactions within local contexts at multiple spatio-temporal scales. Kovashaka and Grauman [61] proposed to learn class conditional visual words by grouping local features of motion and appearance at multiple space-time scales. While these approaches showed significant improvements over the local feature representation, they use non-temporal machine learning algorithms to classify actions (e.g., SVM and MKL), limiting their application to real-world scenarios that exhibit complex temporal

structures [86, 111].

Sequence learning has been a well-studied topic in machine learning (e.g., HMM and CRF), and has been used successfully in action recognition [86, 119, 111]. Quattoni *et al.* [86] incorporated latent variables into CRF (HCRF) to learn hidden spatio-temporal dynamics, while Wang *et al.* [119] applied the max-margin learning criterion to train HCRFs. While simple and computationally efficient, the performance of HCRFs has been shown to decrease when the data has complex input-output relationships [85, 128]. To overcome this limitation, Peng et al. [85] presented Conditional Neural Fields (CNF) that used gate functions to extract nonlinear feature representations. However, these approaches are defined over a single representation and thus cannot benefit from the additional information that hierarchical representation provides.

Our model has many similarities to the deep learning paradigm [8], such as learning from multiple hidden layers with non-linear operations. Deep belief networks (DBN) [45] have been shown to outperform other "shallow" models in tasks such as digit recognition [45], object recognition [89], and face recognition [48]. Recently, Le *et al.* [68] applied an extension of Independent Subspace Analysis with DBN to action recognition. However, obtaining an efficient learning algorithm that is scalable with the number of layers still remains a challenge [45, 89]. Compared to DBN, the learning complexity of our method grows sublinearly with the size of the hierarchy.

Previous approaches to learning with multiple representations using HCRF (e.g., [128]) define each layer as a combination of the original observation and the preceding layer's posteriors, at the *same* temporal resolution. Our work learns each layer at temporally coarser-grained resolutions, making our model capable of learning ever-more high-level concepts that incorporate the surrounding context (e.g., what comes before/after).

While our approach has shown to work well in action recognition, specifically with single modality (visual), experimental results show that there is still room for improvement in several directions. Our model takes the early-fusion approach for data fusion – feature

vectors from different modalities are concatenated to produce one large feature vector – and thus discards structures *among* the modalities, e.g., correlation and interaction. One direction of improvement is therefore multimodal structure learning, described below.

Chapter 4

From Unimodal to Multimodal: Understanding Human Behaviors

We shift our attention from unimodal (visual) to multimodal aspects of video content analysis, focusing on the task of understanding natural human behaviors recorded in video data. Human behavior is inherently multimodal: we express our intents and thoughts via speech, gesture, and facial expressions. Information on human behavior is conveyed in the form of audio (both verbal and nonverbal), visual (face and body expressions), and text (speech transcript). To build a system that understands natural human behaviors from video data, we need algorithms able to sense, learn, and infer from multiple modalities.

This chapter describes the problem of personality recognition based on multimodal human behavior. Based on a well-developed theory in psychology called the Big Five personality traits [25, 71], we collected a dataset for personality recognition from YouTube, which we call the Time10Q dataset. The goal is to predict *people's impression* about the personality of someone appearing in a short video clip; in other words, instead of predicting the true personality of someone in video, we focus on predicting how people would perceive the personality of someone in video. We call this task the *personality impression recognition.*

Below, after briefly reviewing the study of personality from the psychology perspective,

we introduce the task of automatic personality impression recognition, describe the data collection procedure, and explain how we extract features from multiple modalities.

4.1 Personality Understanding

People reason about the personality of others virtually everyday, both intentionally and unintentionally. After watching someone's behavior even briefly, we can judge whether someone is friendly or unfriendly, extrovert or introvert, etc. – we are all an expert in personality assessment.

A tremendous amount of research has been conducted on the effect of someone's personality in a variety of social contexts [51, 126, 34]. When making friends we try to see what kind of personality others have in order to determine if they get along well [51]. In some fashion a job interview is very much like the personality test [126]; a study shows that one's personality strongly affects their job performance [6]. Making the right persona is an extremely important factor in political campaign [34]. All this evidence suggests that our perception of others' personalities have a tremendous effect on our decisions and thoughts.

4.1.1 Human Personality Psychology

The study of human personality dates as far back as the ancient Greek philosophy, where the early philosophers such as Plato and Aristotle set down fundamental insights into the human psyche [36]. The field of personality psychology has evolved into a number of different schools and models over the past centuries, which can be largely categorized into two contradictory views: *idiographic* and *nomothetic* [32]. The former asserts that human personality is unique in its own right and no two are exactly alike, while the latter believes that there exists a set of common traits and that a personality can be described as a combination of quantifiable traits [19, 35].

Big Five Traits	Correlated trait adjectives
Openness (vs. closedness to experience)	Curious, imaginative, artistic, wide interests, excitable, unconventional
Conscientiousness (vs. lack of direction)	Efficient, organized, not careless, thorough, not lazy, not impulsive
Extraversion (vs. introversion)	Sociable, forceful, energetic, adventurous, enthusiastic, outgoing
Agreeableness (vs. antagonism)	Forgiving, not demanding, warm, not stubborn, not show-off, sympathetic
Neuroticism (vs. emotional stability)	Tense, irritable, not contented, shy, moody, not self-confident

Table 4.1: Correlated adjectives to the Big Five traits.

In this work, we take the nomothetic view of human personality that many contemporary psychologists believe to be appropriate [32], without trying to suggest which view is more correct.

4.1.2 The Big Five Personality Traits

Many contemporary psychologists agree that there exists five basic dimensions of human personality, called the Big Five personality traits [25, 71], which include openness (O), conscientiousness (C), extraversion (E), agreeableness (A), and neuroticism (N) — *OCEAN* for short (see Table 4.1). These five dimensions *"have been found to contain and subsume most known personality traits and are assumed to represent the basic structure behind all personality traits"* [81].

Several approaches have been developed to measure the Big Five traits. The most widely used ones typically have a questionnaire format with a number of self-descriptive questions [25, 40]. The consensus among the psychologists is that a format with more questions leads to a more accurate measurement, e.g., the NEO Personality Inventory Revised version (NEO-PI-R) [25] has 240 questions. But in certain circumstances, asking many questions may not be realistic, such as in Internet-based psychology studies [40], where

	Question
Q1	The person is reserved
Q2	The person is generally trusting
Q3	The person tends to be lazy
Q4	The person is relaxed, handles stress well
Q5	The person has few artistic interests
Q6	The person is outgoing, sociable
Q7	The person tends to find fault with others
Q8	The person does a thorough job
Q9	The person gets nervous easily
Q10	The person has an active imagination

Table 4.2: The Big Five Inventory 10-item version (BFI-10) by Rammstedt and John [88]. Each question is rated on a five-point Likert scale: 1 (disagree strongly), 2 (disagree a little), 3 (neither agree nor disagree), 4 (agree a little), 5 (agree strongly). The five traits are then computed as: Openness = Q10 - Q5, Conscientiousness = Q8 - Q3, Extraversion = Q6 - Q1, Agreeableness = Q2 - Q7, Neuroticism = Q9 - Q4.

a shorter version is preferred. The Big Five Inventory 10-item version (BFI-10) [88], for instance, has only 10 questions and takes less than a minute to finish (see Table 4.2).

Because our work takes a crowd-sourced approach to collecting personality impression measurements (see below), we decided to adopt the BFI-10 questionnaire by Rammstedt and John [88] to measure the Big Five personality traits. Table 4.2 lists the 10 questions, where each question is rated on a five-point Likert scale ranging from 1 (disagree strongly) to 5 (agree strongly). The scores for personality traits are computed from the ratings as: Openness = Q10 - Q5, Conscientiousness = Q8 - Q3, Extraversion = Q6 - Q1, Agreeableness = Q2 - Q7, Neuroticism = Q9 - Q4.

4.1.3 Automatic Recognition of Personality Impression

Why are we interested in automatic recognition of personality impression? One reason is task automation: psychologists analyze hours of videos to study people's behaviors; with an increasing amount of data this becomes quickly prohibitive, making an automated system attractive. A second reason is ensuring the consistency of assessment: it is a well-

known phenomenon that the mood of an investigator affects the results of psychological tests [122], which may cause a serious flaw in experimental analysis. One advantage of an automated personality analyzer is consistency – once it is programmed, it behaves in the same way under the same condition. Finally, having an automated system for personality understanding will improve natural human-computer interaction: studies show that people find it more natural if a virtual agent displays some degree of empathy [56].

To build such a system, we need an appropriate dataset that we can train and evaluate our algorithms on. Below we describe a new dataset we collected from the YouTube website.

4.2 Time10Q Dataset

Time Magazine has a series of video episodes called *Time 10 Questions*. In each episode, a reporter from Time interviews with a public figure, asking a number of questions collected from the subscribers. The episodes are available on both Time Magazine's website and their Youtube channel.

Several factors make the dataset particularly interesting for our purpose. First, it contains high quality audio-visual data. The interviews are held at an indoor environment with controlled lighting and a little to no background noise. The videos are recorded by a team of professional camera men, audio engineers, and lighting technicians; they capture the interviewee's important moments with various postures and gestures, with up-close shots of the face. This makes it possible to extract a variety of high quality multimodal signals.

Second, the data contains natural conversations with spontaneous expressions. Unlike other datasets for personality impression recognition that focused on self-speaking scenarios (e.g., web blog videos [76, 9]), the speakers are having natural conversations in an informal setting. This makes verbal and non-verbal expressions much more natural. Moreover, most of the interviewees are professionals who are quite familiar with expressing themselves in front of a camera; so we can expect that their expressions are not posed nor planned.

Finally, the public figures in the dataset have the variety of job occupations among interviewees, from actors and musicians to politicians and academics (see Table 4.4). This allows the dataset to contain a wide spectrum of personalities.

4.2.1 Data Collection and Segmentation

We downloaded 149 episodes from Time's YouTube channel.[1] Each file is encoded using H.264 YUV420p video codec and AAC 44,100 Hz stereo fltp audio codec, mux-ed into an MP4 file format. Due to the difference in the production date of each episode, the video files have slight variations in resolution (varies among 640x360, 480x360, and 480x270) and FPS (varies between 24 and 30). Table 4.3 shows a list of interviewees included in our dataset, Table 4.4 shows a list of job occupation frequencies.

Each episode starts with a short introduction of the interviewee narrated by a Time reporter, followed by a series of questions and answers. The interviews are in an informal setting, and the interview formats varies across episodes.

Our focus is on the interviewees, not the interviewers. However, the video recorded both the interviewee and interviewer. Also, most episodes include B-roll footages, i.e., supplementary footage added to provide additional information about the interviewee. Although effective for the content delivery purpose, these are actually troublesome for our purpose: B-rolls typically include photos or videos that do not include the interviewee's behavior, or (worse yet) the interviewee acting a particular persona in a public setting, e.g., an actor playing a role in a movie. Both the footage from interviewer and the B-rolls can distract and bias the assessment of the interviewee's personality, especially when the interviewee's behavior during the B-Rolls is radically different from the one in the interview.

To minimize unwanted influence from the footage of interviewer and B-rolls, we manually segmented out only the answer part of each Q&A pair. We further discarded segments if (a) they were too short (less than 10 seconds) or (b) more than one third of the entire

[1] `http://www.youtube.com/user/TimeMagazine`

No	Name	No	Name	No	Name
1	A. R. Rahman (7)	51	Hugh Jackman (6)	101	Pete Sampras (5)
2	Al Roker (10)	52	Ian Bremmer (4)	102	Peter Goodwin (3)
3	Alan Mulally (8)	53	Imran Khan (5)	103	Phil Jackson (8)
4	Alex Trebek (8)	54	J. J. Abrams (7)	104	Queen Latifah (8)
5	Alicia Keys (5)	55	Jamaica Kincaid (5)	105	Questlove (8)
6	America Ferrera (7)	56	James Cameron (5)	106	Rachael Ray (5)
7	Amy Poehler (6)	57	Jane Goodall (6)	107	Randy Pausch (5)
8	Andre Gassi (5)	58	Janet Evanovich (6)	108	Ray Kurzweil (7)
9	Annie Leibovitz (5)	59	Jason Reitman (9)	109	Reese Witherspoon (5)
10	Anthony Bourdain (5)	60	Javier Bardem (8)	110	Richard Brandon (5)
11	Aretha Franklin (7)	61	Jeff Bridges (5)	111	Rick Warren (4)
12	Arianna Huffington (5)	62	Jeremy Piven (6)	112	Ricky Gervais (3)
13	Ashton Kutcher (6)	63	Jim Cramer (8)	113	Robert Caro (5)
14	Aziz Ansari (4)	64	Jimmy Wales (8)	114	Robert Groves (6)
15	Bill Gates (4)	65	Jody Williams (6)	115	Robert Kiyosaki (9)
16	Bill Keller (5)	66	John Ashbery (6)	116	Robert Redford (4)
17	Bill O'Reilly (9)	67	John Krasinski (7)	117	Robin Williams (5)
18	Bode Miller (5)	68	John Mellencamp (6)	118	Roger Goodell (7)
19	Brad Anderson (4)	69	John Woo (9)	119	Ron Paul (7)
20	Caitlin Moran (6)	70	Jose Antonio Vargas (1)	120	Salman Rushdie (5)
21	Candace Parker (7)	71	Joshua Bell (7)	121	Sarah Silverman (4)
22	Carrie Fisher (4)	72	Joss Whedon (9)	122	Shakira (4)
23	Chaz Bono (7)	73	Julieta Garibay (1)	123	Shaun White (5)
24	Chris Kyle (6)	74	Julio Salgado (1)	124	Sherman Alexie (10)
25	Chris Rock (7)	75	Kofi Annan (8)	125	Shimon Peres (6)
26	Colin Powell (7)	76	Lang Lang (3)	126	Simon Pegg (6)
27	Dalai Lama (6)	77	Larry King (8)	127	Smokey Robinson (7)
28	Daniel Kahneman (8)	78	Li Na (8)	128	Stephenie Meyer (5)
29	Daniel Radcliffe (9)	79	Louis C.K. (6)	129	Steven Spielberg (5)
30	Danny Boyle (6)	80	Madeleine Albright (8)	130	Sting (5)
31	Darren Aronofsky (4)	81	Magic Johnson (9)	131	Susan Rice (5)
32	Dave Grohl (10)	82	Mandeep Chahal (1)	132	Susan Sarandon (5)
33	David Adjaye (7)	83	Maya Angelou (6)	133	Suze Orman (7)
34	David Brooks (3)	84	Maya Rudolph (3)	134	Taylor Swift (8)
35	David McCullough (6)	85	Michael Chabon (6)	135	Ted Williams (5)
36	David Stern (5)	86	Michael J. Fox (7)	136	Tolu Olobumni (1)
37	Denis Leary (5)	87	Michael Vick (4)	137	Tom Friedman (5)
38	Diane Sawyer (4)	88	Michelle Williams (4)	138	Tom Wolfe (4)
39	Donatella Versace (4)	89	Mickey Rourke (7)	139	Toni Morrison (5)
40	Doug Ulman (7)	90	Mike Tyson (9)	140	Tony Hawk (7)
41	Emma Watson (8)	91	Mitt Romney (7)	141	Ty Burrell (4)
42	Ewan McGregor (7)	92	Muhammad Yunus (6)	142	Van Morrison (6)
43	Gaby Pacheco (1)	93	Nancy Pelosi (5)	143	Victor Palafox (1)
44	Gavin Newsom (8)	94	Nassim Taleb (5)	144	Viggo Mortensen (6)
45	Gordon Brown (5)	95	Natalie Maines (5)	145	Werner Herzog (5)
46	Harrison Ford (7)	96	Natalie Portman (3)	146	Wladimir Klitschko (4)
47	Helen Mirren (5)	97	Neil deGrasse Tyson (6)	147	Woody Allen (9)
48	Henry Paulson (7)	98	Nigella Lawson (6)	148	Zac Efron (6)
49	Hilary Swank (6)	99	Paul Farmer (3)	149	will.i.am (7)
50	Hugh Hefner (8)	100	Perez Hilton (8)		

Table 4.3: A list of 149 interviewees in the Time10Q dataset. We segment each episode (numbers in parenthesis indicate the number of video segments produced from each episode); the total number of segments is 866.

Job Occupation	Count	Job Occupation	Count	Job Occupation	Count
Actor/Actress	25	Comedian	9	Commissioner	2
Musician	16	Civilian	8	Coach	1
Book Writer	15	TV Presenter	7	Photographer	1
Athletic	12	Journalist	7	Architect	1
Politician	11	Entrepreneur	7	Sniper	1
Scholar	11	Chef	2		
Film Director	10	Religious Leader	2		

Table 4.4: A list of job occupations in the Time10Q dataset.

segment contains B-roll material. This resulted in a total of 866 video segments (9 hours 22 minutes in total). Table 4.3 shows the number of video segments we produced from each episode.

The segmentation effectively gives us the "thin slices", a term in psychology that refers to an effective length of movie clips for psychological studies. Ambady *et al.* [2] have found that the results of clinical and social psychological studies did not differ whether the subjects have watched someone's behavior from a short (less than 30-second) or a long (4- and 5-minute) movie clips. The average duration of each episode is 6 minutes, while the average duration of each segment is 38 seconds, with about 90% less than a minute long. Figure 4-1 shows the histogram of segment durations.

4.3 Multimodal Feature Extraction

We next describe in detail how we extract features from each of the modalities – face, body, and speech (both verbal and non-verbal).

4.3.1 Features from Facial Expression

There is a large body of literature on how informative the human face is in revealing one's emotion and personality [30, 31]. Because our video data contains an upper body shot

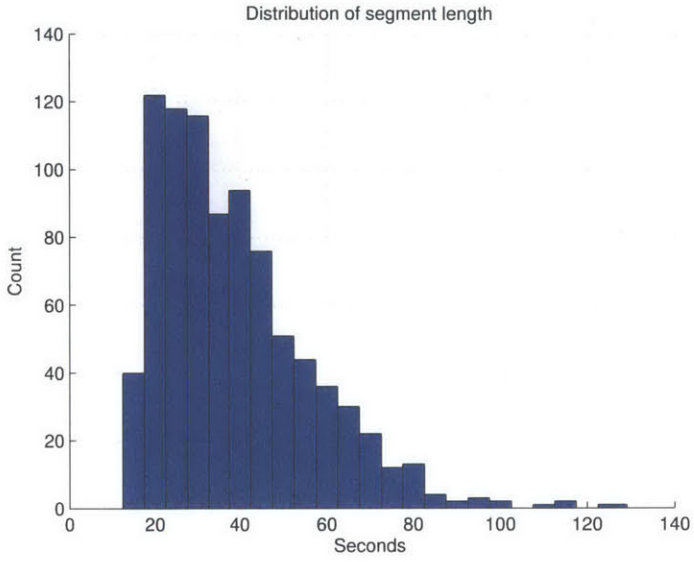
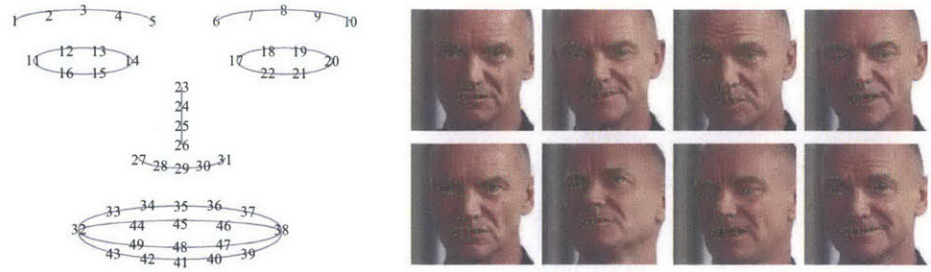


Figure 4-1: A histogram of video segment durations. Our dataset contains 866 video segments. About 90% of the segments are less than a minute long, the average duration is 38 seconds.

of the interviewees for a vast majority of time, we can track the face region and extract appearance features in that region.

For each frame of video, we obtain 49 facial landmarks (see Figure 4-2 (a)) using the IntraFace software package developed by Xiong and De La Torre [124], which uses a supervised descent method to minimize a nonlinear least squares function. We then perform affine transformation on the bounded face patch and normalize its size to 120x120 pixels so that the centers of the eyes are in the same location in all the patches. The resulting face patch is converted to gray scale. From the normalized images, we extract the Pyramid version of Histogram of Oriented Gradients (PHOG) features [14] with eight bins on three different pyramid levels. This results in a 680-dimensional feature vector.



(a) 49 Landmarks

(b) Example results

Figure 4-2: (a) 49 landmarks tracked by the IntraFace [124]. (b) example results on the Time10Q dataset.

4.3.2 Features from Body Motion

Human body motion provides vital information on non-verbal communication, called Kinesics [10]. In addition to the features from facial expression, we extract local visual features to describe body motion. Local feature descriptors have shown to perform well in challenging scenarios in action recognition [1], such as detecting activities from Hollywood movie clips [66]. Part of our contribution is evaluating local feature descriptors in the personality recognition setting.

Note that in the previous chapter we tracked skeleton upper body postures and used them to recognize a predefined vocabulary of body actions. That approach has the advantage that it provides a compact way of representing body motion. But the scenario presented in this chapter makes it difficult to apply the same technique. The video data has the "in the wild" characteristic in the sense that the camera viewpoint changes frequently, and some body parts are often occluded or not recorded, making it harder to estimate skeleton body postures.

We extract local features from densely sampled interest points using the Dense Trajectory software package developed by Wang *et al.* [116]. It samples dense points from each video frame and tracks them based on displacement information from a dense optical flow field. It then extracts three different feature descriptors: Histogram of Oriented Gradients

(HOG), Histogram of Optical Flow (HOF), and Motion Boundary Histograms (MBH). We sample the dense trajectories at a 10-pixel stride, and track each trajectory for 15 frames. Descriptors are computed at 32x32 pixel region around each trajectory. The dimension of each local feature vector is 396 (96 for HOG, 108 for HOF, and 192 for MBH).

4.3.3 Features from Utterances

Utterances have shown to provide important information about human personality [70]. Note that the utterance has much coarser granularity than other modalities (e.g., we can speak only a few words in a second) and that the surrounding context is crucial (e.g., a word can have a completely opposite meaning given different surrounding words). Therefore, we extract the utterance features from the words spoken in an entire video segment (about 38 seconds to 1.5 minutes).

Most episodes on the Time 10 Questions series uploaded on the YouTube website come with automatically recognized verbal transcripts. We downloaded the transcripts, removed common English stopwords[2], and segmented transcripts from each episode to match with the corresponding video segment. The transcripts are then processed using Latent Dirichlet Allocation (LDA) [11] with 50 latent topic classes to extract 50-dimensional feature vector, using the Matlab Topic Modeling Toolbox [42].

4.3.4 Features from Prosody of Speech

The term prosody refers to the rhythm, stress, and intonation of speech. The role of prosody in personality understanding has been studied extensively [90, 73]. In this work we extract the fundamental frequency (F0), the Noise-to-Harmonic Ratio (NHR), and the loudness contour from the audio channel using the OpenSmile software package [33], with a window of size 50ms at 10ms interval. F0 represents the lowest frequency produced

[2]Downloaded from `http://www.textfixer.com/resources/common-english-words.txt`

by air flowing through the vocal folds, the NHR represents the voice quality, and the loudness contour is the sequence of short-term loudness values extracted on a frame-by-frame basis [92]. This results in a 3-dimensional feature vector. For more details in prosody features, see Schuller [92].

4.4 Crowd-Sourced Personality Impression Measurement

For personality label data collection, we use untrained crowd workers on Amazon's Mechanical Turk. Psychology experts and trained labelers might provide higher quality and more reliable data [41], but these solutions are neither cost-effective nor scalable [59]. A major advantage in crowdsourcing data collection is scalability: our Mechanical Turk study presents a scalable method to collect large-scale data on personality impression. Moreover, crowdsourcing provides additional data on how people's personality impression agrees or disagrees with others. A drawback is individual variances and quality control, which we address with several techniques presented below.

4.4.1 Task Setup

On Mechanical Turk, a task requester (researcher) recruits workers (Turkers) by paying money to solve Human Intelligence Tasks (HITs). The task design and configuration parameters determine the quality and speed of data collection. Here we present major task parameters used in our HIT.

- Upon completion of the HIT, a Turker was paid $0.10. Note that a Turker can complete multiple HITs but was not permitted to do the same HIT multiple times.
- Instead of asking a Turker to watch an entire episode of an interview , we decompose a clip into the 38 second - 1.5 minute video segments mentioned above. Assigning a

short, separate HIT for each short video segment makes the task more attractive to Turkers.

- We ask 10 Turkers to complete the task for each video segment. This reduces the chance of a spammer's data tainting the data quality [41]. As a routine practice, we compute inter-rater reliability to see how much the Turkers agree on their answers, shown below.

- As a majority of the interviewees in the video set were well-known in America and represent the American culture, we recruited workers who reside in the U.S. to control for cultural biases. Additionally, we only accepted workers with 95% or higher approval rating for quality control.

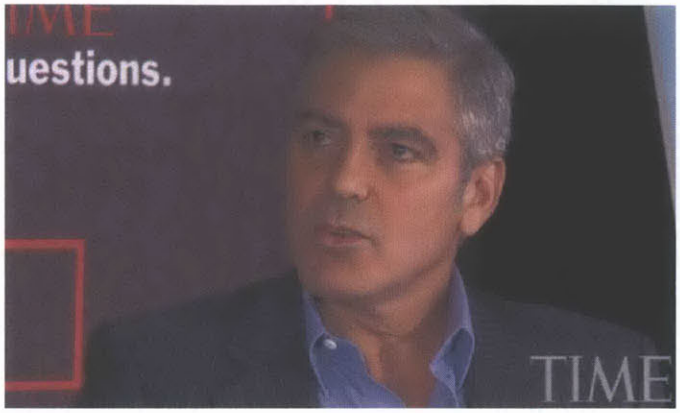
4.4.2 Task Implementation

Mechanical Turk allows tasks created in their built-in interface or linked to an external website. We hosted an external task website to have more control in the task design. The HIT was implemented using YouTube's API and HTML/CSS/Javascript. We used the MTurk command line tool [3] to create, manage, and review results.

4.4.3 Task Walk-through

The HIT is designed to collect a Turker's responses to the BFI-10 questionnaire after watching a short video clip of an interviewee answering a question. We designed the HIT to maximize the quality of the answers by using a variety of techniques, including not revealing the questionnaire until the video is done, not enabling watching fast forward, etc (see below). Figure 4-5 shows the overall task.

[3] `http://aws.amazon.com/developertools/Amazon-Mechanical-Turk/694`



The 10 questions:
How well do the following statements describe the person in the video?

1. The person is reserved
2. The person is generally trusting
3. The person tends to be lazy
4. The person is relaxed, handles stress well
5. The person has few artistic interests
6. The person is outgoing, sociable
7. The person tends to find fault with others
8. The person does a thorough job
9. The person gets nervous easily
10. The person has an active imagination

Figure 4-3: A screen shot of the tutorial HIT (page 1 of 2).

Tutorial HIT

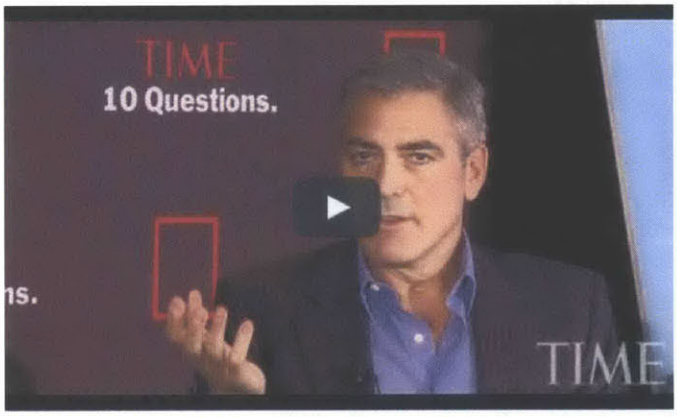
To help the Turker get familiar with the personality questions, the HIT presents a short tutorial with a dummy video clip (not used in the actual study) and the BFI-10 questionnaire (Figure 4-3). We keep track of all Turkers who completed the tutorial, and show the tutorial to each of them only once.

Main HIT

The main HIT starts by giving a short description of the task, along with a notification saying that "This HIT expires in 7 minutes". In our preliminary study without the short HIT expiration time, we noticed that some Turkers' task completion time was too long, even longer than an hour, suggesting those Turkers were not paying close attention to the task. To solve this problem, we set the HIT expiration time to 7 minutes, which ensures consistency in the ratings by forcing the Turker to complete the HIT in one sitting.

Tutorial Session

- In this tutorial you will watch a video clip and answer 10 questions regarding personality.
- After this tutorial, we want you to be comfortable answering the 10 questions on different video.
- You will have to do this tutorial only once. If you decide to do other HITs from us, you won't see this tutorial again.
- Please avoid preconception. Please answer each question based just on what you see in the video. Avoid answering each question based on your prior knowledge of the person.



Answer the 10 questions.

Remember! Please do not use your preconception of the person to answer these questions.

The person in the video	Disagree strongly				Agree strongly
1. The person is reserved					
2. The person is generally trusting					
3. The person tends to be lazy					
4. The person is relaxed, handles stress well					
5. The person has few artistic interests					
6. The person is outgoing, sociable					
7. The person tends to find fault with others					
8. The person does a thorough job					
9. The person gets nervous easily					
10. The person has an active imagination					

Submit

Figure 4-4: A screen shot of the tutorial HIT (page 2 of 2).

How do you perceive the personality of the others?

We want to understand how people perceive the personality of the others.
This questionnaire will gather data to help answer that question.

This HIT expires in 7 minutes. Please make sure you complete the task in one setting.

1. Before we start...

John Mellencamp

Do you know this person?
◉ Yes ○ No

How confident are you that you could describe the personality of this person?
not very confident ○ 1 ○ 2 ◉ 3 ○ 4 ○ 5 **strongly confident**

2. Read the instruction.

- **Think ahead.** We will ask you 10 questions (below) about the person in the video.
  Read them *now*, and think ahead how you would answer each question *as you watch the video.*
- **Avoid preconception.** Please answer each question *based just on what you see in the video.*
  In other words, avoid answering them based on your prior knowledge of the person.

The 10 questions:
How well do the following statements describe the person in the video?
(Adjust your browser so that these questions are visible while watching the video.)

1. The person is reserved
2. The person is generally trusting
3. The person tends to be lazy
4. The person is relaxed, handles stress well
5. The person has few artistic interests
6. The person is outgoing, sociable
7. The person tends to find fault with others
8. The person does a thorough job
9. The person gets nervous easily
10. The person has an active imagination

3. Now, watch the video. (duration: 28 seconds)

Video finished playing. Please answer the questions below.

Figure 4-5: A screen shot of the main HIT (page 1 of 3).

4. Answer the 10 questions.

How well do the following statements describe the person in the video?

John Mellencamp

1. The person is reserved
- ○ Disagree strongly
- ○ Disagree a little
- ○ Neither agree nor disagree
- ◉ Agree a little
- ○ Agree strongly

2. The person is generally trusting
- ○ Disagree strongly
- ○ Disagree a little
- ◉ Neither agree nor disagree
- ○ Agree a little
- ○ Agree strongly

3. The person tends to be lazy
- ○ Disagree strongly
- ◉ Disagree a little
- ○ Neither agree nor disagree
- ○ Agree a little
- ○ Agree strongly

4. The person is relaxed, handles stress well
- ○ Disagree strongly
- ○ Disagree a little
- ◉ Neither agree nor disagree
- ○ Agree a little
- ○ Agree strongly

5. The person has few artistic interests
- ◉ Disagree strongly
- ○ Disagree a little
- ○ Neither agree nor disagree
- ○ Agree a little
- ○ Agree strongly

6. The person is outgoing, sociable
- ○ Disagree strongly
- ○ Disagree a little
- ○ Neither agree nor disagree
- ◉ Agree a little
- ○ Agree strongly

7. The person tends to find fault with others
- ○ Disagree strongly
- ○ Disagree a little
- ○ Neither agree nor disagree
- ○ Agree a little
- ◉ Agree strongly

8. The person does a thorough job
- ○ Disagree strongly
- ○ Disagree a little
- ○ Neither agree nor disagree
- ○ Agree a little
- ◉ Agree strongly

9. The person gets nervous easily
- ○ Disagree strongly
- ○ Disagree a little
- ○ Neither agree nor disagree
- ◉ Agree a little
- ○ Agree strongly

10. The person has an active imagination
- ○ Disagree strongly
- ○ Disagree a little
- ◉ Neither agree nor disagree
- ○ Agree a little
- ○ Agree strongly

Figure 4-6: A screen shot of the main HIT (page 2 of 3). The 10 questions are shown in two-column to save space; in the web form they were shown in one-column.

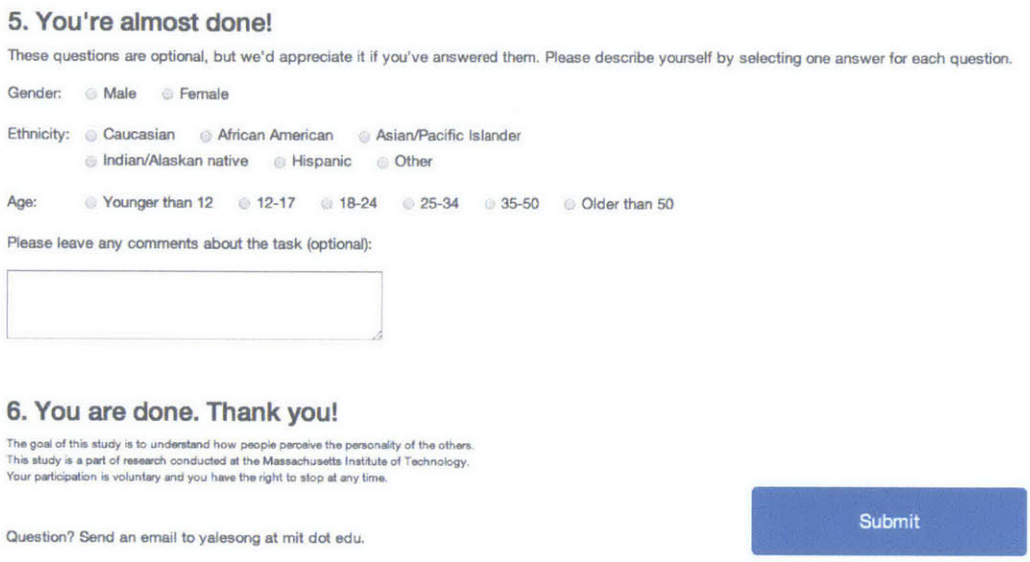


Figure 4-7: A screen shot of the main HIT (page 3 of 3).

The system starts by showing the name and a photo of the interviewee and asking "Do you know this person?" If the Turker answers "Yes", it then asks "How confident are you that you could describe the personality of this person?" on the scale of 1 (not very confident) to 5 (very confident). These questions collect information about the preconception of the Turker to the interviewee.

Then the task presents instructions by asking the Turker to "think ahead" and "avoid preconception," followed by showing the 10 questions. This allows the Turker to focus on the questions they will answer while watching the video. Our pilot tests showed that without this design (show the questions only after watching the video), they often resorted to random answers because they did not pay attention to those aspects that our questions address while watching the video clip.

The Turker then clicks on the play button from the embedded video player. While the Turker can pause and play the clip, the control bar is hidden to prevent the Turker from skipping ahead. The HIT opens the answer form only after the end of the clip, again to

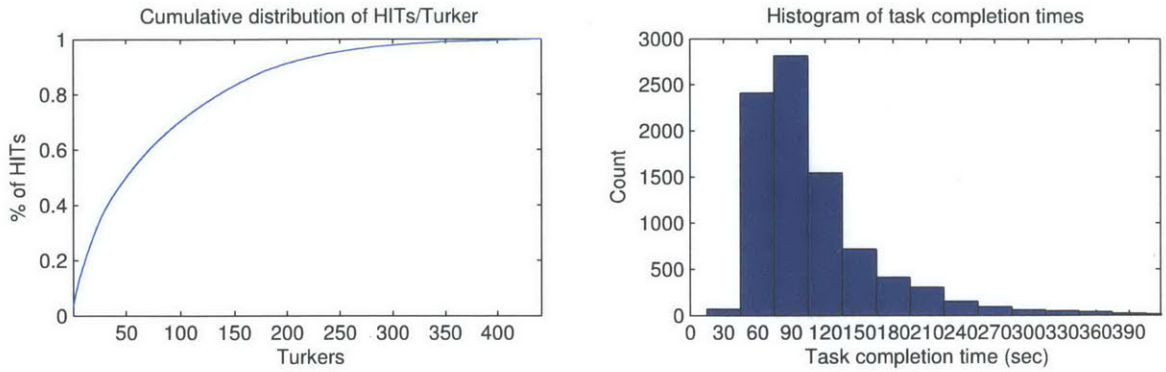


Figure 4-8: A cumulative distribution of HITs/Turker (left) and a histogram of task completion time (right).

ensure that the Turker watches the entire clip before answering the questions.

The main questionnaire shows each question statement (e.g., "The person is reserved") with five choices (Disagree strongly - Disagree a little - Neither agree nor disagree - Agree a little - Agree strongly). We positioned the questions and the choices vertically, which makes choosing a random choice as difficult as choosing the right choice. Making spamming as tedious as correctly answering is a recommended practice in crowdsourcing task design [59].

Post-Task Questionnaire Survey

Once the form detects that all answers are provided, it asks for optional, self-reported demographic information (gender, ethnicity, and age) and free-form comments. The Turker needs to click on the "submit" button below the optional form to complete the HIT.

4.5 Results from Mechanical Turk Study

A total of 441 unique Turkers completed 8,660 HITs in 14 hours. There were a few very active Turkers who completed a majority of the HITs: 14 Turkers finished more than 100 HITs each, 1937 HITs collectively (about 23% of total HITs), while 193 Turkers finished

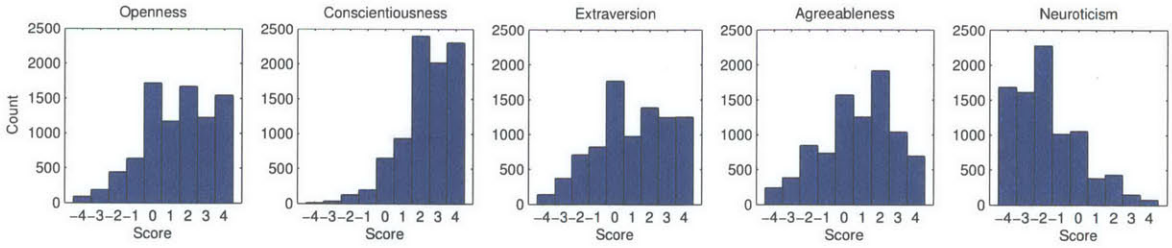


Figure 4-9: Score distributions across all five personality traits.

Trait	Mean	SD	Skew	Min	Max
O	1.37	1.94	-0.41	-4.00	4.00
C	2.35	1.48	-0.96	-4.00	4.00
E	0.99	2.11	-0.27	-4.00	4.00
A	0.74	2.04	-0.38	-4.00	4.00
N	-1.75	1.85	0.80	-4.00	4.00

Table 4.5: Descriptive statistics of the computed five personality traits.

less than 5 HITs each. Figure 4-8 (left) shows a cumulative distribution of HITs per Turker.

Figure 4-8 (right) shows a histogram of task completion times. About 80 of the HITs were finished in between one and two minutes; about 9% were finished in less than one minute, and about 10% were finished in more than 3 minutes. Considering the average length of video segments (38 seconds), and observations in the literature that it takes less than one minute to answer the BFI-10 questionnaire [88], we believe that most Turkers have finished HITs in one sitting with a reasonable amount of attention.

We computed numeric scores of the five personality traits based on the Turkers' annotations, using the formulas given by Rammstedt and John [88] (see Table 4.2). Figure 4-9 and Table 4.5 show distributions of the computed scores and their descriptive statistics across 8,760 annotations. Our Turkers had a tendency to give positive ratings to the interviewees in the videos: the distributions are skewed to the positive side of each dimension (a lower value in the Neuroticism dimension is the positive side).

Perhaps the most important question to ask is: How reliable are the scores? Do the

	Cronbach's Alpha					ICC(1,k)				
	O	C	E	A	N	O	C	E	A	N
Aran [3]	.54	.19	.72	.66	.09	.54	.19	.73	.66	.05
Biel [9]	.48	**.63**	.58	.46	**.61**	.47	.45	.76	.64	.42
Ours	**.79**	.49	**.77**	**.74**	.52	**.79**	**.49**	**.76**	**.74**	**.52**

Table 4.6: Cronbach's alpha coefficients and ICC(1,k) measures. The higher the value is, the more reliable the results are. The rule of thumb is that test scores have good reliability if the coefficient is between 0.7 and 0.9.

Turkers agree on their judgment of the interviewee's personality? To assess the reliability of the scores, we measured both the internal consistency (Cronbach's alpha) and inter-rater reliability (intraclass correlation) of the test scores.

Cronbach's alpha is a coefficient of an internal consistency based on average correlation among test items, while Intraclass Correlation Coefficient (ICC) measures homogeneity of the test scores in the same group. There are six different ways to compute ICC, depending on the design of the study and measurement types. Because each video segment is assessed by a different set of Turkers and we take an average of the scores given by 10 Turkers, we compute ICC(1,k) (k=10 in this case).

Table 4.6 shows both Cronbach's alpha and ICC(1,k) measures. As a calibration point, we also provide the results reported in Aran and Gatica-Perez [3] and Biel and Grtica-Perez [9] who collected similar data using Mechanical Turk. We can see that the scores are more reliable than previous datasets: our scores have higher ICC(1,k) measures than the two other datasets on every personality dimension.

4.6 Experiments

Our goal is to build a system that predicts people's impression about the personality of public figures from their multimodal behavior displayed in a short video clip. In this section, we evaluate how well our extracted features would perform in predicting the big

five personality scores of each video segment.

4.6.1 Problem Formulation

Different kinds of problems can be explored with our dataset, e.g., classification, regression, clustering, and ranking. In this work, we pose our problem as a classification problem with two categories for each of the five dimensions: whether a score is below or above the average of everyone's score in the dataset. Our decision to use the average as the categorization boundary comes from the fact that score distributions of all five dimensions are skewed, as shown in Figure 4-9. Categorizing scores with respect to an absolute boundary (e.g., score 0) would result in a heavily imbalanced data, compounding the experiment and making it harder to study the quality of the extracted features.

Because we use the average score as a categorization boundary, the number of samples per class is well balanced. The numbers of samples above and below the average for each dimension were: 399 vs. 467 (O), 406 vs. 460 (C), 414 vs. 452 (E), 378 vs. 488 (A), 452 vs. 414 (N).

4.6.2 Bag-of-Words Feature Representation

The features we extracted from four modalities (face, body motion, utterance, and prosody) have different representation formats. In terms of the sampling rate, the face features are extracted per frame, the body motion features per trajectory length (lengths vary among trajectories), the utterance features per sequence, and the prosody features per window (of size 50ms at 10ms interval). The body motion features are extracted from local trajectories, so there are varying numbers of features extracted from each video segment. In order to obtain a unified representation of this heterogeneous set of features, we use the bag-of-words approach.

For all three modalities except the utterance, we perform vector quantization using the

K-means algorithm: learn K centroids for each modality and assign the closest centroid ID to each feature vector. The number of centroids K is cross-validated across 100, 200, and 500. When learning the centroids, we subsample by a factor of 100 from the entire set of feature vectors: this results in about 10,000 samples for face, 35,000 samples for body motion, and 30,000 samples for prosody. After the vector quantization, we produce a histogram representation of the features from each modality by aggregating the centroid IDs into the K bins and normalizing the counts to sum up to one.

Note that we do not perform feature bagging for the the utterance modality because the LDA feature vector is represented per-video – this matches the sampling granularity of the histogram representation of three other modalities.

Finally, we obtain a per-video feature vector representation by concatenating the three histograms and one LDA feature vector into one vector. This results in a vector of size $3\times\mathrm{K} + 50$ (LDA). The data is then normalized to have zero mean and standard deviation of one (i.e., the z-score).

4.6.3 Methodology

We use a Support Vector Machine (SVM) [113] with the RBF kernel as our prediction model.[4] The RBF kernel width was fixed to one over feature dimension, which has empirically shown to produce good performance [20]. We cross-validated the penalty parameter of the error term $C = 10^c$ with $c = [1 \cdots 6]$.

In order to choose the optimal values of the hyper-parameters (the number of K-means centroids and the penalty term C in the SVM), we perform 10-fold cross-validation in the subject-independent setting: we first evenly split the 866 samples into 10 subsets, then further refine them by moving the samples across the subsets so that no two subsets share video segments of the same person. Since the number of segments per person is roughly uniform, our splitting scheme produced a uniform distribution of samples per subset: [92

[4]We use the LIBSVM software package developed by Chang and Lin [20].

83 87 85 89 80 94 78 88 90]. We use one subset for testing, another subset for validation, the rest for training the model, repeating ten times. The optimal combination of parameter values is then chosen as the one that gives the best performance on the validation split, averaged out across 10 folds; below we report performance on both the validation split and the test split for completeness.

Our goal in this experiment is to see which combination of modalities has the best predictive power. To this end, we run experiments on each of the fifteen combinations of the four modalities. The performances are reported for each combination. The total number of test cases is 13,500: 10 (fold) $\times$ 3 (K-means centroid) $\times$ 15 (modality combinations) $\times$ 6 (SVM cost term C) $\times$ 5 (OCEAN).

4.6.4 Results and Discussion

Table 4.7 shows the accuracy and the F1 score performances from each of the fifteen combinations of modalities, across all five personality dimensions; Figure 4-10 and Figure 4-11 show bar graphs.

There was no single combination of modalities that performed the best across all five dimensions, but on average a combination of all four modalities performed the best in terms of F1 score on the test split.

It is interesting to see that unimodal features often outperformed multimodal features: in terms of the accuracy on the test split, the face features performed the best on both the extraversion and the neuroticism dimension, and the utterance features performed the best on the agreeableness dimension. In terms of the F1 score on the test split, the face features performed the best on both the conscientiousness and the neuroticism dimension, and the utterance features performed the best on both the openness and agreeableness dimension. As shown in Figure 4-10 and Figure 4-11, however, the differences between the best performing combinations and other combinations were not significant.

		Validation Split						Test Split					
		O	C	E	A	N	Avg.	O	C	E	A	N	Avg.
Accuracy	Face (F)	0.60	0.54	0.55	0.54	**0.56**	0.56	0.58	0.54	**0.57**	0.54	**0.56**	0.56
	Body (B)	0.60	0.53	0.49	0.58	0.55	0.55	0.62	0.48	0.48	0.56	0.55	0.54
	Pros. (P)	0.59	0.53	0.59	0.57	0.55	0.57	0.59	0.52	0.56	0.56	0.55	0.56
	Utte. (U)	0.62	0.53	0.53	**0.63**	0.50	0.56	0.65	0.53	0.54	**0.64**	0.49	0.57
	FB	0.61	0.52	0.55	0.59	0.52	0.56	0.61	0.50	0.56	0.54	0.52	0.55
	FP	0.62	**0.55**	0.57	0.55	0.54	0.57	0.61	0.53	0.55	0.53	0.53	0.55
	FU	0.61	0.53	0.55	0.60	0.50	0.56	0.60	0.53	0.52	0.59	0.51	0.55
	BP	0.57	0.51	0.56	0.53	0.54	0.54	0.58	0.51	0.54	0.53	0.53	0.54
	BU	0.61	0.52	0.51	0.62	0.51	0.55	0.64	0.47	0.53	0.58	0.48	0.54
	PU	**0.64**	0.52	0.58	0.61	0.54	**0.58**	0.63	0.50	0.55	0.62	0.55	0.57
	FBP	0.59	0.54	0.59	0.57	0.51	0.56	0.58	0.53	0.56	0.55	0.53	0.55
	FBU	0.63	0.53	0.55	0.60	0.53	0.57	0.62	0.50	0.53	0.59	0.52	0.55
	FPU	0.62	0.54	0.57	0.57	0.53	0.57	0.64	**0.55**	0.57	0.58	0.54	**0.58**
	BPU	0.61	0.51	0.58	0.57	0.53	0.56	**0.65**	0.52	0.57	0.56	0.54	0.57
	FBPU	0.61	0.51	**0.60**	0.58	0.53	0.56	0.63	0.52	0.55	0.57	0.53	0.56
F1 Score	Face (F)	0.65	**0.63**	0.52	0.62	0.50	0.58	0.59	**0.63**	0.51	0.59	**0.54**	0.57
	Body (B)	0.64	0.59	0.52	0.64	0.52	0.58	0.62	0.52	0.51	0.55	0.49	0.54
	Pros. (P)	0.59	0.59	0.60	0.59	0.53	0.58	0.56	0.60	0.56	0.59	0.45	0.55
	Utte. (U)	0.66	0.60	0.58	0.70	0.46	0.60	**0.68**	0.59	0.58	**0.70**	0.47	0.61
	FB	0.66	0.55	0.58	0.67	0.50	0.59	0.63	0.56	0.55	0.62	0.48	0.57
	FP	0.66	0.60	0.59	0.63	0.45	0.59	0.63	0.59	0.59	0.60	0.47	0.58
	FU	0.65	0.58	0.56	0.67	0.49	0.59	0.62	0.56	0.56	0.67	0.48	0.58
	BP	0.58	0.55	0.57	0.60	0.52	0.57	0.57	0.58	0.54	0.61	0.47	0.55
	BU	0.66	0.57	0.55	0.66	0.46	0.58	0.61	0.54	0.53	0.63	0.44	0.55
	PU	0.65	0.55	0.61	0.66	**0.54**	0.60	0.65	0.54	0.56	0.67	0.52	0.59
	FBP	0.65	0.57	0.61	0.69	0.47	0.60	0.63	0.61	**0.63**	0.66	0.47	0.60
	FBU	**0.67**	0.56	0.58	0.70	0.47	0.60	0.66	0.54	0.56	0.69	0.46	0.58
	FPU	0.66	0.59	0.59	0.69	0.45	0.60	0.67	0.60	0.55	0.67	0.48	0.59
	BPU	0.62	0.54	0.60	0.63	0.51	0.58	0.60	0.58	0.57	0.65	0.49	0.58
	FBPU	0.66	0.57	**0.63**	**0.70**	0.46	**0.60**	0.68	0.61	0.57	0.69	0.48	**0.61**

Table 4.7: Accuracy and F1 score results. Bold faced values indicate the best modality combination for predicting each personality dimension.

In this experiment, we took the early fusion approach: features from different modalities were concatenated into a single vector, which was then used as an input to an SVM as if the features were from a single modality. Although we used information from all four modalities, we did not fully leverage the structure in multimodal data, e.g., correlation and interaction across modalities. We focus on exploiting this sort of multimodal structure in the next chapter.

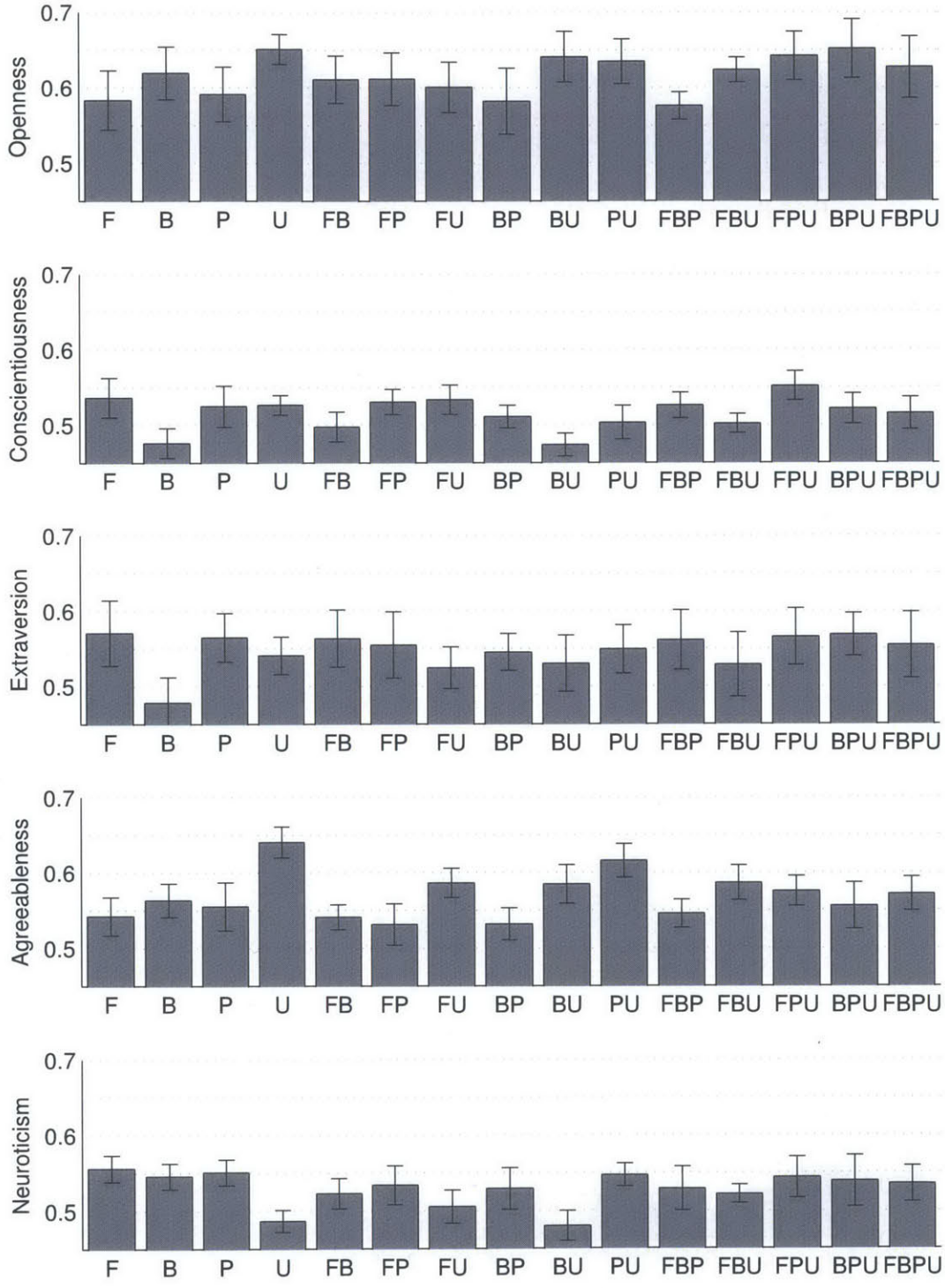


Figure 4-10: Accuracy results. Legend: Face (F), Body (B), Prosody (P), and Utterance (U). Error bars indicate 95% confidence intervals.

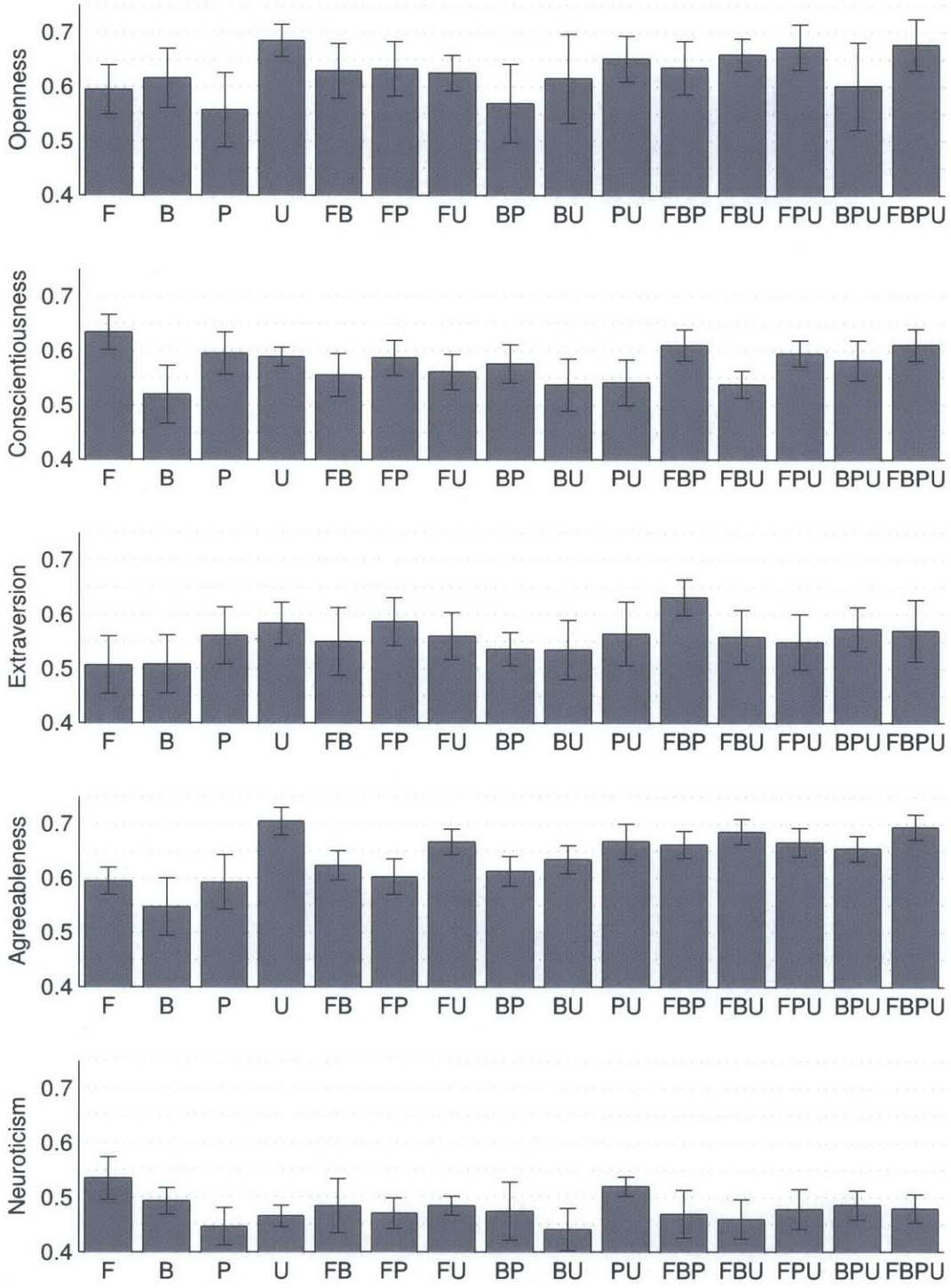


Figure 4-11: F1 score results. Legend: Face (F), Body (B), Prosody (P), and Utterance (U). Error bars indicate 95% confidence intervals.

Chapter 5

Learning Multimodal Structure of Human Behavior

Human communication involves a complex interplay of multiple modalities, e.g., neither speech nor facial expression alone entirely conveys the true intention behind sarcastic expressions like "that was great" (said with an uninterested facial expression); the sarcasm is conveyed only through multimodal signal as a whole. As such, building a system for understanding human behaviors requires reasoning about information from multiple modalities [129]. While it is evident that *"the more the better"* strategy fits well into this scenario, the question remains: how should we combine information from multiple modalities?

This chapter presents a novel data fusion algorithm that uses structured sparsity to combine information from multiple modalities. The key observation behind this work is group sparsity in multimodal features: features from just a few of the modalities can contain all the information needed at any given time. It is not always the case that all modalities convey useful information; sometimes we express emotions while remaining silent.

Structured sparsity has emerged as a powerful technique for representing this kind of structure in a parsimonious way, and has found numerous applications in signal processing and pattern recognition [121, 53, 4]. We use structured sparsity both to construct a dictionary

of multimodal expressions and to transform a set of multimodal features into a sparse set of coefficients with respect to the dictionary.

The main novelty in our work is the hierarchical formulation of factorized multimodal subspaces via structured sparsity: we factorize the multimodal feature space into modality-*shared* subspaces and modality-*private* subspaces, then learn a hierarchical structure among them based on the superset/subset relationship. The dictionary can be seen as a collection of basis vectors embedded in a hierarchy of multimodal subspaces; using structured sparsity, we encourage the hierarchy to capture the intrinsic structure of a multimodal feature space. In essence, the hierarchical formulation allows our dictionary to capture complex dependence/independence structure across multiple modalities in a principled manner.

We start by introducing the data fusion problem in the context of multimodal signal understanding and briefly review related work, and provide an intuition behind our approach that exploits the hierarchical structure among multiple modalities. We then describe our mixed-norm formulation that uses structured sparsity to perform data fusion, and provide experimental results and discuss the effectiveness of our approach.

5.1 Data Fusion for Human Behavior Understanding

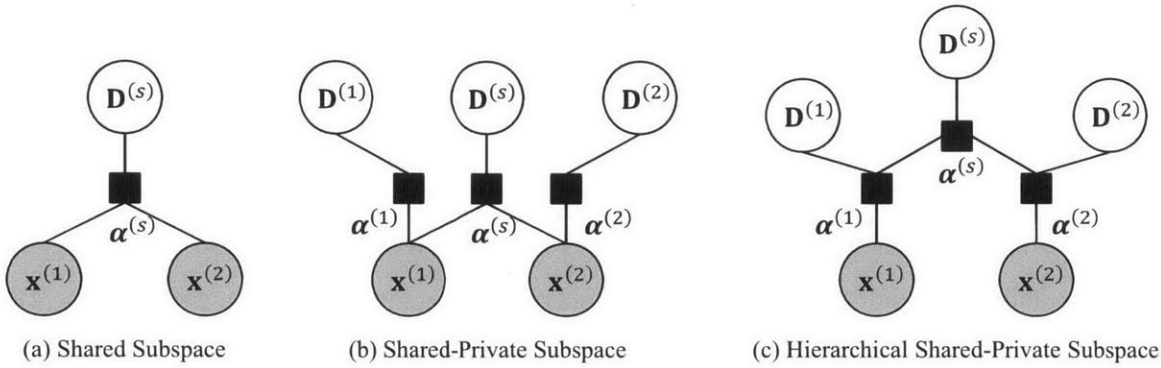
The goal of data fusion is to combine a set of multiple heterogeneous features in a such way that it makes the best use of the richness of information available. Two simple and often used approaches are early and late fusion [97]: early fusion treats information from multiple modalities as if they were from a single modality and simply concatenate them into one feature vector, while late fusion treats information from each modality independently until a decision is made. Experience suggests that both approaches fail to account for statistical relationship among the modalities [125]: in early fusion, a modality with strong dynamics (e.g., high variance) can dominate the inference procedure [22]; while late fusion discards any statistical relationship between modalities (e.g., correlation) [97].

Part of the difficulty in data fusion comes from the fact that information can be complementary, redundant, and contradictory across modalities [129]. Several approaches have been developed to exploit these properties of multimodal data. Co-training [12] and Multiple Kernel Learning (MKL) [64, 108] have shown promising results when the modalities are independent, i.e., they provide different and complementary information. However, when the modalities are not independent, as is common in human behavior understanding, these methods often fail to learn from the data correctly [62]. Canonical Correlation Analysis (CCA) [43] and Independent Component Analysis (ICA) [49] have shown a powerful generalization ability to model correlation and independence between modalities, respectively. However, the assumptions made by these techniques are rather strict in many real-world scenarios.

5.1.1 Multimodal Subspaces

Recently, there has been a surge of interest in learning a multimodal dictionary by exploiting group sparsity in multimodal signals. In particular, it has been shown that factorizing a multimodal signal space into parts corresponding to an individual modality and parts that are shared across multiple modalities leads to improvements in multimodal signal understanding [74, 54, 110, 131, 21]. We call such factorized spaces the *multimodal subspaces*, and the two kinds of subspaces modality-*private* and modality-*shared* subspaces, respectively. Intuitively, modality-private subspaces account for the patterns within each modality that are independent of other modalities, while modality-shared subspaces account for the patterns that are dependent on other modalities.

Jia *et al.* [54] showed that such subspaces can be found in a convex optimization framework using structured sparsity, imposing a group-wise sparsity-inducing norm on the basis vectors (i.e., the columns) of a dictionary matrix. Once such a dictionary is constructed, a multimodal signal vector can be represented as a linear combination of the basis vectors using sparse coding [112, 82]. Their approach is illustrated in Figure 5-1 (b), where signals



(a) Shared Subspace (b) Shared-Private Subspace (c) Hierarchical Shared-Private Subspace

Figure 5-1: Factor graph representations of three two-modality subspace models: (a) shared subspace model, (b) shared-private subspace model [54], (c) our hierarchical shared-private subspace model. An input signal from two modalities $[\mathbf{x}^{(1)}; \mathbf{x}^{(2)}]$ is represented in terms of basis vectors from a shared subspace $\mathbf{D}^{(s)}$ and basis vectors from a private subspace $[\mathbf{D}^{(1)}; \mathbf{D}^{(2)}]$ via the coefficient term $\boldsymbol{\alpha}^{(\cdot)}$.

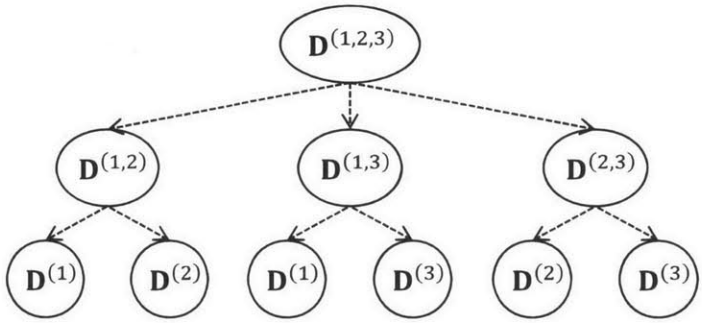


Figure 5-2: Multimodal subspaces form a hierarchical tree structure induced by the superset/subset relationship.

from both modalities $[\mathbf{x}^{(1)}; \mathbf{x}^{(2)}]$ are represented in terms of basis vectors from a shared subspace $\mathbf{D}^{(s)}$ and basis vectors from a private subspace $[\mathbf{D}^{(1)}; \mathbf{D}^{(2)}]$ via the coefficient term $[\boldsymbol{\alpha}^{(1)}; \boldsymbol{\alpha}^{(s)}; \boldsymbol{\alpha}^{(2)}]$. They showed that this sparse representation effectively accounts for dependence and independence among multiple modalities [54].

5.1.2 Intuition: Hierarchical Multimodal Subspaces

We observe that multimodal subspaces have a superset/subset relationship that induces a hierarchical structure of the sort shown in Figure 5-1 (c): a subspace $\mathbf{D}^{(s)}$ is a superset of two subspaces defined over the corresponding modalities $[\mathbf{D}^{(1)}; \mathbf{D}^{(2)}]$, thus $\mathbf{D}^{(s)}$ can be seen as a parent of $[\mathbf{D}^{(1)}; \mathbf{D}^{(2)}]$. Figure 5-2 illustrates a hierarchy of tri-modal subspaces constructed following the superset/subset rule.

Our intuition is that leveraging this hierarchical structure will enable the multimodal subspaces to capture the dependence/independence relationships across modalities accurately. From the hierarchical structure we can use the hierarchical sparsity rule [53]: a subspace $\mathbf{D}^{(i)}$ will participate in reconstructing an input signal $\mathbf{x}$ (i.e., the corresponding weight term $\boldsymbol{\alpha}^{(\cdot)}$ is non-zero), only if all of its parent subspaces $\mathbf{D}^{(j)}$ are participating as well, where j's are the indices of the parent of the i-th node. For example, in order for the subspace $\mathbf{D}^{(3)}$ under $\mathbf{D}^{(2,3)}$ to participate, its parent subspaces $\mathbf{D}^{(2,3)}$ and $\mathbf{D}^{(1,2,3)}$ have to participate as well. The sparsity constraint ensures that only a few paths (from the root to the leaves) are active in reconstructing the original signal. This effectively allows the sparse representation to select the most important subset of modalities that best represent the given signal.

We show that it is possible to learn global and local patterns of multimodal data by constructing a multimodal dictionary using the hierarchical sparsity constraint. Two characteristics make this possible: (i) the range of modalities that each subspace covers, and (ii) the frequency of each subspace being active in signal reconstruction. High-level subspaces span over a wider range of modalities than low-level subspaces, and are active more frequently than low-level subspaces in signal construction. These two characteristics encourage high-level subspaces to capture the global patterns shared across multiple modalities, and low-level subspaces to capture the local details narrowed down to specific modalities. For example, a multimodal signal representing "laughing out loud" will be reconstructed as a combination of a high-level subspace "highly aroused" and low-level

subspaces "the appearance of mouth" and "the pitch of the voice."

5.2 Data Fusion by Structured Sparsity

Our goal is to obtain a discriminative representation of multimodal signal that makes it easier to discover its important patterns in the subsequent analysis step, e.g., classification. In this section, we describe our approach to performing data fusion with structured sparsity, learning hierarchical multimodal subspaces. We start by defining the notations used in this chapter, and briefly review the current dictionary learning and group sparse coding techniques, which becomes the foundation of our approach. We then describe a mixed-norm formulation of this work.

5.2.1 Notation

Lower-case letters denote scalars, lower-case bold-face letters denote vectors, and upper-case bold-face letters denote matrices. For a vector $\mathbf{x}$, we use a subscript x_j to denote the j-th element; for a matrix $\mathbf{X}$, we use the subscript $\mathbf{X}_j$ ($\mathbf{X}_{\cdot,j}$) to denote the j-th column (row). Given a set of indices Ω, we denote a subvector or a submatrix formed by taking only Ω elements in similar ways. For multimodal features, we use a superscript in parenthesis $\mathbf{X}^{(v)}$ to emphasize that the features are from the v-th modality.

5.2.2 Problem Formulation

Suppose the data consists of N real-valued d-dimensional feature vectors stacked column-wise into a matrix $\mathbf{X} \in \mathbb{R}^{d \times N}$; each column vector $\mathbf{X}_i \in \mathbb{R}^d$ is a concatenation of signals collected from multiple modalities.

Given the input data $\mathbf{X}$, our goal is to find a dictionary $\mathbf{D} \in \mathbb{R}^{d \times k}$ of k real-valued basis

vectors $\mathbf{D}_j \in \mathbb{R}^d$, and coefficients $\boldsymbol{\alpha} \in \mathbb{R}^{k \times N}$ that determines the relative influence of the basis vectors in reconstructing the input $\mathbf{X}$. We formulate this objective as the matrix factorization problem [94] that decomposes $\mathbf{X}$ into $\mathbf{D}$ and $\boldsymbol{\alpha}$:

$$\min_{\mathbf{D},\boldsymbol{\alpha}} \|\mathbf{X} - \mathbf{D}\boldsymbol{\alpha}\|_{\mathcal{F}}^2 \tag{5.1}$$

where $\|\mathbf{X}\|_{\mathcal{F}}^2 = \sum_{i,j} |X_{i,j}|^2$ is the squared Frobenius norm.

Our focus is to learn the dictionary $\mathbf{D}$ that successfully encodes intrinsic properties of the multimodal data $\mathbf{X}$, which in turn allows for coefficients $\boldsymbol{\alpha}$ to have the maximal discriminatory power.

5.2.3 Sparse Coding

Recently, sparse representations have proven successful in numerous computer vision and machine learning tasks [121]. They assume that an input space is inherently sparse, and represent an input signal as a linear combination of a few most relevant basis vectors.

Following this line of research, we can rewrite Equation 5.1 using sparse coding [69]:

$$\begin{aligned} &\min_{\mathbf{D},\boldsymbol{\alpha}} \frac{1}{2} \|\mathbf{X} - \mathbf{D}\boldsymbol{\alpha}\|_{\mathcal{F}}^2 + \lambda \|\boldsymbol{\alpha}\|_{1,1} \\ &s.t.\ \|\mathbf{D}_j\|_2 \leq 1\ \forall j \in \{1 \cdots k\},\ \lambda > 0 \end{aligned} \tag{5.2}$$

where $\|\boldsymbol{\alpha}\|_{1,1} = \sum_{i,j} |\alpha_{i,j}|$ is an $l_{1,1}$ norm that encourages element-wise sparsity (most elements equal zero), and λ controls the relative importance of the reconstruction error and sparsity. For numerical stability, the $\mathbf{D}_j$ are often constrained to have an l_2 norm less than one, i.e., $\|\mathbf{D}_j\|_2 \leq 1$.

In the context of multimodal learning, the data possess a strong group structure induced by the modality configuration, e.g., groups of audio and visual signals. Unfortunately, standard (l_1 norm based) sparse coding cannot model any structural information in the

data, which leads us to describe structured sparsity that overcomes this shortcoming.

5.2.4 Structured Sparsity

Group sparse coding [52, 7] encourages sparsity at the group level rather than at the element level, capturing intrinsic properties in groups of correlated variables. This has shown to be useful in many scenarios, e.g., encoding an image using a group of dictionary entries trained on the same object category [7].

In the context of multimodal learning, Jia *et al.* [54] used group sparse coding to learn a dictionary of multimodal basis vectors. Let $\mathcal{V} = \{\Omega_v\}_{v=1}^{V}$ be a set of disjoint index groups, where each Ω_v specifies the indices $j \in \{1 \cdots d\}$ of the vector $\mathbf{X}_i \in \mathbb{R}^d$ that corresponds to the v-th modality. As a toy example, consider a two-modality case where the first modality ($v = 1$) is of dimension 10 and the second modality ($v = 2$) is of dimension 20. Concatenated vertically, the feature vector $\mathbf{X}_i$ is of dimension $d = 30$, and the index set $\Omega_1 = [1 \cdots 10]$ and $\Omega_2 = [11 \cdots 30]$.

Jia *et al.*defines an objective function using an $l_{1,p}$ norm on the dictionary $\Phi_{\mathcal{V}}(\mathbf{D})$:

$$\min_{\mathbf{D},\boldsymbol{\alpha}} \frac{1}{2} \|\mathbf{X} - \mathbf{D}\boldsymbol{\alpha}\|_{\mathcal{F}}^2 + \lambda_1 \Phi_{\mathcal{V}}(\mathbf{D}) + \lambda_2 \|\boldsymbol{\alpha}\|_{1,1} \tag{5.3}$$

where $\Phi_{\mathcal{V}}(\mathbf{D}) = \sum_{j=1}^{k} \|\mathbf{D}_j\|_{1,p}$ is the sum of an $l_{1,p}$ norm induced by group $\mathcal{V}$ (p is usually 2 or ∞), defined as

$$\|\mathbf{D}_j\|_{1,p} = \sum_{v=1}^{V} \|\mathbf{D}_{j,\Omega_v}\|_p = \sum_{v=1}^{V} \left(\sum_{l \in \Omega_g} |\mathbf{D}_{j,l}|^p \right)^{1/p} \tag{5.4}$$

The $l_{1,p}$ norm imposed on each basis vector $\mathbf{D}_j$ makes certain groups of elements zero (or non-zero) simultaneously. Continuing our toy example above, the dictionary $\mathbf{D}$ will be divided into three subspaces: $\mathbf{D}^{(1,2)}$ where all elements are non-zero; $\mathbf{D}^{(1)}$ where elements are non-zero for Ω_1 and zero for Ω_2; and $\mathbf{D}^{(2)}$ where elements are non-zero for Ω_2 and

zero for Ω_1. This enables the dictionary to learn multimodal basis vectors factorized into modality-private ones that correspond to a specific modality and modality-shared ones that span over multiple modalities. This model is illustrated in a factor graph representation in Figure 5-1 (b).

Note that, while the formulation above considers a row-wise group structure on the dictionary $\mathbf{D}$, producing multimodal subspaces, it fails to consider any column-wise structure among the subspaces. Our goal is to exploit the hierarchical structure among the columns of the dictionary. Next we introduce the concept of *hierarchical structured sparsity* [53], and show that there exists a hierarchical structure among the basis vectors that we can encode using hierarchical structured sparsity.

5.2.5 Learning Hierarchical Multimodal Subspaces

We now turn to multimodal dictionary learning using *hierarchical structured sparsity* [53]. We start by observing that a dictionary $\mathbf{D}$ defined in Equation 5.3 has an inherent hierarchical structure among multimodal basis vectors (among columns). This structure is formed by the superset-subset relations induced by the power set of the modality groups $\mathcal{P}(\mathcal{V})$, e.g., given a power set $\mathcal{P}(\mathcal{V}) = \{\{\Omega_1, \Omega_2\}, \{\Omega_1\}, \{\Omega_2\}, \emptyset\}$, the set $\{\Omega_1, \Omega_2\}$ is a superset of $\{\Omega_1\}$ and $\{\Omega_2\}$ (see Figure 5-2).

Let $\mathcal{T} = \{\Omega_g\}_{g=1}^{G}$ be a rooted tree-structured set of groups[1], and $\mathcal{T}(\Omega_g)$ be a subset of $\mathcal{T}$ that contains Ω_g and all the groups that are parents in the tree (all the way up to the root). Jenatton *et al.* [53] defines the hierarchical group sparsity norm $\Psi_{\mathcal{T}}(\mathbf{x})$ that obeys the following constraint:

$$\forall \Omega_g \in \mathcal{T}, \boldsymbol{\alpha}_{\Omega_g} \neq 0 \Rightarrow [\boldsymbol{\alpha}_{\Omega_k} \neq 0, \ \forall \Omega_k \in \mathcal{T}(\Omega_g)] \tag{5.5}$$

In words, all non-zero coefficients in a vector $\boldsymbol{\alpha}$ follow the property that if the elements

[1] A rooted tree is an undirected connected graph without cycles and with a unique root node.

of a set Ω_g are non-zero, all other elements of the sets in $\mathcal{T}(\Omega_g)$ are likewise non-zero. This constraint effectively enforces that basis vectors belonging to a higher-level subspace are used more frequently than the ones belonging to a lower-level subspace. Thus we can expect that basis vectors from a high-level subspace capture generic concepts that span across multiple modalities, while basis vectors from a low-level subspace capture details specific to a narrower subset (or a single) modalities.

More formally, given a set of modality groups $\mathcal{V}$, we define $\mathcal{T}_\mathcal{V}$ as the power set $\mathcal{P}(\mathcal{V})$ with a rooted tree structure. Note that, for more than two modalities, the powerset hierarchy is no longer a tree (e.g., $\mathbf{D}^{(1)}$ is a child node of both $\mathbf{D}^{(1,2)}$ and $\mathbf{D}^{(1,3)}$). To make it a tree structure, we evenly split basis vectors from these nodes and assign them to their corresponding parents (see Figure 5-2 for example). An important thing to note is that, by splitting the child nodes, we allow the basis vectors to adapt to the context the modality is in. For example, basis vectors from $\mathbf{D}^{(1)}$ under $\mathbf{D}^{(1,2)}$ and under $\mathbf{D}^{(1,3)}$ will contain information under the context of its parent subspace.

Using the rules we describe above, we define our mixed-norm formulation for learning hierarchical multimodal subspace as

$$\min_{\mathbf{D},\boldsymbol{\alpha}} \frac{1}{2} \|\mathbf{X} - \mathbf{D}\boldsymbol{\alpha}\|_\mathcal{F}^2 + \lambda_1 \Phi_\mathcal{V}(\mathbf{D}) + \lambda_2 \Psi_{\mathcal{T}_\mathcal{V}}(\boldsymbol{\alpha}) \tag{5.6}$$

The first term ensures that we are minimizing over the reconstruction error; the second term is the same as the corresponding term in Equation 5.3 and ensures the shared/private factorization of multimodal feature space; the third term is the hierarchical group sparsity norm with the rule in Equation 5.5 and ensures that the columns of the matrix $\mathbf{D}$ are organized in a hierarchical tree structure.

The optimization problem in Equation 5.6 is convex in $\mathbf{D}$ with $\boldsymbol{\alpha}$ fixed, and vice versa. We therefore solve the optimization by alternating between minimizing the objective with respect to $\mathbf{D}$ and $\boldsymbol{\alpha}$.

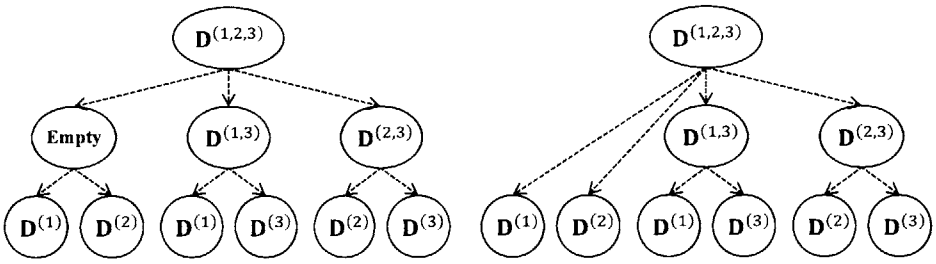


Figure 5-3: When a subspace node in the hierarchy becomes empty (no basis vector for that subspace), we delete that node and assign the descendant nodes to the parent node.

5.2.6 Tree Building Procedure

We now explain the procedure for building a hierarchical tree structure from a dictionary of basis vectors. To explain the procedure, we use a hierarchical tree structure constructed from subspaces of the three modalities shown in Figure 5-2. To be concrete, we assume that each modality is represented as a 10-dimensional feature vector. Note in the figure that there are several duplicate subspaces among leaf nodes, e.g., a node for subspace $\mathbf{D}^{(1)}$ appears both under $\mathbf{D}^{(1,2)}$ and $\mathbf{D}^{(1,3)}$.

Given a dictionary of basis vectors, we first find for each basis vector which subspace(s) each basis vector belongs to. This is done by checking which elements in the vector are non-zero and by finding the modalities that the non-zero indices correspond to. In the example above, for instance, if a basis vector has non-zero elements in indices 11:30, it belongs to the subspace $\mathbf{D}^{(2,3)}$, because features from modality 2 span over the indices 11:20 and features from modality 3 span over the indices 21:30. We then assign the basis vectors to their corresponding subspaces in a hierarchy. For the duplicate subspace case, we evenly distribute the basis vectors across the corresponding subspace nodes.

When a subspace node in the hierarchy becomes empty (no basis vector for that subspace), we delete that node and assign the descendant nodes to the parent node (see Figure 5-3). For example, if there is no basis vector for the subspace $\mathbf{D}^{(1,2)}$ (labeled as "empty" in Figure 5-3), its two descendants $\mathbf{D}^{(1)}$ and $\mathbf{D}^{(2)}$ are connected to the root node. Compare

this approach to a different approach where the descendants are combined with existing nodes with the same subspace, e.g., basis vectors of the subspace $\mathbf{D}^{(1)}$ under an empty subspace $\mathbf{D}^{(1,2)}$ are assigned to the subspace $\mathbf{D}^{(1)}$ under the subspace $\mathbf{D}^{(1,3)}$. We believe that our approach makes the basis vectors "contextualized" with respect to their locations in the tree. Nodes at different locations in the tree have different ancestors, and because of the hierarchical sparsity constraint (Equation 5.5), basis vectors in a node will participate in reconstruction together with basis vectors in its ancestor nodes. So basis vectors from different locations are participating in reconstruction in different contexts.

Our tree building procedure is performed at each iteration of the optimization. After the first iteration, we initialize the hierarchical structure by constructing the powerset $\mathcal{P}(\mathcal{V})$. Throughout the optimization, the structure gets refined to best explain the data. Note that, as the objective function converges to its minima, basis functions from some subspace $\mathbf{D}^{(\cdot)}$ will be dropped if not necessary in the signal reconstruction. In this case, the child nodes of that subspace are assigned to the parent of the dropped node, as in the example above.

5.3 Experiments

We evaluated our hierarchical multimodal subspace learning approach on the Time10Q dataset. In particular, in order to study the benefit of using structured sparsity for data fusion, we compared our approach to two other sparsity approaches shown in Figure 5-1.

We first describe the data preprocessing procedure necessary to use the Time10Q dataset on our approach. We then explain the dictionary learning and feature coding procedure with three different models we evaluate (shown in Figure 5-1). Next we detail our experimental methodology and discuss the result.

5.3.1 Data Preprocessing

As features from different modalities in the Time10Q dataset have different sampling rates, we need proper time synchronization of multimodal features. The face features (PHOG) are extracted per frame, the body motion features (HOG/HOF/MBH) are extracted per local voxel, the prosody features (F0/NHR/loudness contour) are extracted every 10ms, and the utterance features (LDA) are extracted per sequence (see the previous chapter for details).

We first align the face and body motion features. Each body motion feature vector is associated with meta information indicating where the voxel is located in a video sequence: an exact location of the voxel in time (frame number) and space (pixel coordinate) as well as the size of the voxel in time (duration) and space (width and height). To align the body motion features to the face features, we collect all the body motion features that overlap with each frame of the video, and obtain per-frame feature representation by performing max pooling over the collection of features.

Given a set of vectors, max pooling creates a single vector with the same dimension as each of the given vectors, taking the maximum absolute value for each dimension across all the given vectors. Another pooling method often used is average pooling, which instead takes the average of the values. Research has shown that max pooling provides a better representation than average pooling, one that is robust to image transformations, noise, and clutter [127]. A theoretical analysis given by Boureau *et al.* [15] highlights the superiority of max pooling over average pooling. We note that different pooling methods can also be used, such as pooling from a pyramid of multiple spatio-temporal scale similar to Lazebnik *et al.* [67], which has been shown to help capture surrounding context; for a thorough discussion of various feature encoding and pooling methods, see [23, 55].

Next, we align the visual features (the face and body motion features we aligned) to the prosody features. The prosody features have a higher sampling rate than the visual features: they are extracted every 10ms (100 times a second), while visual features are

extracted every 33ms to 42ms (24 to 30 times a second), depending on the codec used. We therefore upsample the visual features by linear interpolation, to match the sampling rate of the prosody features. (Another way would be downsampling the prosody features to match the sampling rate of the visual features. We chose to upsample instead of downsample because downsampling could miss out on vital information in acoustic signals such as high spikes.)

The last to align is the utterance features, which is extracted per sequence. For the same reason described above, we upsample the utterance features by replicating the LDA vector as many times as the length of the video sequence.

5.3.2 Dictionary Learning and Feature Coding

We compare three approaches: LASSO [112], the factorized multimodal subspace (FMS) approach by Jia *et al.* [54], and our hierarchical multimodal subspace (HMS) approach. All three approaches consist of two steps: dictionary learning and feature coding.

We trained the dictionaries for all three approaches using 10% of data (316,000 randomly selected samples). The size of the dictionary K was cross-validated across $K = [100, 200, 500]$, the same range of values we evaluated for the K-means algorithm in the previous chapter. The weight parameter λ for the LASSO (see Equation 5.2) was cross-validated across $\lambda = [0.1, 1, 10]$; for both the FMS (Equation 5.3) and HMS (Equation 5.6) approaches we set $\lambda_1 = \lambda_2 = \lambda$ and cross-validated its value across $\lambda = [0.1, 1, 10]$. We use the online learning method to solve the optimization problem, using the SPAMS library [69].

After the dictionaries are trained, we performed feature coding by solving Equations 5.2 (for LASSO), Equation 5.3 (for FMS), and Equation 5.6 (our HMS) with the dictionary $\mathbf{D}$ fixed, again using the SPAMS library [69].

	Accuracy						F1 Score					
	O	C	E	A	N	Avg.	O	C	E	A	N	Avg.
Raw	0.63	0.52	0.55	0.57	0.53	0.56	0.68	0.61	0.57	0.69	0.48	0.61
LASSO [112]	0.63	0.53	0.65	0.61	0.52	0.59	0.68	0.69	0.66	0.72	0.42	0.64
FMS [54]	0.62	0.53	0.63	0.62	0.49	0.58	0.70	0.69	0.68	0.73	0.45	0.65
HMS (Ours)	**0.64**	**0.55**	**0.65**	**0.62**	**0.54**	**0.60**	**0.71**	**0.69**	**0.68**	**0.74**	**0.48**	**0.66**

Table 5.1: Mean accuracy and mean F1 score results on the Time10Q dataset. Bold faced values indicate the best method for predicting each personality dimension. Our approach (HMS) outperforms all the others on every measure.

5.3.3 Methodology

In order to ensure that the experimental results are comparable to the results from our previous chapter, we followed the same experimental methodology using the Support Vector Machine (SVM) [113] with the RBF kernel as our prediction model. The RBF kernel width was fixed to one over feature dimension, which has empirically shown to produce good performance [20]. We cross-validated the penalty parameter of the error term $C = 10^c$ with $c = [1 \cdots 6]$.

We performed 10 cross-validation, with the same set of data splits as in Chapter 4. The total number of test cases was 8,100: 3 (three dictionary learning approaches) $\times$ 10 (number of folds) $\times$ 3 (dictionary size K) $\times$ 3 (sparse code weight term λ) $\times$ 6 (SVM cost term C) $\times$ 5 (five traits – OCEAN).

5.3.4 Results and Discussion

Table 5.1 shows the recognition performance across all five personality dimensions, compared against Raw (results are from the previous chapter), LASSO [112], FMS [54], and our hierarchical multimodal subspace learning approach. Figure 5-4 shows bar plots of F1 scores. Our approach (HMS) outperformed all the other approaches on every dimension both in terms of mean accuracy and mean F1 score, but the differences were statistically not significant.

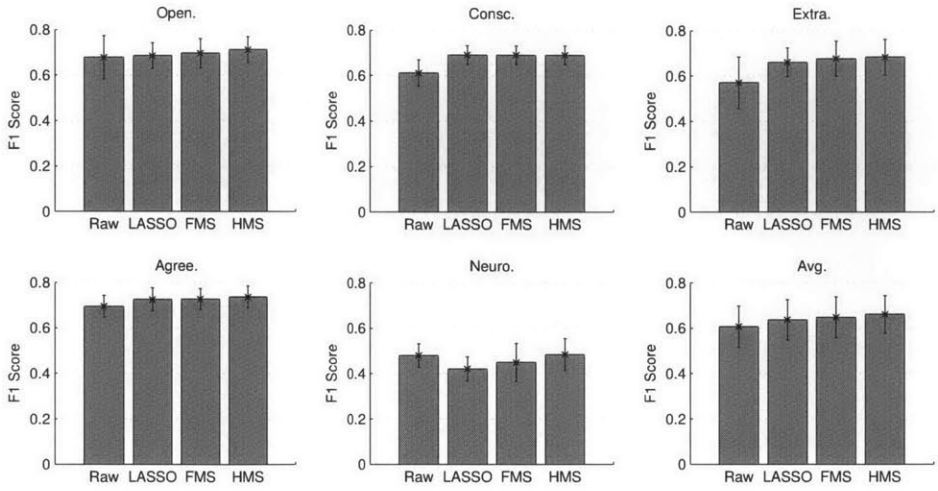


Figure 5-4: F1 score results across all five personality dimensions and an average of the five dimensions on the Time10Q dataset.

Figure 5-5 shows a visualization of the learned dictionary split into four modalities, which lets us visually confirm that the learned dictionary contains basis vectors that are shared across multiple modalities and basis vectors private to a single modality. Note that most basis vectors for the prosody and utterance modalities are zero. We believe that this is due to the lower-dimensional observational features of these two modalities (the prosody feature is 3-dimensional and the utterance feature is 50-dimensional), compared to the body (680-dimensional) and face (396-dimensional), which makes their contribution to the reconstruction error term relatively smaller (see Equation 5.6).

Figure 5-6 shows a hierarchical tree structure of the learned dictionary shown in Figure 5-5. Note that there are two nodes with the same subspace label $\mathbf{D}^{(1,2)}$, one directly under the root node, another under a subspace $\mathbf{D}^{(1,2,4)}$. This is due to the tree building procedure we explained above. When a node is empty (no basis vector for the corresponding subspace), we remove the node and connect the node's descendants ($\mathbf{D}^{(1,2)}$,$\mathbf{D}^{(1,3)}$, and $\mathbf{D}^{(2,3)}$) to the node's parent ($\mathbf{D}^{(1,2,3,4)}$), instead of combining the node's descendants with other nodes with the same subspace but under different non-empty parent node. As discussed, we believe that this makes the basis vectors contextualized with respect to the location in

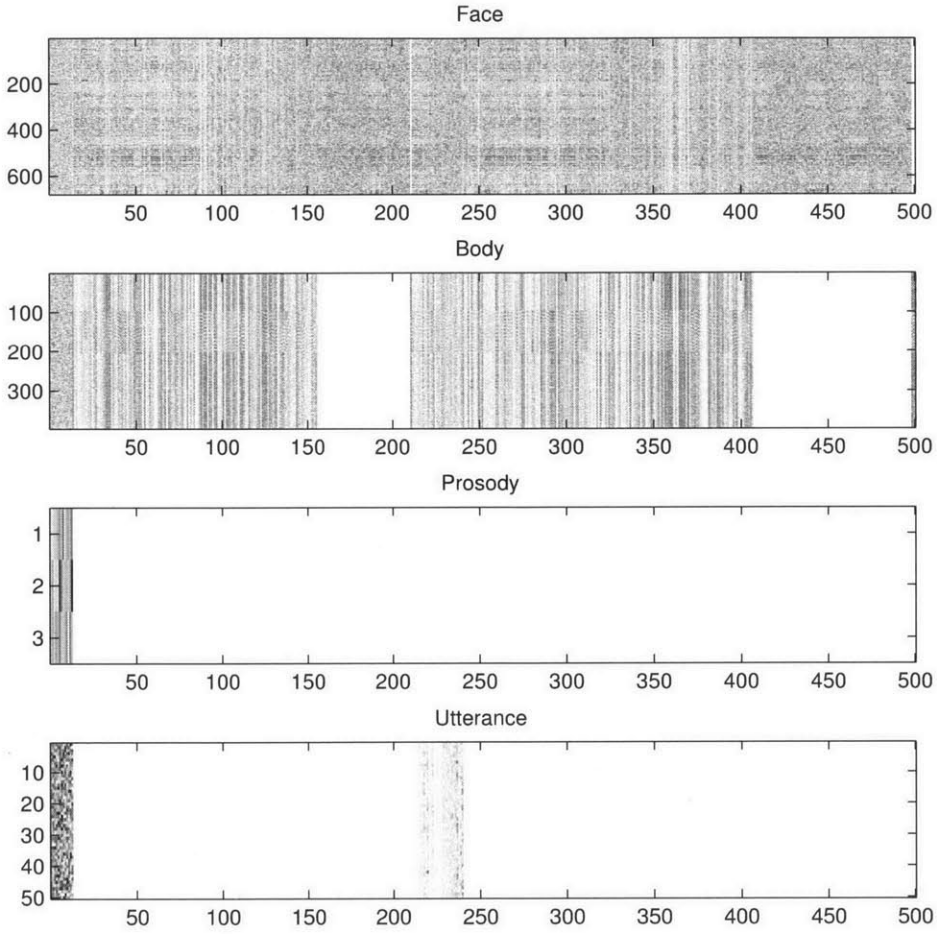


Figure 5-5: A visualization of the learned dictionary divided into four modalities. White columns indicate the basis vector has all-zero values. We can see that the dictionary contains basis vectors factorized into modality-private/shared subspaces, e.g., the ones in column 1∼13 are shared across all four modalities, the ones in column 14∼156 are shared between the two modalities (face and body), while the ones in column 157∼210 are private to a single modality (face).

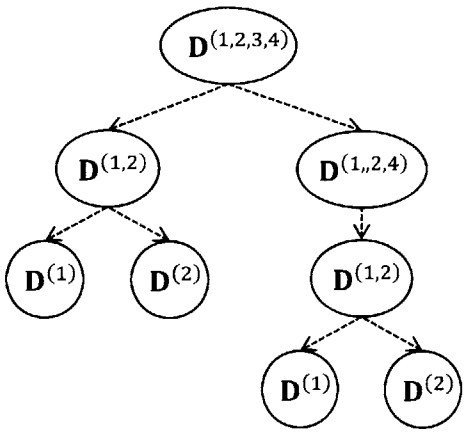


Figure 5-6: A hierarchical tree structure of the learned dictionary shown in Figure 5-5. Superscript numbers indicate modalities: (1) face, (2) body motion, (3) prosody, (4) and utterance.

the tree, allowing the sparse coefficients to have more discriminatory power, improving recognition performance in subsequence analyses.

Chapter 6

Learning Correlation and Interaction Across Modalities

This chapter describes a framework for learning correlation and interaction across modalities for multimodal sentiment analysis. Our framework is based on Canonical Correlation Analysis [43] (CCA) and Hidden Conditional Random Fields [86] (HCRFs): CCA is used to find a projection of multimodal signal that maximizes correlation across modalities, while a multi-chain structured HCRF is used to learn interaction across modalities. The multi-chain structured HCRF incorporates disjoint sets of latent variables, one set per modality, to jointly learn both modality-shared and modality-private substructures in the data. We evaluated our approach on sentiment analysis (agreement-disagreement classification) from non-verbal audio-visual cues based on the Canal 9 dataset [114]. Experimental results show that CCA makes capturing non-linear hidden dynamics easier, while a multi-chain HCRF helps learning interaction across modalities.

6.1 Task and Motivation

6.1.1 Agreement and Disagreement Recognition

In the previous two chapters, we used the Time10Q dataset for personality recognition as our application scenario. This chapter uses the Canal 9 dataset,[1] where the task is to recognize agreement and disagreement from non-verbal audio-visual cues during spontaneous political debates [114].

The Canal 9 dataset is a collection of 72 political debates produced by the Canal 9 TV station in Switzerland, with a total of roughly 42 hours of recordings. In each debate there is a moderator and two groups of participants who argue about a controversial political question. The dataset contains highly spontaneous verbal and non-verbal multimodal human behaviors.

We used a subset of the dataset used by Bousmalis *et al.* [16], which includes 11 debates segmented into 53 sequences of agreement and 94 sequences of disagreement. Bousmalis *et al.*selected the episodes based on two criteria: (a) the verbal content clearly indicates agreement/disagreement, which ensures that the ground truth label for each segment is known and evident; (b) each segment includes a single speaker with a close-up shot of the upper body, which ensures that the extracted audio-visual features represent the speaker's behavior.

We use the extracted audio-visual features provided by Bousmalis *et al.* [16]. The audio features include 2-dimensional prosodic features: the fundamental frequency (F0) and the energy. The visual features include 8 canonical body gestures: *'Head Nod', 'Head Shake', 'Forefinger Raise', 'Forefinger Raise-Like', 'Forefinger Wag', 'Hand Wag', 'Hands Scissor', and 'Shoulder Shrug'*, where the presence/absence of the gestures in each frame was manually annotated into a 8-bit vector.

[1]The work described in this chapter was done before the Time10Q dataset was collected.

6.1.2 Correlation and Interaction between Audio-Visual Signals

We believe that human behavioral signals of the sort similar to the Canal 9 data contains information that correlates between modalities and interacts over time. For example, when someone is angry the voice tends to get loud and the gestures are exaggerated. When learning with this type of data, we believe that it is important to consider the correlation and interaction patterns across modalities.

In this chapter, we show that we can improve recognition performance on sentiment analysis by projecting the original data to be maximally correlated across modalities, and by capturing the interaction across modalities explicitly from the projected data. We present a novel approach to multimodal sentiment analysis that captures non-linear correlations and interactions across modalities, based on Kernel CCA (KCCA) [43] and a Multimodal HCRF (MM-HCRF) we developed.

Our approach uses a non-linear kernel to map a multimodal signal to a high-dimensional feature space and finds a projection of the signal that maximizes the correlation across modalities. Figure 6-1(b) shows the projected signals found by KCCA, where the relative importance of gestures 'head shake' and 'shoulder shrug' have been emphasized to make the statistical relationship (correlation) between the audio and visual signals become as clear as possible. We then capture the interaction across modalities using a multi-chain structured HCRF. The model uses disjoint sets of latent variables, one set per modality, and jointly learns both modality-shared and modality-private substructures of the projected data.

6.2 Our Approach

Consider a dataset of N labeled sequences $\mathcal{D} = \{(\mathbf{x}_i, y_i) | \mathbf{x}_i \in \mathbb{R}^{d \times T_i}, y \in \mathcal{Y}\}_{i=1}^{N}$ where $\mathbf{x}_i$ is a multivariate observational sequence of length T_i and y_i is a categorical label. Since we have audio-visual data, we use the notation $\mathbf{x}_i = (\mathbf{a}_i, \mathbf{v}_i)$ where $\mathbf{a}_i \in \mathbb{R}^{d_a \times T_i}$ is the audio feature sequence $\mathbf{v}_i \in \mathbb{R}^{d_v \times T_i}$ is the visual feature sequence. The sequences in a dataset

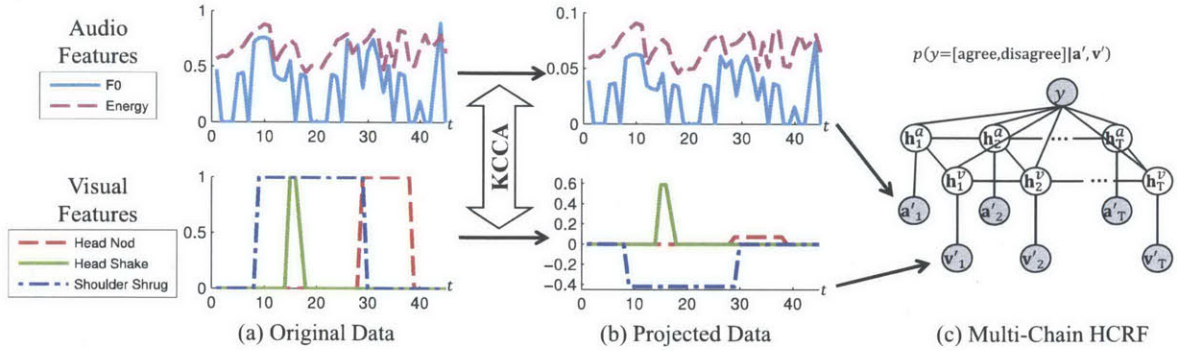


Figure 6-1: An overview of our approach. (a) An audio-visual observation sequence from the Canal 9 dataset [16]. KCCA uses a non-linear kernel to map the original data to a high-dimensional feature space, and finds a new projection of the data in the feature space that maximizes the correlation between audio and visual channels. (b) The projected data shows that emphasizing the amplitude of the 'head shake' and 'shoulder shrug' gestures maximized the correlation between audio and visual channels. (c) MM-HCRF for jointly learning both modality-shared and modality-private substructures of the projected data. $\mathbf{a}_t$ and $\mathbf{v}_t$ are observation variables from audio and visual channels, and $\mathbf{h}_t^a$ and $\mathbf{h}_t^v$ are latent variables for audio and visual channels.

have variable lengths T_i.

Figure 6-1 shows an overview of our approach. We first find a new projection of the original input sequence using KCCA [43] such that the correlation between audio and visual signals is maximized in the projected space. The projected signal is then fed into a MM-HCRF to learn interaction between audio and visual signals.

6.2.1 Kernel CCA

Given a set of paired samples $\{\mathbf{x}_i = (\mathbf{a}_i, \mathbf{v}_i)\}_{i=1}^N$, written as $\mathbf{A} = [\mathbf{a}_1, \cdots, \mathbf{a}_N]$ and $\mathbf{V} = [\mathbf{v}_1, \cdots, \mathbf{v}_N]$, Canonical Correlation Analysis (CCA) aims to find a pair of transformations $\mathbf{w}_a$ and $\mathbf{w}_v$ such that the correlation between the corresponding projections $\rho(\mathbf{w}_a^\top \mathbf{A}, \mathbf{w}_v^\top \mathbf{V})$ is maximized. However, since CCA finds $\mathbf{w}_a$ and $\mathbf{w}_v$ that are linear in the vector space, it may not reveal non-linear relationships in the data [91].

Kernel CCA (KCCA) [43] uses the kernel trick [91] to overcome this limitation by projecting

the original data onto a high-dimensional feature space before running CCA. A kernel is a function $K(\mathbf{x}_i, \mathbf{x}_j)$ that, for all $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}$,

$$K(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle$$

where $\langle \cdot, \cdot \rangle$ denotes an inner product, and Φ is a non-linear mapping function to a Hilbert space $\mathcal{F}$, $\Phi : \mathbf{x} \in \mathbb{R} \mapsto \Phi(\mathbf{x}) \in \mathcal{F}$.

To apply the kernel trick, KCCA rewrites $\mathbf{w}_a$ (and $\mathbf{w}_v$) as a product of the data $\mathbf{A}$ (and $\mathbf{V}$) with a direction α (and β),

$$\mathbf{w}_a = \mathbf{A}^\top \alpha, \quad \mathbf{w}_v = \mathbf{V}^\top \beta \tag{6.1}$$

If we assume that $\mathbf{a}$'s and $\mathbf{v}$'s are centered (i.e., mean zero), the goal is to maximize the correlation coefficient

$$\begin{aligned}
\rho(\cdot, \cdot) &= \max_{\mathbf{w}_a, \mathbf{w}_v} \frac{\mathbb{E}[\mathbf{w}_a^\top \mathbf{a} \mathbf{v}^\top \mathbf{w}_v]}{\sqrt{\mathbb{E}[\mathbf{w}_a^\top \mathbf{a} \mathbf{a}^\top \mathbf{w}_a]} \sqrt{\mathbb{E}[\mathbf{w}_v^\top \mathbf{v} \mathbf{v}^\top \mathbf{w}_v]}} \\
&= \max_{\mathbf{w}_a, \mathbf{w}_v} \frac{\mathbf{w}_a^\top \mathbf{A} \mathbf{V}^\top \mathbf{w}_v}{\sqrt{\mathbf{w}_a^\top \mathbf{A} \mathbf{A}^\top \mathbf{w}_a \mathbf{w}_v^\top \mathbf{V} \mathbf{V}^\top \mathbf{w}_v}} \\
&= \max_{\alpha, \beta} \frac{\alpha \mathbf{A} \mathbf{A}^\top \mathbf{V} \mathbf{V}^\top \beta}{\sqrt{\alpha \mathbf{A} \mathbf{A}^\top \mathbf{A} \mathbf{A}^\top \alpha \cdot \beta \mathbf{V} \mathbf{V}^\top \mathbf{V} \mathbf{V}^\top \beta}} \\
&= \max_{\alpha, \beta} \frac{\alpha^\top K_a K_v \beta}{\sqrt{\alpha^\top K_a^2 \alpha \cdot \beta^\top K_v^2 \beta}}.
\end{aligned} \tag{6.2}$$

where $K_a = K(\mathbf{A}, \mathbf{A})$ and $K_v = K(\mathbf{V}, \mathbf{V})$ are kernel matrices.

Since Equation 6.2 is scale invariant with respect to α and β (they cancel out), the optimization problem is equivalent to:

$$\max_{\alpha, \beta} \alpha^\top K_a K_v \beta \quad \text{subject to} \quad \alpha^\top K_a^2 \alpha = \beta^\top K_v^2 \beta = 1 \tag{6.3}$$

The corresponding Lagrangian dual form is

$$L(\alpha, \beta, \theta) = \alpha^\top K_a K_v \beta - \frac{\theta_\alpha}{2}(\alpha^\top K_a^2 \alpha - 1) - \frac{\theta_\beta}{2}(\beta^\top K_v^2 \beta - 1) \tag{6.4}$$

The solution to Equation 6.3 is found by taking derivatives of Equation 6.4 with respect to α and β, and solving a standard eigenvalue problem [80]. However, when K_a and K_v are non-invertible, as is common in practice with large datasets, problems can arise such as computational issues or degeneracy. This problem is solved by applying the partial Gram-Schmidt orthogonalization (PGSO) with a precision parameter η to reduce the dimensionality of the kernel matrices and approximate the correlation [43].

After we find α and β, we plug the solution back in to Equation 6.1 to obtain $\mathbf{w}_a$ and $\mathbf{w}_v$, and finally obtain new projections:

$$\mathbf{A}' = [\langle \mathbf{w}_a, \mathbf{a}_1 \rangle, \cdots, \langle \mathbf{w}_a, \mathbf{a}_N \rangle], \quad \mathbf{V}' = [\langle \mathbf{w}_v, \mathbf{v}_1 \rangle, \cdots, \langle \mathbf{w}_v, \mathbf{v}_N \rangle] \tag{6.5}$$

6.2.2 Multimodal HCRF

Given the new projection's audio-visual features $\mathbf{A}'$ and $\mathbf{V}'$ (Equation 6.5), the next step is to learn the hidden dynamics and interaction across modalities (see Figure 6-1 (b) and (c)).

We developed a Multimodal Hidden Conditional Random Field [102] (MM-HCRF), a conditional probability distribution that factorizes according to a multi-chain structured undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each chain is a discrete representation of each modality. We use disjoint sets of latent variables $\mathbf{h}^a \in \mathcal{H}^a$ for audio and $\mathbf{h}^v \in \mathcal{H}^v$ for visual to learn both modality-shared and modality-specific substructures in the data. An MM-HCRF defines $p(y \mid \mathbf{a}', \mathbf{v}')$ as

$$p(y \mid \mathbf{a}', \mathbf{v}') = \frac{\sum_{\mathbf{h}} \exp\{\Lambda^\intercal \Phi(y, \mathbf{h}, \mathbf{a}', \mathbf{v}')\}}{\sum_{y', \mathbf{h}} \exp\{\Lambda^\intercal \Phi(y', \mathbf{h}, \mathbf{a}', \mathbf{v}')\}} \tag{6.6}$$

where $\mathbf{h} = \{\mathbf{h}^a, \mathbf{h}^v\}$ and $\Lambda = [\lambda, \omega]$ is a model parameter vector. The function $\Lambda^\intercal \Phi(y, \mathbf{h}, \mathbf{a}', \mathbf{v}')$

is factorized with feature functions $f_k(\cdot)$ and $g_k(\cdot)$ as

$$\Lambda^{\intercal}\Phi(y, \mathbf{h}, \mathbf{a}', \mathbf{v}') = \sum_{s\in\mathcal{V}}\sum_{k}\lambda_k f_k(y, h_s, \mathbf{a}', \mathbf{v}') + \sum_{(s,t)\in\mathcal{E}}\sum_{k}\omega_k g_k(y, h_s, h_t, \mathbf{a}', \mathbf{v}'). \tag{6.7}$$

We define three types of feature functions. Let $\mathbb{1}[\cdot]$ be an indicator function, and $y' \in \mathcal{Y}$ and $(h', h'') \in \mathcal{H}$ be the assignments to the label and latent variables, respectively. The *label* feature function $f_k(y, h_s) = \mathbb{1}[y = y']\mathbb{1}[h_s = h']$ encodes the relationship between a latent variable h_s and a label y. The *observation* feature function $f_k(h_s, \mathbf{a}', \mathbf{v}') = \mathbb{1}[h_s]\mathbf{a}'$ or $\mathbb{1}[h_s]\mathbf{v}'$ encodes the relationship between a latent variable h_s and observations $\mathbf{x}$. The *edge* feature function $g_k(y, h_s, h_t) = \mathbb{1}[y = y']\mathbb{1}[h_s = h']\mathbb{1}[h_t = h'']$ encodes the transition between two latent variables h_s and h_t.

We use the linked topology from [102] to define the edge set $\mathcal{E}$ (shown in Figure 6-1(c)), which models contemporaneous connections between audio and visual observations, i.e., the concurrent latent states in the audio and visual channel mutually affect each other. Note that the $f_k(\cdot)$ are modeled under the assumption that modalities are conditionally independent given respective sets of latent variables, and thus encode the modality-specific substructures. The feature function $g_k(\cdot)$ encodes both modality-shared and modality-specific substructures.

The optimal parameter set Λ^* is found by minimizing the negative conditional log-probability

$$\min_{\Lambda} L(\Lambda) = \frac{1}{2\sigma^2}\|\Lambda\|^2 - \sum_{i=1}^{N}\log p(y_i \mid \mathbf{a}'_i, \mathbf{v}'_i;\ \Lambda) \tag{6.8}$$

where the first term is the Gaussian prior over Λ that works as an L_2-norm regularization. We find the optimal parameters Λ^* using gradient descent with a quasi-newton optimization method, the limited-memory BFGS algorithm [80]. The marginal probability of each node is obtained by performing an inference task using the Junction Tree algorithm [26].

6.3 Experiment

6.3.1 Methodology

The first step in our approach is to run KCCA to obtain a new projection of the data. We used a Gaussian RBF kernel as our kernel function $K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\left\|\mathbf{x}_i - \mathbf{x}_j\right\| / 2\gamma^2\right)$ because of its empirical success in the literature [91]. We validated the kernel width $\gamma = 10^k, k = [-1, 0, 1]$ and the PGSO precision parameter $\eta = [1 : 6]$ using grid search. The optimal parameter values were chosen based on the maximum correlation coefficient value.

Our experiments followed a leave-two-debates-out cross-validation approach, where we selected 2 debates of the 11 debates as the test split, 3 debates for the validation split, and the remaining 6 debates for the training split. This was repeated five times on the 11 debates. The F1 scores were averaged to get the final result. We chose four baseline models: Hidden Markov Models (HMM) [87], Conditional Random Fields (CRF) [63], Hidden Conditional Random Fields (HCRF) [86], and Multimodal HCRF (MM-HCRF) [102]. We compared this to our KCCA with MM-HCRF approach. Note that HMM and CRF perform per-frame classification, while HCRF and MM-HCRF perform per-sequence classification. The classification results of each model in turn were measured accordingly.

We automatically validated the hyper-parameters of all models. For all CRF-family models, we varied the L_2-norm regularization factor $\sigma = 10^k, k = [0, 1, 2]$ (see Equation 6.8). For HMM and HCRF, the number of hidden states were varied $|\mathcal{H}| = [2 : 6]$; for MV-HCRF, they were $\left(|\mathcal{H}^A|, |\mathcal{H}^V|\right) = ([2 : 4], [2 : 4])$. Since the optimization problems in HMM, HCRF and MV-HCRF are non-convex, we performed two random initializations of each model; the best model was selected based on the F1 score on the validation split. The L-BFGS solver was set to terminate after 500 iterations.

Models	Original Data	CCA	KCCA
HMM	.59 (.09)	.59 (.12)	.61 (.13)
CRF	.61 (.04)	.63 (.03)	.67 (.08)
HCRF	.64 (.13)	.65 (.07)	.69 (.06)
MM-HCRF	**.68 (.13)**	**.71 (.07)**	**.72 (.07)**

Table 6.1: Experimental results (means and standard deviations of F1 scores) comparing KCCA to CCA and the original data. The results show that learning nonlinear correlation in the data was important in our task.

Models	Audio	Video	Audio+Video
HMM	.54 (.08)	.58 (.11)	**.59 (.09)**
CRF	.48 (.05)	.58 (.15)	**.61 (.04)**
HCRF	.52 (.09)	.60 (.09)	**.64 (.13)**
MM-HCRF	·	·	**.68 (.13)**
KCCA + MM-HCRF	·	·	**.72 (.07)**

Table 6.2: Experimental results (means and standard deviations of F1 scores) comparing unimodal (audio or video) features to the audio-visual features. The results confirms that using both audio and visual features are important in our task.

6.3.2 Result and Discussion

We first compared our approach to existing methods: HMM [87], CRF [63], HCRF [86], and MM-HCRF. Figure 6-2 shows a bar plot of mean F1 scores and their standard deviations. We can see that our approach achieves a higher F1 score (.72) than four baseline methods.

For further analysis, we investigated whether learning nonlinear correlation was important, comparing KCCA to CCA and the original data. Table 6.1 shows that models trained with KCCA always outperformed the others, suggesting that learning non-linear correlation in the data was important. Figure 6-1(b) shows the data projected in a new space found by KCCA, where the 'head shake' and 'shoulder shrug' gestures were relatively emphasized compared to 'head nod', which maximized the correlation between the audio and visual signals. We believe that this made our data more descriptive, allowing the learning algorithm to capture the hidden dynamics and interactions between modalities more effectively.

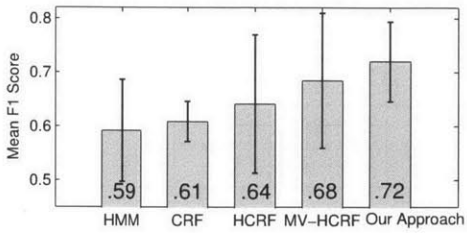


Figure 6-2: A bar plot of mean F1 scores with error bars showing standard deviations. This shows empirically that our approach successfully learned correlations and interactions between audio and visual features using KCCA and MM-HCRF.

We also investigated whether our approach captures interaction between audio-visual signals successfully. We compared the models trained with a unimodal feature (audio or visual) to the models trained with audio-visual features. Table 6.2 shows means and standard deviations of the F1 scores. In the three single-chain models, HMM, CRF, and HCRF, there was an improvement when both audio and visual features were used, confirming that using a combination of audio and visual features for our task is indeed important. Also, MV-HCRF outperformed HMM, CRF, HCRF, showing empirically that learning interaction between audio-visual signals explicitly improved the performance.

6.4 Related Work

Due to its theoretical and practical importance, multimodal human behavior analysis has been a popular research topic. While audio-visual speech recognition is probably the most well known and successful example, multimodal affect recognition has recently been getting considerable attention [129]. Bousmalis *et al.* [16] proposed a system for spontaneous agreement and disagreement recognition based only on prosodic and gesture cues, as we did here. They used an HCRF to capture the hidden dynamics of the multimodal cues. However, their approach did not consider the correlation and interaction across modalities explicitly.

Canonical correlation analysis (CCA) has been successfully applied to multimedia content analysis [43, 130]. Hardoon *et al.* [43] used kernel CCA (KCCA) for learning the semantic representation of images and their associated text. However, their approach did not consider capturing hidden dynamics in the data. Latent variable discriminative models, e.g., HCRF [86], have shown promising results in human behavior analysis tasks, for their ability to capture the hidden dynamics (e.g., spatio-temporal dynamics). Recently, we showed that the multi-modal counterpart [102] gives a significant improvement over single-view methods in recognizing human actions. However, our previous work did not learn non-linear correlation across modalities. We extend this body of work, enabling it to modeling multimodal human behavior analysis.

Chapter 7

Conclusion and Future Work

This thesis presented a collection of work on video content analysis. The focus has been two-fold: learning spatio-temporal structure of human action (Chapter 3) and learning multimodal structure of human multimodal behavior (Chapter 5 and Chapter 6). Under each focus, we presented novel application scenarios for the algorithms: aircraft handling signals recognition (Chapter 2) and personality impression recognition (Chapter 4). This chapter summarizes our contributions and discuss future direction of our research.

7.1 Summary of Contributions

Aircraft Handling Signals Recognition

Chapter 2 introduced the task of recognizing aircraft handling signals from body and hand movements, and presented the NATOPS dataset that contains the vocabulary of visual signals used by the US Navy. Unlike other datasets in the human action and gesture recognition literature, our dataset is a unique in that information about both body poses and hand shapes are required to classify visual signals correctly. We outlined the background of the project, described the data collection procedure, and explained our body

and hand tracking procedure.

Learning Spatio-Temporal Structure of Human Action

Chapter 3 presented a novel probabilistic graphical model that learns spatio-temporal structure of human action in a fine-to-coarse manner. The main contribution is in the formulation of sequence classification problem by constructing a hierarchical summary representation of video content and by learning the spatio-temporal structure from each of the summary representation. We showed that our approach achieves state-of-the-art recognition performances on three human action datasets, including a near-perfect recognition accuracy on the ArmGesture dataset [86], improving upon previous state-of-the-art results.

Personality Impression Recognition

Chapter 4 introduced the task of predicting personality impression in an interview setting, and presented the Time10Q dataset containing 866 video clips with human annotated labels. We presented the design of experiments to obtain human annotation of personality impression and showed that the labels we collected have a higher inter-rater reliability than other personality dataset in the literature. We also presented procedures for extracting multimodal features (face, body motion, utterance, and prosody of speech), and a framework for recognizing personality impression. We showed that combining the four modalities achieves the best recognition performance, achieving an F1 score of 0.61.

Learning Multimodal Structure of Human Behavior

Chapter 5 presented a novel data fusion algorithm that uses structured sparsity to combine information from multiple modalities. The main novelty is in the hierarchical formulation of multimodal subspaces via structured sparsity: we factorize the multimodal feature space

into modality-shared and modality-private subspaces, then learn a hierarchical structure among the multimodal subspaces. The learned dictionary can be seen as a collection of basis vectors embedded in a hierarchy of multimodal subspaces; using structured sparsity, we encourage the hierarchy to capture the intrinsic structure of a multimodal feature space. In essence, the hierarchical formulation allows our dictionary to capture complex dependence/independence structure across multiple modalities in a principled manner. We showed that our approach improves the performance on personality impression recognition, achieving an F1 score of 0.66 compared to using the original feature representation (0.61).

Learning Correlation and Interaction Across Modalities

Chapter 6 presented a novel framework for multimodal sentiment analysis that learns correlation and interaction across modalities. We use Kernel Canonical Correlation Analysis (KCCA) [43] to find a projection of multimodal signal that maximizes correlation across modalities. The projected signal is then fed into our novel multi-chain structured HCRF that learns interaction across modalities. The multi-chain structured HCRF incorporates disjoint sets of latent variables, one set per modality, to jointly learn both modality-shared and modality-private substructures in the data. We evaluated our approach on sentiment analysis (agreement-disagreement classification) from non-verbal audio-visual cues based on the Canal 9 dataset [114]. Experimental results show that CCA makes capturing non-linear hidden dynamics easier, while a multi-chain HCRF helps learning interaction across modalities.

7.2 Directions for Future Work

Aircraft Handling Signals Recognition

We collected the NATOPS dataset (Chapter 2) in an indoor environment under a controlled lighting condition. A stereo camera at a fixed location recorded a single person at a time,

whose location is also fixed. This simplified setting allowed us to obtain clean video data that is easier to work with.

In the real-world setting, however, things are much more complex. On an aircraft carrier deck, strong sun glare and steam produced by catapults can severely obstruct the camera view. The locations of the camera (mounted on the aircraft) and the marshallers change continuously. And there is more than one person moving in the camera view. These real-world conditions make it difficult to detect the person of interest and to track their body movements, both of which are necessary steps for recognizing aircraft handling signals.

There is much work to be done to make our framework work in the real-world setting. We believe that using contextual information can help perform the task on a carrier deck environment, such as the color of the jersey worn by the aircraft marshallers. In the US Navy, the role of each marshaller is color-coded into their jerseys: a yellow jersey is worn by aircraft handling officers, a purple jersey is worn by aviation fuels, etc. This information can help narrow down possibilities, e.g., whose gestures to follow. Other contextual information we can leverage is the routine sequence of operations. There is a kind of grammar to actions in practice; for example, once the "brakes on" action is performed, a number of other actions are effectively ruled out (e.g., "move ahead"). We look forward to exploring these possibilities in future work.

Hierarchical Sequence Summarization

Our hierarchical sequence summarization approach (Chapter 3) is designed to work on a sequence classification task where each sequence is associated with a categorical label. In other words, our approach assumes that the sequence boundaries are known a priori. In many real-world action recognition scenarios, however, action boundaries are unknown and should be inferred as a part of the recognition process. We would like to extend our approach to work in this continuous setting. One possible way is to use a sliding window: Our previous work [101] has explored one way to perform continuous sequence labeling and

segmentation by using a sliding window and multi-layered filtering. But that approach did not explore the hierarchical structure of video. We look forward to extend our hierarchical approach to enable continuous action recognition.

Personality Impression Recognition

We posed the problem of recognizing personality impression as a binary classification task (Chapter 4): whether someone has a personality trait that is below or above an average of the ones in a given population. However, the Time10Q dataset contains real-valued labels on every trait ranging from -4 to +4, which is more informative than binary labels. In the future, we would like to explore other ways to formulate the problem, such as regression (predicting the real-valued labels directly) and relative ranking (comparing the relative intensity of someone's personality trait compared to the others).

Data Fusion with Structured Sparsity

This thesis has focused on data fusion to improve the discriminatory power of multimodal signals. In the future, we would like to work on reconstruction of the original video using the learned dictionary. In particular, we want to explore ways to modify video contents such that a certain personality trait is more or less pronounced (e.g., modify the appearance of Woody Allen's face to be less neurotic). Recently, , inspired by the work in human memory from the cognitive science literature [115], Khosla *et al.* [57] presented an approach to modify the memorability of individual face photographs based on Active Appearance Models (AAMs) [24] and Histograms of Oriented Gradients (HOG) [27]. We would like to achieve similar effects in the personality domain.

Several questions need to be answered. What makes a person appear as an introvert? How can we modify the appearance of the face, body motion, and the prosody of speech so that they are natural to human eyes? This would make very interesting future work.

Bibliography

[1] J. K. Aggarwal and M. S. Ryoo. Human activity analysis: A review. *ACM Comput. Surv.*, 43(3):16, 2011.

[2] Nalini Ambady and Robert Rosenthal. Thin slices of expressive behavior as predictors of interpersonal consequences: A meta-analysis. *Psychological bulletin*, 111(2):256, 1992.

[3] Oya Aran and Daniel Gatica-Perez. One of a kind: inferring personality impressions in meetings. In *ICMI*, pages 11–18, 2013.

[4] Francis Bach, Rodolphe Jenatton, Julien Mairal, Guillaume Obozinski, et al. Structured sparsity through convex optimization. *Statistical Science*, 27(4):450 468, 2012.

[5] A.H. Barr. Superquadrics and angle-preserving transformations. *IEEE Computer Graphics and Applications*, 1(1):11–23, 1981.

[6] Murray R Barrick and Michael K Mount. The big five personality dimensions and job performance: a meta-analysis. *Personnel psychology*, 44(1):1–26, 1991.

[7] Samy Bengio, Fernando C. N. Pereira, Yoram Singer, and Dennis Strelow. Group sparse coding. In *NIPS*, pages 82 89, 2009.

[8] Yoshua Bengio. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2(1):1–127, 2009.

[9] Joan-Isaac Biel and Daniel Gatica-Perez. The youtube lens: Crowdsourced personality impressions and audiovisual analysis of vlogs. *IEEE Transactions on Multimedia*, 15(1):41–55, 2013.

[10] Ray L Birdwhistell. *Kinesics and context: Essays on body motion communication.* University of Pennsylvania press, 2011.

[11] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[12] Avrim Blum and Tom M. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT)*, pages 92–100, Madison, WI, July 1998.

[13] Aaron F. Bobick and James W. Davis. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):257–267, 2001.

[14] Anna Bosch, Andrew Zisserman, and Xavier Muñoz. Representing shape with a spatial pyramid kernel. In *CIVR*, pages 401–408, 2007.

[15] Y-Lan Boureau, Jean Ponce, and Yann LeCun. A theoretical analysis of feature pooling in visual recognition. In *ICML*, pages 111–118, 2010.

[16] Konstantinos Bousmalis, Louis-Philippe Morency, and Maja Pantic. Modeling hidden dynamics of multimodal cues for spontaneous agreement and disagreement recognition. In *Proceedings of the 9th IEEE Conference on Automatic Face and Gesture Recognition (FG)*, Santa Barbara, CA, March 2011.

[17] Konstantinos Bousmalis, Louis-Philippe Morency, and Maja Pantic. Modeling hidden dynamics of multimodal cues for spontaneous agreement and disagreement recognition. In *FG*, 2011.

[18] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library.* O'Reilly, Cambridge, MA, 2008.

[19] Raymond B Cattell. The scree test for the number of factors. *Multivariate behavioral research*, 1(2):245–276, 1966.

[20] Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM TIST*, 2(3):27, 2011.

[21] Yu-Tseh Chi, Mohsen Ali, Ajit Rajwade, and Jeffrey Ho. Block and group regularized sparse modeling for dictionary learning. In *CVPR*, pages 377–382, 2013.

[22] Chris Mario Christoudias, Raquel Urtasun, and Trevor Darrell. Multi-view learning in the presence of view disagreement. *CoRR*, abs/1206.3242, 2012.

[23] Adam Coates and Andrew Y. Ng. The importance of encoding versus training with sparse coding and vector quantization. In *ICML*, pages 921–928, 2011.

[24] Timothy F. Cootes, Gareth J. Edwards, and Christopher J. Taylor. Active appearance models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(6):681–685, 2001.

[25] PT Costa Jr and Robert R McCrae. Neo personality inventory–revised (neo-pi-r) and neo five-factor inventory (neo-ffi) professional manual. *Odessa, FL: Psychological Assessment Resources*, 1992.

[26] Robert G. Cowell, A. Philip Dawid, Steffen L. Lauritzen, and David J. Spiegelhalter. *Probabilistic Networks and Expert Systems*. Springer-Verlag, 1999.

[27] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893, 2005.

[28] J. Denavit and R. S. Hartenberg. A kinematic notation for lower-pair mechanisms based on matrices. *ASME Journal of Applied Mechanisms*, 23:215–221, 1955.

[29] Jonathan Deutscher, Andrew Blake, and Ian D. Reid. Articulated body motion capture by annealed particle filtering. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2126–2133, 2000.

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

[31] Paul Ekman and Erika L Rosenberg. *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS).* Oxford University Press, 1997.

[32] Albert Ellis, Mike Abrams, and Lidia Abrams. *Personality theories: Critical perspectives.* Sage, 2009.

[33] Florian Eyben, Felix Weninger, Florian Gross, and Björn Schuller. Recent developments in opensmile, the munich open-source multimedia feature extractor. In *ACM Multimedia*, pages 835–838, 2013.

[34] Hans J Eysenck. *The psychology of politics*, volume 2. Transaction publishers, 1954.

[35] Hans Jürgen Eysenck. *The biological basis of personality*, volume 689. Transaction publishers, 1967.

[36] Hans Jurgen Eysenck. *The Structure of Human Personality (Psychology Revivals).* Routledge, 2013.

[37] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59(2), 2004.

[38] Simon Fothergill, Helena M. Mentis, Pushmeet Kohli, and Sebastian Nowozin. Instructing people for training gestural interactive systems. In *CHI*, pages 1737–1746, 2012.

[39] William T Freeman and Michal Roth. Orientation histograms for hand gesture recognition. In *International Workshop on Automatic Face and Gesture Recognition*, volume 12, pages 296 301, 1995.

[40] Samuel D Gosling, Peter J Rentfrow, and William B Swann Jr. A very brief measure of the big-five personality domains. *Journal of Research in personality*, 37(6):504–528, 2003.

[41] Samuel D Gosling, Simine Vazire, Sanjay Srivastava, and Oliver P John. Should we trust web-based studies? a comparative analysis of six preconceptions about internet questionnaires. *American Psychologist*, 59(2):93, 2004.

[42] Thomas L Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National academy of Sciences of the United States of America*, 101(Suppl 1):5228 5235, 2004.

[43] David R. Hardoon, Sándor Szedmák, and John Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural Comp.*, 16(12):2639–2664, 2004.

[44] Jeffrey Robert Karplus Hartline. *Incremental Optimization.* PhD thesis, Cornell University, 2008.

[45] Geoffrey E. Hinton, Simon Osindero, and Yee Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527 1554, 2006.

[46] Mohammed (Ehsan) Hoque, Matthieu Courgeon, Jean-Claude Martin, Bilge Mutlu, and Rosalind W. Picard. Mach: my automated conversation coach. In *UbiComp*, pages 697–706, 2013.

[47] Weiming Hu, Tieniu Tan, Liang Wang, and Stephen J. Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 34(3):334–352, 2004.

[48] Gary B. Huang, Honglak Lee, and Erik G. Learned-Miller. Learning hierarchical representations for face verification with convolutional deep belief networks. In *CVPR*, 2012.

[49] Aapo Hyvärinen, Juha Karhunen, and Erkki Oja. *Independent component analysis*, volume 46. John Wiley & Sons, 2004.

[50] Michael Isard and Andrew Blake. CONDENSATION - conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[51] C Izard. Personality similarity and friendship. *The Journal of Abnormal and Social Psychology*, 61(1):47, 1960.

[52] Laurent Jacob, Guillaume Obozinski, and Jean-Philippe Vert. Group lasso with overlap and graph lasso. In *ICML*, page 55, 2009.

[53] Rodolphe Jenatton, Julien Mairal, Guillaume Obozinski, and Francis Bach. Proximal methods for hierarchical sparse coding. *Journal of Machine Learning Research*, 12:2297–2334, 2011.

[54] Yangqing Jia, Mathieu Salzmann, and Trevor Darrell. Factorized latent spaces with structured sparsity. In *NIPS*, pages 982–990, 2010.

[55] Yangqing Jia, Oriol Vinyals, and Trevor Darrell. On compact codes for spatially pooled features. In *ICML*, pages 549–557, 2013.

[56] Zerrin Kasap and Nadia Magnenat-Thalmann. Intelligent virtual humans with autonomy and personality: State-of-the-art. *Intelligent Decision Technologies*, 1(1-2):3–15, 2007.

[57] Aditya Khosla, Wilma A. Bainbridge, Antonio Torralba, and Aude Oliva. Modifying the memorability of face photographs. In *ICCV*, pages 3200–3207, 2013.

[58] Kyungnam Kim, Thanarat H. Chalidabhongse, David Harwood, and Larry S. Davis. Real-time foreground-background segmentation using codebook model. *Real-Time Imaging*, 11(3):172–185, 2005.

[59] Aniket Kittur, Ed H. Chi, and Bongwon Suh. Crowdsourcing user studies with mechanical turk. In *CHI*, pages 453–456, 2008.

[60] Teddy Ko. A survey on behavior analysis in video surveillance for homeland security applications. In *AIPR*, pages 1–8, 2008.

[61] Adriana Kovashka and Kristen Grauman. Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In *CVPR*, 2010.

[62] Mark-A. Krogel and Tobias Scheffer. Multi-relational learning, text mining, and semi-supervised learning for functional genomics. *Machine Learning*, 57(1-2):61 81, 2004.

[63] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning (ICML)*, pages 282–289, Williamstown, MA, July 2001.

[64] Gert R. G. Lanckriet, Nello Cristianini, Peter L. Bartlett, Laurent El Ghaoui, and Michael I. Jordan. Learning the kernel matrix with semidefinite programming. *JMLR*, 5:27–72, 2004.

[65] Ivan Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2-3):107–123, 2005.

[66] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

[67] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, pages 2169–2178, 2006.

[68] Quoc V. Le, Will Y. Zou, Serena Y. Yeung, and Andrew Y. Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *CVPR*, 2011.

[69] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11:19–60, 2010.

[70] François Mairesse, Marilyn A. Walker, Matthias R. Mehl, and Roger K. Moore. Using linguistic cues for the automatic recognition of personality in conversation and text. *J. Artif. Intell. Res. (JAIR)*, 30:457–500, 2007.

[71] Robert R McCrae and Oliver P John. An introduction to the five-factor model and its applications. *Journal of personality*, 60(2):175–215, 1992.

[72] Daniel McDuff, Rana El Kaliouby, David Demirdjian, and Rosalind W. Picard. Predicting online media effectiveness based on smile responses gathered over the internet. In *FG*, pages 1–7, 2013.

[73] Gelareh Mohammadi and Alessandro Vinciarelli. Automatic personality perception: Prediction of trait attribution based on prosodic features. *T. Affective Computing*, 3(3):273–284, 2012.

[74] Gianluca Monaci, Philippe Jost, Pierre Vandergheynst, Boris Mailhé, Sylvain Lesage, and Rémi Gribonval. Learning multimodal dictionaries. *IEEE Transactions on Image Processing*, 16(9):2272–2283, 2007.

[75] Arthur G. Money and Harry W. Agius. Video summarisation: A conceptual framework and survey of the state of the art. *J. Visual Communication and Image Representation*, 19(2):121–143, 2008.

[76] Louis-Philippe Morency, Rada Mihalcea, and Payal Doshi. Towards multimodal sentiment analysis: harvesting opinions from the web. In *Proceedings of the 13th ACM International Conference on Multimodal Interaction (ICMI)*, pages 169–176, 2011.

[77] Louis-Philippe Morency, Ariadna Quattoni, and Trevor Darrell. Latent-dynamic discriminative models for continuous gesture recognition. In *Proceedings of the 20th*

IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Minneapolis, MN, June 2007.

[78] NASA. *Man-Systems Integration Standards: Volume 1. Section 3. Anthropometry And Biomechanics*, 1995.

[79] Juan Carlos Niebles and Li Fei-Fei. A hierarchical model of shape and appearance for human action classification. In *CVPR*, 2007.

[80] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization.* Springer-Verlag, 1999.

[81] Brian P O'Connor. A quantitative review of the comprehensiveness of the five-factor model in relation to popular personality inventories. *Assessment*, 9(2):188–203, 2002.

[82] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.

[83] Maja Pantic and Leon JM Rothkrantz. Toward an affect-sensitive multimodal human-computer interaction. *Proceedings of the IEEE*, 91(9):1370–1390, 2003.

[84] Judea Pearl. Reverend bayes on inference engines: A distributed hierarchical approach. In *AAAI National Conference on AI*, pages 133–136, 1982.

[85] Jian Peng, Liefeng Bo, and Jinbo Xu. Conditional neural fields. In *Advances in Neural Information Processing Systems 22 (NIPS)*, pages 1419–1427, Vancouver, BC, December 2009.

[86] Ariadna Quattoni, Sy Bor Wang, Louis-Philippe Morency, Michael Collins, and Trevor Darrell. Hidden conditional random fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(10):1848–1852, 2007.

[87] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, pages 257–286, 1989.

[88] Beatrice Rammstedt and Oliver P John. Measuring personality in one minute or less: A 10-item short version of the big five inventory in english and german. *Journal of Research in Personality*, 41(1):203–212, 2007.

[89] Ruslan Salakhutdinov and Geoffrey E. Hinton. An efficient learning procedure for deep boltzmann machines. *NECO*, 24(8), 2012.

[90] Edward Sapir. Speech as a personality trait. *American Journal of Sociology*, pages 892–905, 1927.

[91] Bernhard Scholkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press, Cambridge, MA, USA, 2001.

[92] Björn Schuller. Voice and speech analysis in search of states and traits. In *Computer Analysis of Human Behavior*, pages 227–253. Springer, 2011.

[93] Björn Schuller, Michel François Valstar, Roddy Cowie, and Maja Pantic. AVEC 2012: the continuous audio/visual emotion challenge - an introduction. In *ICMI*, pages 361–362, 2012.

[94] D Seung and L Lee. Algorithms for non-negative matrix factorization. *NIPS*, 2001.

[95] Jamie Shotton, Andrew W. Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-time human pose recognition in parts from single depth images. In *Proceedings of the 24th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1297–1304, Colorado Springs, June 2011.

[96] Alex Shyr, Raquel Urtasun, and Michael I. Jordan. Sufficient dimension reduction for visual sequence classification. In *Proceedings of the 23rd IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3610–3617, San Francisco, CA, June 2010.

[97] Cees Snoek, Marcel Worring, and Arnold W. M. Smeulders. Early versus late fusion in semantic video analysis. In *ACM Multimedia*, pages 399–402, 2005.

[98] Cees G. M. Snoek and Marcel Worring. Concept-based video retrieval. *Foundations and Trends in Information Retrieval*, 2(4):215 322, 2009.

[99] Yale Song, David Demirdjian, and Randall Davis. Multi-signal gesture recognition using temporal smoothing hidden conditional random fields. In *Proceedings of the 9th IEEE Conference on Automatic Face and Gesture Recognition (FG)*, pages 388–393, Santa Barbara, CA, March 2011.

[100] Yale Song, David Demirdjian, and Randall Davis. Tracking body and hands for gesture recognition: Natops aircraft handling signals database. In *Proceedings of the 9th IEEE Conference on Automatic Face and Gesture Recognition (FG)*, pages 500–506, Santa Barbara, CA, March 2011.

[101] Yale Song, David Demirdjian, and Randall Davis. Continuous body and hand gesture recognition for natural human-computer interaction. *ACM Trans. Interact. Intell. Syst.*, 2(1):5:1–5:28, March 2012.

[102] Yale Song, Louis-Philippe Morency, and Randall Davis. Multi-view latent variable discriminative models for action recognition. In *Proceedings of the 25th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Providence, RI, June 2012.

[103] Yale Song, Louis-Philippe Morency, and Randall Davis. Multimodal human behavior analysis: Learning correlation and interaction across modalities. In *Proceedings of the 14th ACM International Conference on Multimodal Interaction (ICMI)*, Santa Monica, CA, October 2012.

[104] Yale Song, Louis-Philippe Morency, and Randall Davis. Action recognition by hierarchical sequence summarization. In *CVPR*, pages 3562–3569, 2013.

[105] Yale Song, Louis-Philippe Morency, and Randall Davis. Distribution-sensitive learning for imbalanced datasets. In *FG*, pages 1–6, 2013.

[106] Yale Song, Louis-Philippe Morency, and Randall Davis. Learning a sparse codebook of facial and body microexpressions for emotion recognition. In *ICMI*, pages 237–244, 2013.

[107] Yale Song, Zhen Wen, Ching-Yung Lin, and Randall Davis. One-class conditional random fields for sequential anomaly detection. In *IJCAI*, 2013.

[108] Sören Sonnenburg, Gunnar Rätsch, Christin Schäfer, and Bernhard Schölkopf. Large scale multiple kernel learning. *JMLR*, 7:1531–1565, 2006.

[109] Ju Sun, Xiao Wu, Shuicheng Yan, Loong Fah Cheong, Tat-Seng Chua, and Jintao Li. Hierarchical spatio-temporal context modeling for action recognition. In *CVPR*, 2009.

[110] Zoltán Szabó, Barnabás Póczos, and András Lörincz. Online group-structured dictionary learning. In *CVPR*, pages 2865–2872, 2011.

[111] Kevin Tang, Fei-Fei Li, and Daphne Koller. Learning latent temporal structure for complex event detection. In *CVPR*, 2012.

[112] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[113] Vladimir N. Vapnik. *The nature of statistical learning theory.* Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[114] Alessandro Vinciarelli, Alfred Dielmann, Sarah Favre, and Hugues Salamin. Canal9: A database of political debates for analysis of social interactions. In *Proceedings of the 3rd International Conference on Affective Computing and Intelligent Interaction (ACII)*, Amsterdam, Netherlands, September 2009.

[115] John R Vokey and J Don Read. Familiarity, memorability, and the effect of typicality on the recognition of faces. *Memory & Cognition*, 20(3):291 302, 1992.

[116] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Action recognition by dense trajectories. In *CVPR*, pages 3169–3176, 2011.

[117] Jiang Wang, Zhuoyuan Chen, and Ying Wu. Action recognition with multiscale spatio-temporal contexts. In *CVPR*, 2011.

[118] Sy Bor Wang, Ariadna Quattoni, Louis-Philippe Morency, David Demirdjian, and Trevor Darrell. Hidden conditional random fields for gesture recognition. In *Proceedings of the 19th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1521–1527, 2006.

[119] Yang Wang and Greg Mori. Hidden part models for human action recognition: Probabilistic versus max margin. *PAMI*, 33(7), 2011.

[120] Martin Wöllmer, Felix Weninger, Tobias Knaup, Björn Schuller, Congkai Sun, Kenji Sagae, and Louis-Philippe Morency. Youtube movie reviews: Sentiment analysis in an audio-visual context. *IEEE Intelligent Systems*, 28(3):46–53, 2013.

[121] John Wright, Yi Ma, Julien Mairal, Guillermo Sapiro, Thomas S. Huang, and Shuicheng Yan. Sparse representation for computer vision and pattern recognition. *Proceedings of the IEEE*, 98(6):1031–1044, 2010.

[122] William F Wright and Gordon H Bower. Mood effects on subjective probability assessment. *Organizational behavior and human decision processes*, 52(2):276–291, 1992.

[123] Zhengzheng Xing, Jian Pei, and Eamonn Keogh. A brief survey on sequence classification. *SIGKDD Explor. Newsl.*, 12(1):40–48, November 2010.

[124] Xuehan Xiong and Fernando De la Torre. Supervised descent method and its applications to face alignment. In *CVPR*, pages 532–539, 2013.

[125] Chang Xu, Dacheng Tao, and Chao Xu. A survey on multi-view learning. *arXiv preprint arXiv:1304.5634*, 2013.

[126] R. S. Yadav. Interview as a means of personality assessment: Some baffling dilemmas. *Indian Journal of Psychometry & Education*, 21(2):67–79, 1990.

[127] Jianchao Yang, Kai Yu, Yihong Gong, and Thomas S. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, pages 1794–1801, 2009.

[128] Dong Yu and Li Deng. Deep-structured hidden conditional random fields for phonetic recognition. In *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 2986–2989, Makuhari, Japan, September 2010.

[129] Zhihong Zeng, Maja Pantic, Glenn I. Roisman, and Thomas S. Huang. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(1):39–58, 2009.

[130] Hong Zhang, Yueting Zhuang, and Fei Wu. Cross-modal correlation learning for clustering on image-audio dataset. In *ACM Multimedia*, pages 273–276, 2007.

[131] Yueting Zhuang, Yan Fei Wang, Fei Wu, Yin Zhang, and Weiming Lu. Supervised coupled dictionary learning with group structures for multi-modal retrieval. In *AAAI*, 2013.