Probabilistic Graphical Models for Human Interaction Analysis

presentée a la Facultè des sciences et techniques de l'ingènieur

École Polytechnique Fédérale de Lausanne

pour l'obtention du grade de docteur ès sciences

par

DONG ZHANG

Bachelor of Technology in Electrical Engineering,

Beijing Institute of Technology, Beijing, China (1998)

Master of Science in Electrical Engineering,

Chinese Academy of Sciences, Beijing, China (2001)

Thesis committee members

Dr. Daniel Gatica-Perez, IDIAP, Switzerland

Prof. Hervé Bourlard, directeur de thèse, IDIAP/EPFL , Switzerland

Prof. Jean-Philippe Thiran, EPFL, Switzerland

Prof. Rigoll Gerhard, Munich University of Technology, Germany

Prof. Rainer Stiefelhagen, Universitaet Karlsruhe, Germany

Lausanne, EPFL

2006

Abstract

The objective of this thesis is to develop probabilistic graphical models for analyzing human interaction in meetings based on multimodel cues. We use meeting as a study case of human interactions since research shows that high complexity information is mostly exchanged through face-to-face interactions.

Modeling human interaction provides several challenging research issues for the machine learning community. In meetings, each participant is a multimodal data stream. Modeling human interaction involves simultaneous recording and analysis of multiple multimodal streams. These streams may be asynchronous, have different frame rates, exhibit different stationarity properties, and carry complementary (or correlated) information.

In this thesis, we developed three probabilistic graphical models for human interaction analysis. The proposed models use the "probabilistic graphical model" formalism, a formalism that exploits the conjoined capabilities of graph theory and probability theory to build complex models out of simpler pieces.

We first introduce the multi-layer framework, in which the first layer models typical individual activity from low-level audio-visual features, and the second layer models the interactions. The two layers are linked by a set of posterior probability-based features.

Next, we describe the team-player influence model, which learns the influence of interacting Markov chains within a team. The team-player influence model has a two-level structure: individual-level and group-level. Individual level models actions of each player, and the group-level models actions of the team as a whole. The influence of each player on the team is jointly learned with the rest of the model parameters in a principled manner using the Expectation-Maximization (EM) algorithm.

Finally, we describe the semi-supervised adapted HMMs for unusual event detection. Unusual events are characterized by a number of features (rarity, unexpectedness, and relevance) that limit the application of traditional supervised model-based approaches. We propose a semi-supervised adapted Hidden Markov Model (HMM) framework, in which usual event models are first learned from a large amount of (commonly available) training data, while unusual event models are learned by Bayesian adaptation in an unsupervised manner.

Version abrégée

L'objectif de cette thèse est le développement de modèles graphiques probabilistes pour l'analyse des interactions entre personnes durant les réunions, en se basant sur des caractéristiques multimodales. Nous considérons les meetings comme base de notre recherche, car les informations complexes sont principalement échangées au travers d'interactions face-à-face.

La modélisation des interactions entre personnes représente plusieurs challenge pour la recherche en apprentissage machine. Chaque participant d'une réunion est une source de données multimodales. La modélisation des interactions entre personnes nécessite donc l'enregistrement et l'analyse simultanés de plusieurs sources de données multimodales. Ces sources peuvent tre asynchrones, avoir différents taux d'échantillonnage, présenter des propriétés stationnaires différentes, et comporter de l'information complémentaire (ou corrélée).

Dans cette thèse, nous avons développé trois modèles graphiques probabilistes pour l'analyse des interactions humaines. Les modèles proposés utilisent le formalisme des 'modèles graphiques probabilistes', qui exploite les possibilités conjointes de la théorie des graphes et des probabilités afin de créer des modèles complexes à partir de pièces plus simples.

Nous introduisons tout d'abord le système multi-couche, dont la première couche modélise des activités individuelles typiques à partir de caractéristiques audiovisuelles bas-niveau, et la seconde couche modélise les interactions. Les deux couches sont reliées par un ensemble de caractéristiques basées sur des probabilités postérieures.

Ensuite, nous décrivons le modèle de l'influence du joueur d'une équipe, qui apprend l'influence de chaînes de Markov cachées au sein d'une équipe. Le modèle a une structure à deux niveaux : une couche individuelle et une couche de groupe. La couche individuelle modélise les actions de chaque joueur, et la couche de groupe modélise les actions de l'équipe comme un tout. L'influence de

chaque joueur est apprise de manière conjointe avec le reste des paramètres du modèle en utilisant un algorithme de maximisation d'espérance.

Finalement, nous décrivons les modèles de Markov cachés adaptés de manière semi-supervisée pour la détection d'événements inhabituels. Les événements inhabituels présentent un nombre de caractéristiques spécifiques (rareté, imprévu, et pertinence) qui ne permettent pas l'application d'approches semi-supervisées traditionnelles. Nous proposons un modèle de Markov caché adapté de manière non-supervisée, où les modèles d'événements habituels sont tout d'abord entraînés à partir d'un grand nombre de données, tandis que les modèles d'événements inhabituels sont adaptés de manière Bayesienne non-supervisée.

Contents

List of Figures

List of Tables

Acknowledgements

This dissertation would not have been possible without the help and support of my supervisors, friends and family.

First and foremost, I would like to thank my advisors, Dr. Daniel Gatica-Perez and Dr. Samy Bengio. They have showed much kindness and patience towards me. For the past years, they have been a teacher, friends, mentors and collaborators. I am fortunate to have them as my life-long supervisors.

My gratitude extends to Prof. Herve Bourlard, the director of IDIAP Research Institute. Under his leadership, IDIAP has become a haven of scientific research. The stimulating research environment at IDIAP has a substantial impact on my research.

Special thanks to my colleagues: Florent, Kevin, Sileye, Mathew, David, Norman, Agnes, Guillaume, Hemant, Mael and more, who not only helped me with my work but also made my stay in Martigny a memorable one.

My comfortable life in Switzerland would not have been possible without the generous financial support from several projects sponsored by Swiss National Science Foundation (SNSF) and European Society Technologies (IST). This work was carried out in the framework of the Swiss National Center of Competence in Research (NCCR) on Interactive Multimodal Information Management (IM2). This work was also possible through support from other European projects including Augmented Multi-party Interaction (AMI), Pattern Analysis Statistical Modelling and Computational Learning (PASCAL), Multimodal Meeting Manager (M4).

Finally and most importantly, I thank all my friends and family for their support. I especially thank my parents who have given me a lifetime of love and care.

Chapter 1

Introduction

1.1 Motivations

Given the increasing capacity to acquire data from various sensors (camera, microphones, and mobile phones, etc.), together with ever cheaper, and ever increasing processing speed, storage, and bandwidth, there is a growing interest in several areas – computer vision, machine learning, social network analysis and wearable computing – to analyze human actions and interactions. **Modeling human interactions from sensor information using machine learning techniques is the goal of this thesis**.

This is the result of the convergence of some trends observed in a number of research areas.

The first trend is the growing interest of the computer vision and signal processing communities in the analysis of human actions and interactions. Recognition of human activities is important for many applications, such as surveillance [98, 53], helping people with disabilities [120], and human-machine interaction [104]. Over the last decade, a lot of work has been done on analysis of human behavior from low- or/and middle-level sensor information. These systems typically consist of a low- or mid-level computer vision system to detect and segment an object – human, for example – and a higher level interpretation module that classifies the motion into 'atomic' behaviors such as, for example, *a smile*, or *a pointing gesture*. Moving beyond the person-centered paradigm, recent work has started to explore multi-person scenarios, where not only individual but also group actions or interactions become relevant [44, 59, 99, 10].

The second trend followed by this thesis aligns with recent developments in the field of statistical machine learning, namely probabilistic graphical models. In the last decade, probabilistic graphical models have become one of the most popular tools to structure uncertain knowledge about high-dimensional domains in order to make reasoning in such domains feasible [38]. Many of the problems in artificial intelligence, statistics, computer systems, computer vision, natural language processing, and computational biology, among many other fields, can be viewed as the search for a coherent global conclusion from local information. The probabilistic graphical model framework provides a unified view for this wide range of problems, as it combines a natural mechanism for expressing contextual knowledge with the power of efficient algorithms for statistical learning and inference.

The third trend is the increasing interest in social network analysis. A social network is a social structure made of nodes which are generally individuals or organizations. Social network analysis has emerged as a key technique in modern sociology, anthropology, social psychology, information science, and organizational studies, as well as a popular topic of speculation and study. Research in a number of academic fields has demonstrated that social networks operate on many levels, from families up to the level of nations, and play a critical role in determining the way problems are solved, organizations are run, and the degree to which individuals succeed in achieving their goals. Online social networks are becoming increasingly popular, which helps connect friends, business partners, or other individuals together using a variety of tools, such as emails and instant messengers. There are all instances of human interactions.

The fourth trend is that wearable computing becomes more and more integrated into our daily life. Wearable computing aims at assisting people in various everyday activities (e.g., life saving, security, health monitoring, mobile web services) by using small devices such as cameras, microphones (e.g., recording all what one sees and all what one hears), and multiple extra sensors (e.g., recording diverse physiological signals, *etc.*.). This provides the opportunity to use appropriate sensors to capture information about how groups of people interact over period of weeks, months and even years. For example, in [25], wearable devices and methods have been designed for automatically and unobtrusively learning the structure of face-to-face interactions within groups based on wearable sensors. Another project, called *Reality Mining* [37], introduced a system to recognize social patterns in daily user activity, infer relationships, and model organizational rhythms using data collected from one hundred mobile phones over the course of six months (approximately 500,000 hours of data on users' location, communication and device usage behavior).

Finally, there is a long history of work in social sciences aiming at understanding the interactions between individuals and how interactions influence their behaviors. In the psychology community, there are many instances of work studying the above effects [83, 40]. In almost any social and work situation, our decision-making is influenced by the actions of others around us. Who are the people we talk to? For how long and how often? How actively do we participate in conversations we have? While modeling human interactions is a recent research domain, a large body of literature on group interactions exists in the field of social psychology. This literature gives valuable insight into the nature and value of information present in human interactions.

One of the goals of this thesis is to make connections and build upon previous work in these research areas.

1.2 Research Objectives

In this thesis, we have four objectives.

The first objective is to develop statistical models for probabilistic inference of group interaction patterns using multi-sensor data.

Viewed as a whole, a group in a meeting shares information, engages in discussions, and makes decisions, proceeding through diverse communication phases both in single meetings and during the course of long-term collaborative work. We attempt to structure a meeting as a continuous sequence of exclusive events taken from the set of N group actions: $V = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N\}$.

In the first place, we need to define a set of relevant group actions. Different action lexicons give us different view of meetings, as shown in Figure 1.1. In this thesis, we defined a set of group actions based on *location-based turn-taking*.

As documented in psychology literature, turn-taking provides a rich basis for analyzing how people interact in group discussions. At its simplest level, segmenting a meeting into speaker turns is useful for structuring speech transcripts for browsing and retrieval. Analysis of speaker turns can also provide insights about the participants, such as their inherent latency in responding and

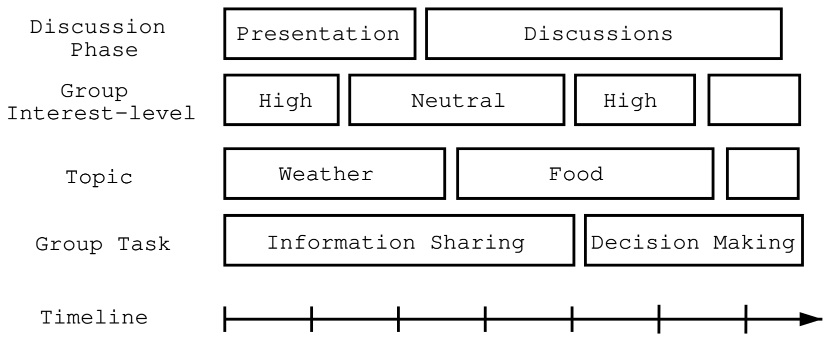


Figure 1.1. Joint segmentation and classification of sequences of human interactions in meetings.

degree of 'talkativeness', their role within a group, or their interest in particular topics [82, 101, 109].

Moving beyond simple speaker turns, turn-taking may be analyzed at a higher-level by defining actions that may span several individual speaker turns, such as distinguishing between a series of monologues and a group discussion. Turns not based purely on speech, such as presentations, white-board usage or group note-taking, could also be defined if visual cues such as gaze and gestures were taken into account.

In this thesis, an illustrative set of group actions based on location-based turn-taking was defined as { *discussion, monologue, monologue + note-taking, note-taking, presentation, presentation + note-taking, whiteboard, whiteboard + note-taking* } . These are all natural actions in which participants play and exchange similar, opposite, or complementary roles. For example, during a monologue, one person speaks to the group, while the other participants listen and direct their gaze towards the speaker or to their notes. During a discussion, multiple participants take relatively short turns at speaking, and more movement could be expected. In this set of actions, we define note-taking as a group event, in which the majority of participants take notes concurrently. Intuitively, it is expected that such an action would indicate periods where important information has been conveyed.

In a similar manner, other lexicon of group actions could be defined to provide alternative views of a meeting. While actions should be non-overlapping within a given set of meeting actions, rich multi-layer views of meetings could be built by applying parallel sets of meeting actions to the same meeting. For example further lexicon could be based on tasks (brainstorming, information sharing, decision making, etc), and the interest level of the group (high, neutral, low). Recent research in recognizing emotion from speech [73, 61], recognizing interest level from posture [90], recognizing hot-spots (regions of high involvement or emphasis) in meetings [132, 131, 66], and detecting agreement and disagreement in meetings [56], suggests that the automatic recognition of such high-level concepts may become feasible.

The second objective is to develop statistical models of automatically detecting participant's influence levels in meetings.

During the course of a meeting, some people seem particularly capable of driving the conversation and dominating its outcome. These people, skilled at establishing the leadership, have the largest influence on a meeting, and often shift its focus when they speak. Can we tell who the most influential participant is? Can we quantify this amount of influence? How does the behavior of each individual affect the group decision-making? A computational model that addresses these questions involves challenges for the following reasons,

- To build a model that can determine influence among meeting participants, we need to extract relevant features, with the assumption that influence can indeed be inferred from a set of low-level observations. In this sense, a large range of audio, visual and language features could be used. How to determine the most discriminative features is a non-trivial task.

- The task might be hard to evaluate. The manual annotation of influence of meeting participants is to some degree a subjective task since a unique ground-truth does not exist.

- To model a significant number of interacting people, a naive model may require an exponential number of parameters in the number of persons, which might make learning and inference intractable. This motivates the development of simplified models that at the same time retain representation power.

The third objective is to develop statistical models of automatically detecting unusual events in meetings.

Unusual events, such as "laughter" in meetings, are characterized by a number of features – rarity, unexpectedness, and relevance – that limit the application of traditional supervised model-based approaches. It is clear from such a definition that unusual event detection entails a number of challenges.

- The rarity of an unusual event means that collecting sufficient training data for supervised learning will often be infeasible, necessitating methods for learning from small numbers of examples.

- In addition, more than one type of unusual event may occur in a given data sequence, where the event types can be expected to differ markedly from one another. This implies that training a single model to capture all unusual events will generally be infeasible, further exacerbating the problem of learning from limited data.

- As well as such modeling problems due to rarity, the unexpectedness of unusual events means that defining a complete event lexicon will not be possible in general, especially considering the genre- and task-dependent nature of event relevance.

Finally, from a machine learning perspective, one objective of this thesis is to develop statistical models for handling multiple interacting streams.

In meetings, each participant can be seen as a multimodal data stream. Modeling human interaction in meetings thus involves simultaneous recording and analysis of multiple streams. These streams may be asynchronous, have different frame rates, exhibit different stationarity properties, and carry complementary (or correlated) information. Some of these problems can already be tackled by one of the many existing statistical approaches of sequence modeling. However, several challenging research issues are still open, such as taking into account asynchrony and correlation between multiple interacting streams, or handling the underlying growing complexity.

1.3 Contributions

The three problems we address and models we propose in this thesis are shown in Figure 1.2, and the contributions of this thesis can be enlisted as follows,

I. Theory

The main contributions of this thesis from a theoretical perspective are three novel dynamic Bayesian networks (DNBs) for handling multiple interacting streams.

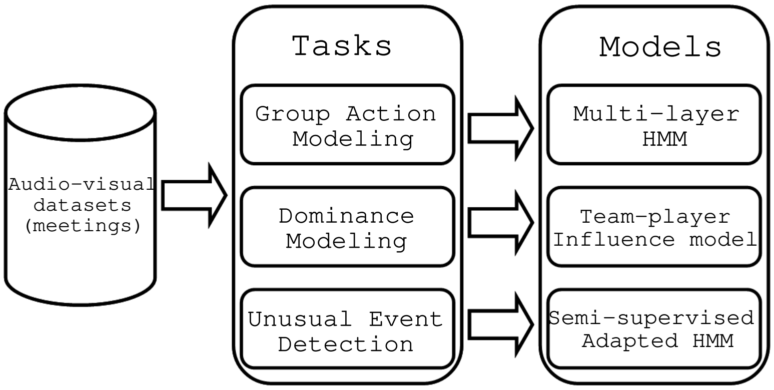


Figure 1.2. The three problems we address and the three models we propose in this thesis.

1. **The Two-layer Hidden Markov Model**

   We proposed a two-layer framework, shown in Figure 1.3, which decomposes the problem of human interaction modeling into *individual action* and *group action* stages, thus simplifies the complexity of the task. By defining a proper set of individual actions, group actions can be modeled as a two-layer process, one that models basic individual activities from low-level audio-visual features, and another one that models the interactions. We proposed a two-layer framework that implements such concept in a principled manner, and that has advantages over previous works. First, by decomposing the problem hierarchically, learning is performed on low-dimensional observation spaces, which results in simpler models. Second, our framework is easier to interpret, as both individual and group actions have a clear meaning, and thus easier to improve. Third, different models can be used in each layer to better reflect the nature of each subproblem. This framework is general and extensible, and experiments and comparison with a single-layer HMMs baseline system showed its validity.

2. **The Team-player Influence Model**

   We proposed a model, shown in Figure 1.4, that learns the influence of interacting Markov chains within a team. The proposed model is a dynamic Bayesian network (DBN) with a two-level structure: individual-level and group-level. Individual level models actions of each player, and the group-level models actions of the team as a whole. Unlike existing multi-stream models, the influence of each player on the team, which can also be interpreted as reliability in the case of data fusion, is jointly learned with the rest of the model parameters

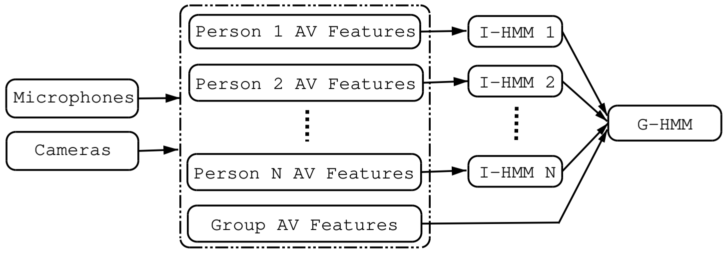


Figure 1.3. The two-layer Hidden Markov Model: the lower layer recognizes individual actions of participants using low-level audio-visual (AV) features. The output of this layer provides the input to the second layer, which models interactions. Individual actions naturally constitute the link between the low-level audio-visual features and high-level group actions. Note that *I-HMM* indicates individual action layer, and *G-HMM* indicates group action layer.

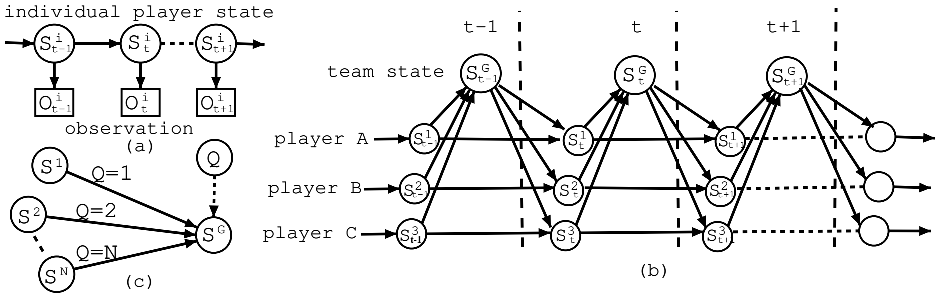


Figure 1.4. The team-player Influence Model: (a) Markov Model for individual player. (b) The team-player influence model (for simplicity, we omit the observation variables of individual Markov chains, and the switching parent variable Q). (c) Switching parents. Q is called a switching parent of S^G, and $\{S^1 \cdots S^N\}$ are conditional parents of S^G. When $Q = i$, S^i is the only parent of S^G. Note that O_t^i and S_t^i denote the observation and hidden states of the i_{th} participant (player) at time t; S_t^G denotes the group state, and Q is the switching parent.

in a principled manner using the Expectation-Maximization (EM) algorithm. Experiments on twi different applications: synthetic multi-player game and multi-party meetings showed the effectiveness of the proposed model.

3. **The semi-supervised adapted HMMs**

We proposed a semi-supervised adapted HMMs, shown in Figure 1.5, for unusual event detection. Unusual events are characterized by a number of features (rarity, unexpectedness, and relevance) that limit the application of traditional supervised model-based approaches. We proposed a semi-supervised adapted Hidden Markov Model (HMM) framework, in which usual event models are first learned from a large amount of (commonly available) training data, while unusual event models are learned by Bayesian adaptation in an unsupervised manner. The proposed framework has an iterative structure, which adapts a new unusual event model at each iteration. We showed that such a framework can address problems due to

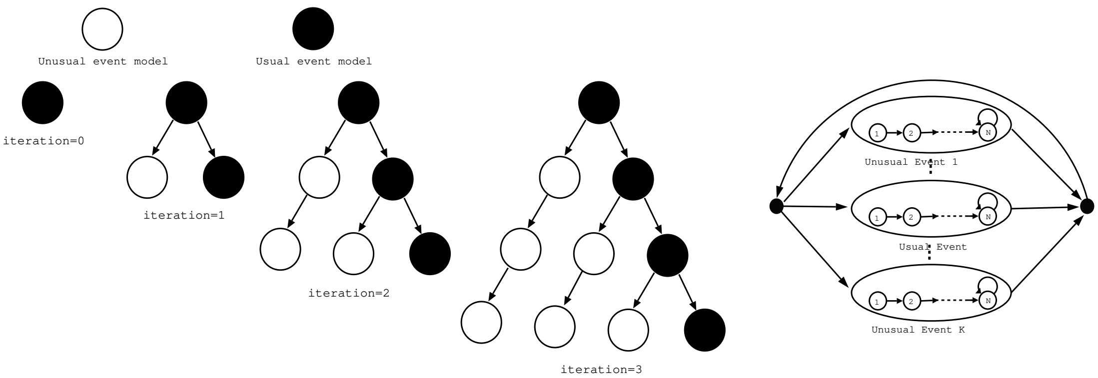


Figure 1.5. The semi-supervised adapted HMMs: At each iteration, two leaf nodes, one representing usual events and the other one representing unusual events, are split from the parent usual event node; A leaf node representing an unusual event is also adapted from the parent unusual event node.

the scarcity of training data and the difficulty in pre-defining unusual events. Experiments on audio, visual, and audio-visual data streams illustrate its effectiveness, compared with both supervised and unsupervised baseline methods.

II. Application

1. **Applications in meetings**

   Meetings are an integral part of our working lives. Recent developments in recording and storage techniques have made multimodal meeting recordings readily available, and while it is straightforward to play back such recordings, it is much more laborious for users to browse them. Our work can be useful for meeting browsing and retrieval. IDIAP developed a meeting browser, JFerret [129], which enables people to access meeting information. Some snapshots are shown in Figure 1.6 and Figure 1.7. In Figure 1.6, meeting actions are indicated with different colors. In Figure 1.7, the influence level of each meeting participant is shown over the meeting depicted by a graph. The application allows for live tracking of the influence levels. Occurrences for each of the features, such as interruptions, are observed and further processed by the model responsible for producing the final value.

2. **Other applications**

   The proposed models are general, and can be easily applied to other applications besides meetings. (i) For the first model, the multi-layer HMMs, it essentially tries to decomposes a

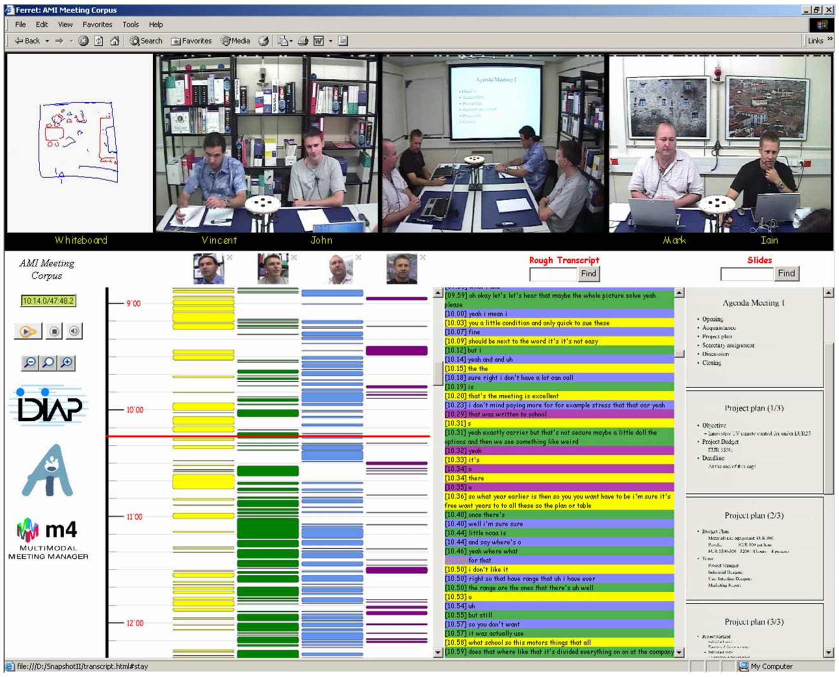


Figure 1.6. JFerret: meeting browser and retrieval system, taken from (129).

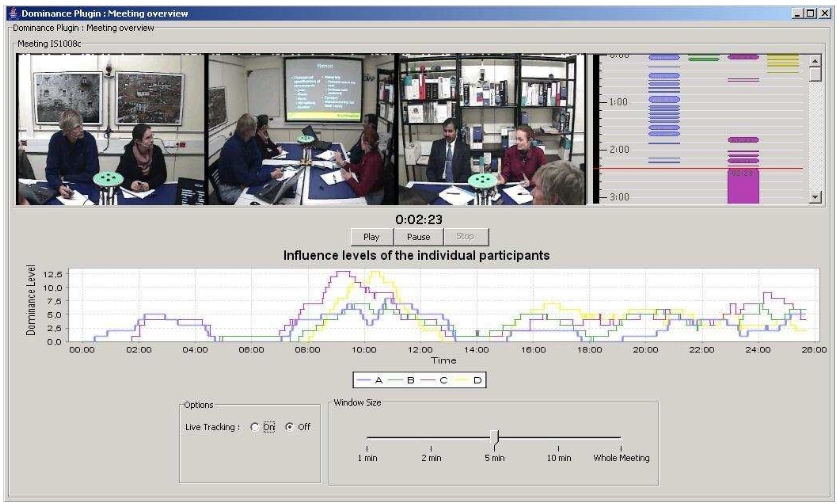


Figure 1.7. Visualization of participants' influence level, taken from (112).

complex problem (group action recognition) into stages. The basic idea of this model coincides very well with TANDEM-based automatic speech recognition (ASR) [55]. The same framework has been applied to automatic speech recognition with improved performance [67]. (ii) For the second model, the team-player influence model, it tries to learn the reliability of each individual stream among several interacting streams, and thus suitable to many multi-stream applications.

1.4 Plan of the Thesis

This thesis is organized as follows:

- In Chapter 2, we present an overview of research tasks in meetings. This is followed by discussing related work on the three problems we address in this thesis: group action modeling, dominance modeling, and unusual event modeling.
- In Chapter 3, we give a simplified tutorial of *representation*, *inference* and *learning* for probabilistic graphical models. We also present various directed and undirected graphical models. We discuss connections between various models that had previously been considered quite different, thus promoting the viewpoint that probabilistic graphical models provide a unified and useful framework for a large class of problems involving probabilistic inference.
- In Chapter 4 – 5, we address the problem of group action modeling using the multi-layer framework, which decomposes the problem of human interaction modeling into *individual action* and *group action* two states, thus simplifying the complexity of the task. We apply the framework to group action recognition (Chapter 4), and group action clustering (Chapter 5).
- In Chapter 6, we address the problem of dominance modeling in meetings using the team-player influence model. The proposed model is a dynamic Bayesian network (DBN) with a two-level structure: individual-level and group-level. The individual level models actions of each player, and the group-level models actions of the team as a whole. We applied the model to both synthetic multi-player game, and multi-party meetings.
- In Chapter 7, we present a semi-supervised adapted HMM framework for unusual event detection, in which usual event models are first learned from a large amount of (commonly

available) training data, while unusual event models are learned by Bayesian adaptation in an unsupervised manner. The proposed framework has an iterative structure, which adapts a new unusual event model at each iteration. We showed that such a framework can address problems due to the scarcity of training data and the difficulty in pre-defining unusual events.

- We conclude the thesis with Chapter 8, which summarizes the contributions and comments on the directions to explore.

Chapter 2

Research Tasks in Meetings and Related Work

A continued exponential decline in the cost of both communications and information technology would seem to enable a large and diverse array of services for enhancing human-to-human intearction [27]. This motivates several projects, such as IM2 (Interactive Multimodal Information Management), M4 (Multimodal Meeting Manager), and AMI (Augmented Multi-party Interaction), dedicated to the research and development of technology that will augment communications between individuals and groups of people in meetings.

The following section outlines research tasks in meetings. This is followed by discussing related work on the three problems we will address in this thesis – group action modeling, dominance modeling and unusual event modeling.

2.1 Research Tasks in Meetings

The automatic analysis of meetings has recently attracted attention in a number of fields, including audio and speech processing, computer vision, human-computer interaction, and information retrieval [126, 87, 28, 80, 56, 131, 33]. Analyzing meetings poses many technical challenges, and also opens doors to a number of relevant applications. A number of groups are investigating the application of speech, video, and natural language processing techniques to the meeting domain. Some

Table 2.1. Research topics in the context of meetings and the potentially involved modalities

Topic	Audio	Video	Text	White-board
Keyword spotting	√			
Speaker Identification	√			
Speaker Clustering	√			
Syntax	√			
Prosody	√			
Gaze tracking		√		
Face detection		√		
Face recognition		√		
Facial expression		√		
Gesture recognition		√		
Posture recognition		√		
Lip tracking		√		
Handwriting recognition		√	√	√
Diagram analysis		√	√	√
Document retrieval			√	
Topic clustering	√		√	
Tracking	√	√		
Person identification	√	√		
Speech recognition	√	√		
Action / Interaction recognition	√	√	√	√
...	...	...	...	...

meeting projects are summarized as followings. The meeting project at ICSI [87] focuses primarily on the challenging problem of automatic recognition of natural conversational speech in meetings. In CMU meeting project, researchers worked on speech transcription and summarization, development of the meeting browser [126], and tracking the focus of attention in meetings [122]. The meeting project at Microsoft developed a distributed meeting system that supports live broadcast of audio and video meeting data [28]. The CHIL (Computers In the Human Interaction Loop) project (`http://chil.server.de/servlet/is/101/`) develops systems that gather all relevant information (speech, faces, people, writing, emotion, *etc.*), and model and interpret human activity, behavior, and actions to better serve what a human really needs within indoor environments. The goal of the VACE (Video Analysis and Content Extraction) project is to answer questions such as *"who was at the meeting?"*, *"what is the relationship between participants?"*, and *"what events took place?"*, using both visual and audio cues. The IDIAP smart meeting room related projects (IM2, M4, AMI) are investigating how information from meetings can be captured, stored, structured, queried, and browsed using multimodal sensors, human interaction analysis, and development of user interfaces [86].

Meetings can be characterized by their multimodal nature. In the context of meetings, research tasks related with speech, the main modality in meetings, such as speech recognition, speaker segmentation and clustering, topic detection, and dialogue modelling, are being actively researched [70, 87, 126, 109]. Research tasks based on visual processing, such as face detection, tracking and recognition, and focus of attention detection, have also been examined in [127, 28]. Besides these research topics, text analysis, gestures recognition, and facial expressions recognition, as well as many other audio-only, visual-only and audio-visual research tasks can be identified in the meeting scenario. Some research topics in meetings are summarized in Table 2.1.

2.2 Related Work

Research shows that high complexity information is mostly exchanged through face-to-face interactions [4]. In this thesis, we use meeting as a study case of human interactions. In particular, we focus on the following three tasks: group action modeling, dominance modeling, and unusual event modeling. Next, we describe related work on the three tasks separately.

2.2.1 Related Work: Group Action Modeling

Learning-based approaches for the automatic interpretation of human activities in videos have been used for the past ten years. Most works have focused on supervised learning methods, which define models for specific activities that suit the goal in a particular domain, and use statistical methods for recognition. Predominately, the recognition of individual actions [120], or interaction involving few people [99, 59] has been investigated using visual features [46, 59, 99, 120, 133], although some work in the speech community can also be categorized as interaction recognition [56, 131]. In [56], recognition of a specific kind of interaction in meetings (agreement vs. disagreement) has been addressed using both word-based features (such as the total number of words, and the number of "positive" and "negative" keywords), as well as prosodic cues (such as pause, frequency and duration). In [131], the relationship between "hot spots" (defined in terms of participants highly involved in the discussion) and dialogue acts has been examined using contextual features (such as speaker identity or type of the meeting), and lexical features (such as utterance length and perplexity).

Regarding statistical models, most of the existing work has used Hidden Markov Models (HMMs)

[106], and extensions, including coupled HMMs, input-output HMMs, multi-stream HMMs, and asynchronous HMMs (see Chapter 3 for a review of models). Although the basic HMM, a discrete state-space model with an efficient learning algorithm, works well for temporally correlated sequential data, it is challenged by a large number of parameters, and the risk of over-fitting when learned from limited data [97]. This situation might occur in the case of multimodal group action recognition where, in the simplest case, possibly large vectors of audio-visual features from each participant are concatenated to define the observation space [80, 81].

The above problem is general, and has been addressed using hierarchical representations [134, 33, 97]. In [134], an approach for unsupervised discovery of multi-level video structures using hierarchical HMMs was proposed, in the context of sports videos. In this model, the higher-level structure elements usually correspond to semantic events, while the lower-level states represent variations occurring within the same event. In [33], two methods for meeting structuring from audio were presented, using multilevel DBNs. The first DBN model decomposes group actions in meetings as sequences of sub-actions which have no explicit meanings, and obtained from the training process. The second DBN model processes independently features of different nature, and integrates them at a higher level. In both [134, 33], the low-level actions have no obvious interpretation, and the number of low-level actions is a model parameter learned during training, or set by hand, which makes the structure of the models difficult to interpret. In [97], a layered HMMs were proposed to model multimodal office activities involving only mainly one person at various time granularities. The lowest layer captures one video and two audio channels, plus keyboard and mouse activity features; the middle layer classifies audio-visual features into basic events like *"speech"*, *"music"*, *"one person"*, *"nobody"*, etc. Finally, the highest layer uses the outputs of previous layers to recognize office activities with longer temporal extent. In this way, actions at different semantic levels and with different time granularities have been modeled with a cascade pyramid of HMMs. This hierarchical representation has been tested in SEER, a real-time system for recognizing typical office activities, and produced improvement over a simple baseline HMM.

In the meeting scenario, there is recently a large body of work of automatically segmentation and recognition of group meeting actions using dynamic Bayesian network (DBN) and artificial neural networks (ANN) [130, 29, 142, 1, 108]. In chapter 4 and 5, we present our own work on group action recognition and clustering.

2.2.2 Related Work: Dominance Modeling

According to Merriam-Webster Online (`www.m-w.com`), *influence* is *'the power to direct the thinking or behavior of others'*, and *dominance* is *'controlling power or influence over others'*. Obviously, these two words are closely related. Social psychology has studied the concepts of influence and dominance arising from group discussions for several years. One theory is called *Status Characteristics and Expectation states theory* [14]. According to it, if members of a group are either observable or known to be differentiated with respect to *social status* (occupation, age, race, or gender), the group's influence ranking will be correlated with variations in social status. The theory assumes that in the course of problem solution seeking, solutions offered by those of higher social status are most likely to be correct. Another theory is called *Two Process Theory* [96]. This theory assumes that *variations in demeanor* are correlated with influence. Variations in assertiveness and other components of demeanor explain attainment of positions in the eventual rankings. Finally, social psychologists also found that the *unequal distribution of amounts of verbal participation*, the *directionality of initiation of speech exchanges* and the *rates of addressing the group as a whole* are also correlated with influence [8, 51].

Although the literature on modelling and understanding the concepts of dominance and influence in multi-party interaction is abundant, very few attempts to automatically estimate such quantities in real discussions have been made so far [9, 111, 140]. Regarding influence, existing approaches assume that (1) this high-level concept can potentially be deduced from low and mid-level signal observations [107], and (2) such observations present regularities (patterns) that models for recognition and discovery are able to extract. Basu et al. [9] described an approach for automatic discovery of influence, in a multi-sensor lounge room where people played interactive debating games, using the *influence model*. This model is a Dynamic Bayesian Network (DBN) which regards group interactions as a group of Markov chains, each of which influences the others' state transitions. Although this model is a tractable option, it has the limitation that it only models influence between pairs of players, and does not explicitly model the group as such.

In another work, Rienks et al. [111] recently proposed a supervised learning approach. The method was based on the formulation of the problem as a three-class classification task in which, through manually annotated data, meeting participants are labelled as having high, normal or low dominance, using a Support Vector Machine (SVM). A number of features related to speaker-turns and their content were extracted for each participant from speaker-turn segmentations, speech transcriptions, and addressing labels, all of which were manually produced.

In Chapter 6, we propose the team-player influence model to quantitatively investigate the influence of individual players on their team. In the proposed model, the player level represents the actions of individual players. The team level represents group-level actions. The explicit hierarchy in the model allows for the estimation of the influence of each of the players on the team state, and the distribution of player-to-team influence is automatically learned from data in an unsupervised fashion.

2.2.3 Related Work: Unusual Event Modeling

In some event detection applications, events of interest occur over a relatively small proportion of the total time: e.g. alarm generation in surveillance systems, and extractive summarization of raw video events. The automatic detection of temporal events that are relevant, but whose occurrence rate is either expected to be very low or cannot be anticipated at all, constitutes a problem which has recently attracted attention in computer vision and multimodal processing under an umbrella of names (abnormal, unusual, or rare events) [121, 141, 21]. In this thesis, we employ the term *unusual event*, which we define as events with the following properties: (1) they seldom occur (rarity); (2) they may not have been thought of in advance (unexpectedness); and (3) they are relevant for a particular task (relevance).

There is a large amount of work on event detection. Most works have been centered on the detection of predefined events in particular conditions using supervised statistical learning methods, such as HMMs [99, 22, 128], and other graphical models [19, 84, 58, 52]. In particular, some recent work has attempted to recognize *highlights* in videos, e.g., sports [113, 22, 128]. In our view, this concept is related but not identical to unusual event detection. On one hand, typical highlight events in most sports can be well defined from the sports grammar and, although rare, are predictable (e.g., goals in football, home-runs in baseball, etc). On the other hand, truly unusual events (e.g. a blackout in the stadium) could certainly be part of a highlight.

Fully supervised model-based approaches are appropriate if unusual events are well-defined and enough training samples are available. However, such conditions often do not hold for unusual events, which render fully supervised approaches ineffective and unrealistic. To deal with the problem, an HMM approach was proposed in [21] to detect unusual events in aerial videos. Without any models for usual activities, and with only one training sample, unusual event models are hand-coded using a set of predefined spatial semantic primitives (e.g. "close" or "adjacent"). Although unusual event models can be created with intuitive primitives for simple cases, it is infeasible for complex events, in which primitives are difficult to define.

As an alternative, unsupervised approaches for unusual event detection have also been proposed [121, 141]. In a far-field surveillance setting, the use of co-occurrence statistics derived from motion-based features was proposed in [121] to create a binary-tree representation of common patterns. Unusual events were then detected by measuring aspects of how usual each observation sequence was. The work in [141] proposed an unsupervised technique to detect unusual human activity events in a surveillance setting, using analysis of co-occurrence between video clips and motion / color features of moving objects, without the need to build models for usual activities.

In Chapter 7, we propose a semi-supervised adapted HMM framework for unusual event modeling. Our work attempts to combine the complementary advantages of supervised and unsupervised learning in a probabilistic setting. On one hand, we learn a general usual event model exploiting the common availability of training data for such an event type. On the other hand, we use Bayesian adaptation techniques to create models for unusual events in an iterative, data-driven fashion, thus addressing the problem of lack of training samples for unusual events, without relying on pre-defined unusual event sets.

2.3 Summary

In this chapter, we presented a summary of research tasks in meetings. We also described some related work on the three problems: group action modeling, dominance modeling and unusual event modeling.

Now that the problems and related work have been introduced, we will focus in the next chapters on computational models. In the next chapter, we will introduce probabilistic graphical models.

Chapter 3

Probabilistic Graphical Models

3.1 Introduction

The following quotation, from the preface of [38], provides a very concise introduction to graphical models.

> *Graphical models are a marriage between probability theory and graph theory. They provide a natural tool for dealing with two problems that occur throughout applied mathematics and engineering – uncertainty and complexity – and in particular they are playing an increasingly important role in the design and analysis of machine learning algorithms. Fundamental to the idea of a graphical model is the notion of modularity – a complex system is built by combining simpler parts. Probability theory provides the glue whereby the parts are combined, ensuring that the system as a whole is consistent, and providing ways to interface models to data. The graph theoretic side of graphical models provides both an intuitively appealing interface by which humans can model highly-interacting sets of variables as well as a data structure that lends itself naturally to the design of efficient general-purpose algorithms.*
>
> *Many of the classical multivariate probabilistic systems studied in fields such as statistics, systems engineering, information theory, pattern recognition and statistical mechanics are special cases of the general graphical model formalism – examples include mixture models, factor analysis, hidden Markov models, Kalman filters and Ising*

models. The graphical model framework provides a way to view all of these systems as instances of a common underlying formalism. This view has many advantages – in particular, specialized techniques that have been developed in one field can be transferred between research communities and exploited more widely. Moreover, the graphical model formalism provides a natural framework for the design of new systems.

The primary goal of this chapter is to present a coarse introduction to graphical models, and to point the reader to the relevant literature on the topic, with an emphasis on various dynamic Bayesian networks (DBNs) and their connections.

This chapter is organized as follows. Section 3.2 describes directed graphical models (i.e. Bayesian networks). From Section 3.2.1 to Section 3.2.12, we describe various dynamic Bayesian networks. We sum up the relationships of various DBN models in Section 3.2.13. We then describe their advantages in Section 3.2.14. Section 3.3 introduces undirected graphical models, and Section 3.4 briefly describes computations, *i.e.* inference, decoding and learning, in graphical models. Conclusions are drawn in Section 3.5.

3.2 Directed Graphical Models

A basic concept in probability theory is that a physical domain can be completely characterized by the joint probability distribution function (PDF) of all the random variables in the domain. A directed graphical model (i.e. a Bayesian network) of a physical domain represents the causal relationship between random variables, i.e., it explicitly "factorizes" the random variables set, in terms of conditional probability distributions, to simplify the computation of joint PDFs. In a directed graphical model, each node represents a variable, and the arc from one node A to another node B indicates statistical dependence, and can be informally interpreted as indicating that A "causes" B. The *lack* of possible arcs in the model encode conditional independencies. In particular, given the model structure, the joint probability distribution is given by

$$p(x_1, ..., x_n) = \prod_{i=1}^{n} p(x_i | pa(x_i)) \tag{3.1}$$

where x_i denotes the variable node, and $pa(x_i)$ denotes the parent of node x_i.

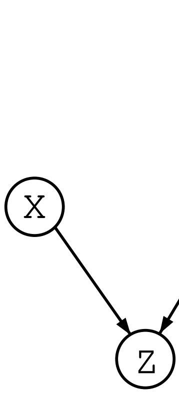


Figure 3.1. A BN represent the joint distribution of four variables: $\{W, X, Y, Z\}$. W and X are independent variables, Y only depends on W, and Z depends on both X and Y. The joint PDF is $P(W, X, Y, Z) = P(W)P(X)P(Y|W)P(Z|X, Y)$.

For example, Figure 3.1 shows a Bayesian network (BN) representing a physical domain with four variables: $\{W, X, Y, Z\}$. We can see that W and X are independent variables, Y only depends on W, and Z depends on both X and Y. The joint PDF is given by

$$P(W, X, Y, Z) = P(W)P(X)P(Y|W)P(Z|X, Y). \tag{3.2}$$

A dynamic Bayesian network (DBN) is a Bayesian network with the same structure unrolled in the time axis as shown in Figure 3.2. The DBNs themselves are still Bayesian networks in nature, but they are generally considered as state-space representations which model the system state dynamics by observed variables, though it is not necessarily required that the system states be modeled explicitly as "hidden nodes" (Hidden nodes are nodes whose values are not known). Their periodic structure can be taken advantage of to simplify the computation. Hidden Markov models (HMMs) and Kalman filters can be considered as special types of DBNs. They both have system states represented explicitly as hidden nodes. HMMs are DBNs with discrete state nodes, while Kalman filters are DBNs with continuous state and observation nodes.

Although we will focus on temporal domain DBNs in this thesis, BNs can in fact also be unrolled in the spatial domain and other domains. It is also possible to change the structure of the Bayesian network at each time slice, which is called dynamic Bayesian multi-nets as discussed in [16].

Next, we will briefly describe various DBN examples proposed to address specific applications and to overcome certain limitations of the traditional HMM. Table 3.1 summarizes these DBN

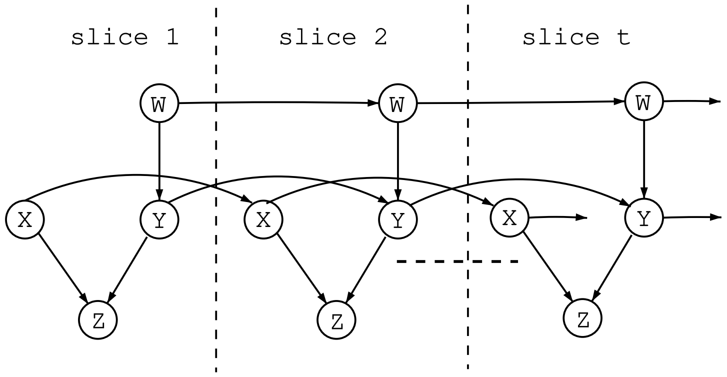


Figure 3.2. A dynamic Bayesian network (DBN) representation by unrolling a Bayesian network (BN) in the temporal domain.

Table 3.1. Comparison of various dynamic Bayesian networks (DBN) with respect to *representation*, *inference* and *learning* (Note that the columns of 'state' and 'observation' indicate the number of state and observation sequences respectively. Note: [1]Ascent, rather than descent, since we are trying to maximize log-likelihood. [2]The Junction tree algorithm is equivalent to the classic Forward-Backward algorithm for HMMs.

Model	**Representation**		**Inference**		**Learning**
	state	observation	exact	approximate	
HMM	1	1			
Multi-observation HMM	1	$\geqslant 2$		Boyen-Koller algorithm	Expectation
Hierarchical HMM	$\geqslant 2$	1		Factored Frontier algorithm	-Maximization
Input-output HMM	1	2		Loopy belief propagation	(EM)
Asynchronous HMM	1	$\geqslant 2$		Variational methods	
Multi-stream HMM	$\geqslant 2$	1	Junction	Particle Filtering	Gradient
Factorial HMM	$\geqslant 2$	1	tree	Markov Chain	ascent[1]
Coupled HMM	$\geqslant 2$	$\geqslant 2$	algorithm[2]	Monte Carlo (MCMC)	
Influence Model	$\geqslant 2$	$\geqslant 2$			
Dynamic Bayesian Multi-net	$\geqslant 2$	$\geqslant 2$			
Mixed-memory Markov Model	$\geqslant 2$	$\geqslant 2$			
Dynamical System Trees	$\geqslant 2$	$\geqslant 2$			

models concerned with *representation*, *inference* and *learning*. Our descriptions are mainly focused on *model semantics*. We then describe the general graphical model learning and inference in Section 3.4.

To facilitate description, we first define the following notations:

- N: the total number of sequences.
- $\mathrm{O}_t^{(i)}$: the observation of the i^{th} sequence at time t.
- $\mathbf{O}_t^a$: the observation of the audio sequence at time t.
- $\mathbf{O}_t^v$: the observation of the video sequence at time t.
- $\mathrm{S}_t^{(i)}$: the hidden state of the i^{th} sequence at time t.
- X: the input sequence (used in input-out HMMs, see Section 3.2.4).
- θ: the model parameters.

3.2.1 Hidden Markov Models

HMMs [106], as shown in Figure 3.3, are the simplest case of DBNs, with one observation variable (in black) and one hidden variable (in white). As in all BNs, the joint probability of observation and hidden variables is defined by $P(x_i|pa(x_i))$, which in the case correspond to the so-called transition and emission probabilities of HMMs, given by

$$\begin{cases} P(\mathrm{S}_t|\mathrm{S}_{t-1}) \\ P(\mathrm{O}_t|\mathrm{S}_t). \end{cases} \tag{3.3}$$

HMMs have been widely used in modeling time series, including speech recognition [105], natural language processing [92], traffic surveillance [42], protein modeling [30], musical analysis/synthesis [34], visual gesture recognition [135], and numerous others.

3.2.2 Multi-Observation Hidden Markov Models

Multi-Observation HMMs, as shown in Figure 3.4, can be used to factorize the observation space, *i.e.*, different observations for different sequences. If these observation sequences are independent

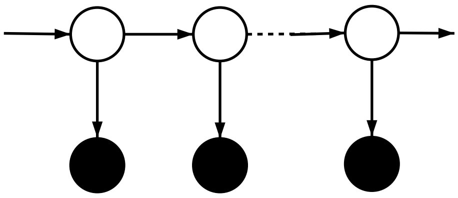


Figure 3.3. Hidden Markov Model: one observation variable and one hidden variable at each time slice. Black nodes indicate *observation* variables, while white nodes indicate *hidden* variables.

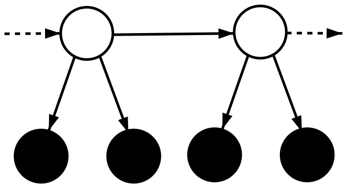


Figure 3.4. Multi-Observation Hidden Markov Model: multiple observation variables and one hidden variable at each time slice.

of each other, the observation probability can be expressed as a product of individual observation probabilities. Hence, while the transition probabilities are the same as HMMs, the emission probabilities are modeled as a factorized distribution, that is,

$$\begin{cases} P(\mathrm{S}_t|\mathrm{S}_{t-1}) \\ P(\mathrm{O}_t|\mathrm{S}_t) = \prod_{i=1}^{N} P(\mathrm{O}_t^{(i)}|\mathrm{S}_t). \end{cases} \tag{3.4}$$

3.2.3 Hierarchical Hidden Markov Models

The Hierarchical HMMs [41], as shown in Figure 3.5, are designed to model domains with hierarchical structure and/or dependencies at multiple length/time scales. Hierarchical HMMs have multiple "levels" of states which describe input sequences at different levels of granularity. For example, for the task of text information retrieval, the top level of the HMMs represent sentences at the level of phrases, and the lower level of the HMMs represent sentences at the level of individual words.

Hierarchical HMMs have been used for text information retrieval [118] and video content analysis [134].

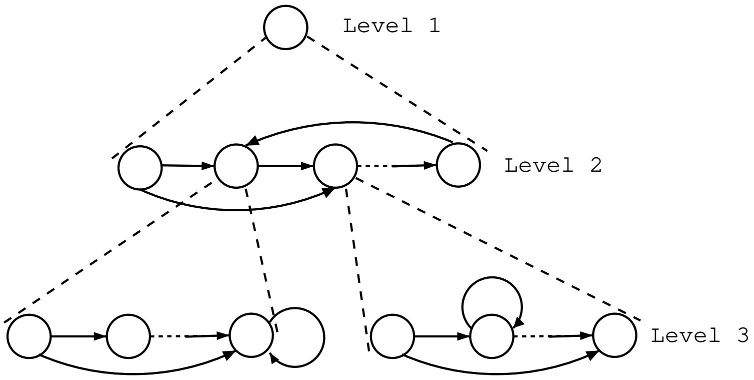


Figure 3.5. Hierarchical Hidden Markov Model (a three level case), multiple levels of states which describe input sequences at different levels of granularity. Note that this graph should not be confused with the other dynamic Bayesian models we introduce in this chapter. Other models are represented as slices evolving over time.

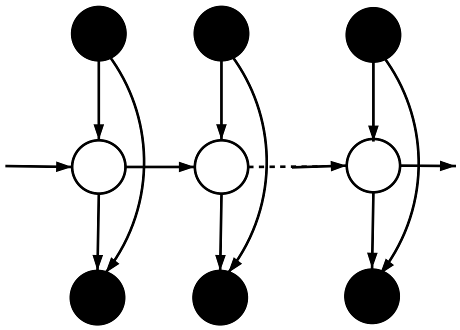


Figure 3.6. Input-Output Hidden Markov Model, modeling the input-output sequence pair.

3.2.4 Input-Output Hidden Markov Models

Input-output HMMs [13], as shown in Figure 3.6, model the input-output sequence pair. The advantages of input-output HMMs over HMMs mainly lie in two aspects [13]: (i) When the output sequence is discrete, the training criterion is discriminant, since input-output HMMs use the maximum a posteriori criterion. (ii) We expect long-term dependencies to be more easily learned in input-output HMMs than in HMMs, because the transition probabilities are less ergodic (i.e., the state variable does not “mix” and forget past contexts as quickly).

The emission and transition probabilities of input-output HMMs are given by

$$\begin{cases} P(\mathrm{S}_t|\mathrm{S}_{t-1}, \mathrm{X}_t) \\ P(\mathrm{O}_t|\mathrm{S}_t, \mathrm{X}_t), \end{cases} \tag{3.5}$$

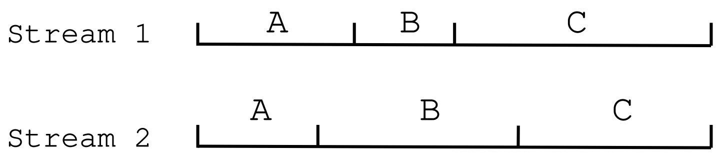


Figure 3.7. Asynchrony between streams: stream 1 and stream 2 describe the same sequence of three states $A-B-C$, but there is some asynchrony between them.

where O is the output (observations), and X is the input observed as well as output O. Input-output HMMs are a probabilistic mapping from inputs, $X_{1:T}$, to outputs, $O_{1:T}$.

Input-output HMMs have been used for hand gesture recognition [64], EEG rhythms modeling [23], and synthesizing facial animation from an input audio sequence [76].

3.2.5 Asynchronous Hidden Markov Models

Asynchronous HMMs [11] were proposed to handle the possible asynchrony between observation streams. For instance, assume there are two streams (taking audio and video as example) describing the same sequence of three events $A-B-C$, as shown in Figure 3.7. There is asynchrony between them because the best piecewise stationary alignment of each stream to the sequence $A-B-C$ do not coincide temporally with each other.

Asynchronous HMMs [11] combine two streams by learning the joint distribution of pairs of sequences when these sequences are not synchronized and are not of the same length or rate. The asynchronous HMMs model the joint distribution of the two streams by maximizing the likelihood of L observation sequences as follows (taking audio-visual streams as an example),

$$\theta^* = \arg\max_{\theta} \prod_{l=1}^{L} P(\mathbf{O}_l^a, \mathbf{O}_l^v, \tau_t|\theta). \tag{3.6}$$

The additional hidden synchronization variable τ_t can be seen as alignment between sequences $\mathbf{O}_l^a$ and $\mathbf{O}_l^v$. With the hidden variable τ_t and using several reasonable independence assumptions, the model in [11] can factor the joint likelihood of the data and the hidden variables into several simple conditional distributions, which makes the model tractable using the EM algorithm. The Viterbi algorithm can be used to obtain the optimal state sequence as well as the alignment between the two sequences.

Asynchronous HMMs have been used for audio-visual speech recognition and the task of multi-

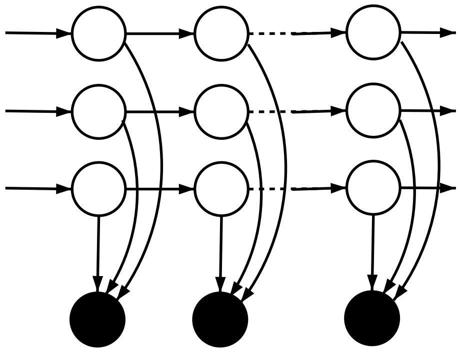


Figure 3.8. Factorial Hidden Markov Model, linking multiple Markov chains through a common output emission stream.

modal authentication [11].

3.2.6 Multi-stream Hidden Markov Models

Multi-stream HMMs combine several sequences where each stream is modeled independently. Multi-stream HMMs have been widely used in audio-visual speech recognition [36, 89].

We take audio and video streams as an example. Let $\theta^* = (\theta_a^*, \theta_v^*)$ be the best model parameters to maximize the likelihood of audio-only and visual-only sequences respectively, trained based on,

$$\theta_a^* = \arg\max_{\theta_a} \prod_{l=1}^{L} P(\mathbf{O}_l^a|\theta_a). \tag{3.7}$$

$$\theta_v^* = \arg\max_{\theta_v} \prod_{l=1}^{L} P(\mathbf{O}_l^v|\theta_v). \tag{3.8}$$

The final classification is based on the fusion of the outputs of the two sequences by estimating their joint occurrence as follows:

$$P(\mathbf{O}_l^{a+v}|S_t) = P(\mathbf{O}_l^a|S_t, \theta_a)^\omega P(\mathbf{O}_l^v|S_t, \theta_v)^{(1-\omega)}, \tag{3.9}$$

where the weighting factor $\omega (0 \leq \omega \leq 1)$, which represents the relative reliability of one of the two streams, is often chosen manually based on expert knowledge.

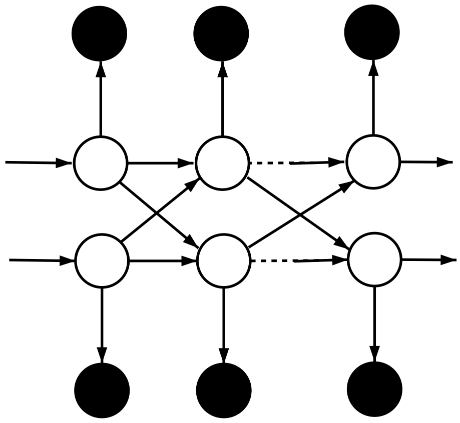


Figure 3.9. Coupled Hidden Markov Model, directly linking hidden states of multiple interacting processes.

3.2.7 Factorial Hidden Markov Models

Factorial HMMs [47], as shown in Figure 3.8, enrich the representation power of hidden states by indirectly linking multiple Markov chains through a common output emission stream. Factorial HMMs are used to model one sequence in a complicated way, rather than explore the interaction or dependency between two sequences. The emission and transition probabilities of factorial HMMs are given by

$$\begin{cases} P(\mathrm{S}_t|\mathrm{S}_{t-1}) = \prod_{i=1}^{N} P(\mathrm{S}_t^{(i)}|\mathrm{S}_{t-1}^{(i)}) \\ P(\mathrm{O}_t|\mathrm{S}_t^{(1)}, \mathrm{S}_t^{(2)} ... \mathrm{S}_t^{(K)}) = \prod_{i=1}^{N} P(\mathrm{O}_t|\mathrm{S}_t^{(i)}) \end{cases} \tag{3.10}$$

Factorial HMMs have been successfully applied to the visual tracking problem [43].

3.2.8 Coupled Hidden Markov Models

Coupled HMMs, as shown in Figure 3.9, directly link hidden states of multiple interacting processes that have Markovian temporal dynamics which generate different output emission streams. The state of one model at time t depends on the states of all models (including itself) at time $t-1$. This is probably a natural structure one can come up with. For N individual HMMs coupled together, the state transition probability and emission probability is given by,

$$\begin{cases} P(\mathrm{S}_t^{(i)}|\mathrm{S}_{t-1}^{(1)}, \mathrm{S}_{t-1}^{(2)}, ..., \mathrm{S}_{t-1}^{(N)}), \quad for \;\; i = 1, ..., N \\ P(\mathrm{O}_t^{(i)}|\mathrm{S}_t^{(i)}), \quad for \;\; i = 1, ..., N. \end{cases} \tag{3.11}$$

Therefore, the state transition probability is described by a $N+1$ dimensional matrix and the number of free parameters for this transition probability matrix is $M^{(N+1)}$ (where M is the number of states for each individual HMMs, and assume the number of hidden states M is the same for all models), which is exponential in the number of models coupled together. Obviously, this is not a desirable feature because it makes accurate parameter learning very difficult.

Coupled HMMs directly link hidden Markov chains together so they can influence each other in time. But, unlike simpler models, this involves untractable inference and may require sampling or other approximation methods.

Murphy [72, 94] has used coupled HMMs for two different applications: freeway traffic modeling, and audio-visual speech recognition (AVSR). Oliver [98] applied coupled HMM to Tai-Chi gesture recognition (two-hand gesture interactions).

3.2.9 The Influence Model

Coupled HMMs become intractable for more than two streams. The Influence model [5, 9, 24] simplifies the joint probability of coupled HMMs into a combination of pairwise conditional probabilities,

$$\begin{cases} P(\mathrm{S}_t^{(i)}|\mathrm{S}_{t-1}^{(1)}, \mathrm{S}_{t-1}^{(2)}, ..., \mathrm{S}_{t-1}^{(N)}) = \sum_{j=1}^{N} \alpha_{ji} P(\mathrm{S}_t^{(i)}|\mathrm{S}_{t-1}^{(j)}),, \quad for \;\; i = 1, ..., N \\ P(\mathrm{O}_t^{(i)}|\mathrm{S}_t^{(i)}),, \quad for \;\; i = 1, ..., N. \end{cases} \tag{3.12}$$

In [9, 24], the influence model was used to analyze speaking turn-taking patterns in order to determine how much influence one participant has on the others. In other words, α_{ji} is regarded as the influence of the j^{th} participant on the i^{th} participant. In [5], the influence model has been applied to complex control systems, such as power stations.

3.2.10 Dynamic Bayesian Multi-nets

Dynamic Bayesian multi-nets [16] are motivated by model structure learning. Although a fully-connected graphical model can represent any probability distribution representable by a sparsely structured one, a graphical model should represent a dependence between two random variables only when necessary, where "necessary" depends on the problem domain. There are many reasons for not using such a fully connected model: (i) there are fewer computational and memory requirements for sparse network structures than the fully-connected structure; and (ii) learning a sparse network needs less training examples and less prone to over-fitting; and (iii) the resulting structure might be easier to interpret, thus reveal high-level knowledge about the underlying problem domain.

A dynamic Bayesian multi-net can be thought of as a network where edges can *appear or disappear* depending on the values of certain nodes in the graph. Dynamic Bayesian multi-nets can be implemented by the use of "switching parents". The basic idea of switching parents is as follows: a variable, S^G in this case, has a set of parents $\{Q, S^1 \cdots S^N\}$. Q is the switching parent that determines what other parents should change (or switch) conditioned on the current value of the switching parent. Figure 3.10 shows the case where variable Q switches the parents of S^G among $\{S^1 \cdots S^N\}$, corresponding to the probability distribution,

$$\begin{aligned}
P(S^G|S^1 \cdots S^N) &= \sum_{i=1}^{N} P(S^G, Q = i|S^1 \cdots S^N) && (3.13)\\
&= \sum_{i=1}^{N} P(Q = i|S^1 \cdots S^N) P(S^G|S^i \cdots S^N, Q = i) && (3.14)\\
&= \sum_{i=1}^{N} P(Q = i) P(S_t^G|S_t^i) && (3.15)
\end{aligned}$$

From Equation 3.14 to Equation 3.15, we made the following two assumptions: (i) Q is independent of $\{S^1 \cdots S^N\}$, and (ii) when $Q = i$, S_t^G only depends on S_t^i.

3.2.11 Mixed-Memory Markov Models

Mixed-memory models [115] are motivated to decompose a complex model into simpler ones. For example, a mixed-memory model could decompose a higher-order Markov model (assuming the

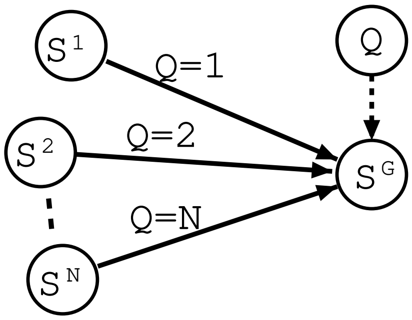


Figure 3.10. Switching parents: when $Q = 1$, S^1 is the only parent of S^G; when $Q = 2$, S^2 is the only parent of S^G. Q is called a switching parent of S^G, and $\{S^1 \cdots S^N\}$ are conditional parents of S^G.

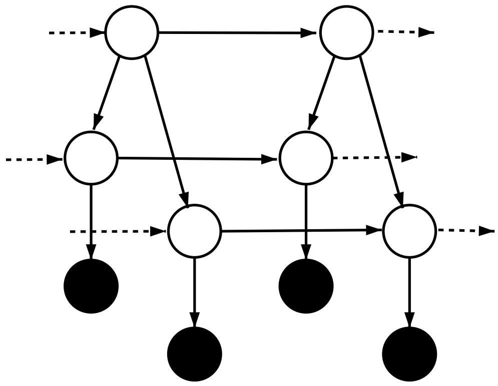


Figure 3.11. Dynamical System Trees, describing multiple processes that interact via a hierarchy of aggregating processes.

current state depends only on the K preceding states), say, K-order, into mixtures of first-order Markov models, i.e., $P(S_t|S_{t-1}S_{t-2}\cdots S_{t-K}) = \sum_{i=1}^{K} \alpha_i P(S_t|S_{t-i})$.

We can see that the mixed-memory Markov model and the influence model use the same strategy to reduce complex models with large state spaces into a combination of simpler ones with smaller state spaces. Mixed-memory Markov models have been used for automatic language identification [68].

3.2.12 Dynamical System Trees

Dynamical system trees (DSTs), as illustrated in Figure 3.11, were proposed in [60] as a flexible model for describing multiple processes that interact via a hierarchy of aggregating processes. DSTs extend nonlinear dynamical systems to the scenario of an interacting group. Various individual processes interact as communities and sub-communities in a tree structure that is unrolled in time.

[60] provides tractable inference and learning algorithms for arbitrary DSTs topologies via an efficient structured mean-field algorithm. The diverse applicability of DSTs is demonstrated on two data sets: gene expression data and a data set of group behavior in the setting of an American football game.

3.2.13 Relationships Between Various DBNs

Above, we described various DBN examples. Although these DBNs are quite different in terms of model structure, there are strong relationships among them, summarized as follows.

First, complex DBN models can be constructed from simple DBN models by adding more nodes (representing random variables) and edges between nodes (denoting direct dependencies between the corresponding variables). For example, we can add more observation variables (assuming independence from each other) in a hidden Markov model in order to form a multi-observation hidden Markov model.

Second, training of complex DBN models can be performed on simple DBN models, and these simple models can then be used to construct complex models in which learning and inference are performed. That is, we can often break down the task of training a complex DBN model into a number of simpler subproblems, which is called *'divide and conquer'*.

Finally, we also notice that there are similarities among some of the above DBN models. For example, coupled HMMs, influence models, mixed-memory models and dynamic Bayesian multi-nets have a similar representation, but have been independently reported in the literature [16, 115, 68, 98, 5, 9, 24]. The relationships between these DBN models are summarized as follows: (i) Dynamic Bayesian multi-nets were motivated by the model selection, i.e., learning the model structure from the data; (ii) a dynamic Bayesian multi-net can be thought of as a network where edges can appear or disappear depending on the values of certain nodes in the graph; (iii) therefore, dynamic Bayesian multi-nets are more general. Mixed-memory models, the influence models, coupled HMMs can be viewed as special cases of dynamic Bayesian multi-nets.

3.2.14 Advantages of DBNs

Dynamic Bayesian networks (DBNs) are attractive modeling tools for many applications because they combine a natural mechanism for expressing contextual knowledge with the power of efficient algorithms for statistical learning and inference.

We list some advantages of DBNs over the standard HMM, a special and simplest case of DBNs.

- **Flexibility of the model structure:** In comparison to HMMs which only allow one hidden node and one observation node at each time slice, general DBNs allow arbitrary structures at each time slice: (i) At each time slice, there could be more than one observation node. (ii) Similarly, unlike the one-hidden-node requirement as in the HMM, there can be more than one hidden nodes representing the corresponding state of the physical world in a DBN. All the nodes of a DBN can be directly mapped from corresponding interpretations. Although this flexibility of model structure will introduce a larger computational complexity, it makes DBNs applicable to many situations.

- **Easy incorporation of expert knowledge:** The causal relations of nodes in DBNs can be encoded using expert knowledge. In this case, the nodes are not necessarily fully connected: unlink nodes are independent of each other. This simplifies the computation of the joint probability distribution of a complex system with many variables.

- **Easy interpretation of hidden variables:** For many applications, hidden variables of DBNs represent semantic concepts, thus may have physical interpretations.

Now we see that, using DBNs, we are able to represent, and hence learn much more complex models of sequential data. At the same time, we need to keep in mind that the price of using complex models is the increased algorithmic and computational complexity.

3.3 Undirected Graphical Models

Graphical models can be divided into two types: those based on directed graphs, and those based on undirected graphs. Up to now, we have discussed directed graphical model only. Next, we will describe undirected graphical model.

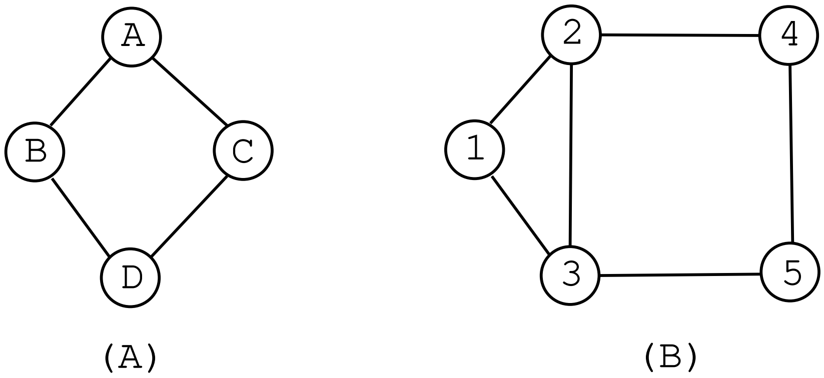


Figure 3.12. Examples of undirected graphical models.

An undirected graphical model, as shown in Figure 3.12, also known as Markov random field, is a pair (ϕ, G), where G is an undirected graph, and ϕ is a set of factors corresponding to the cliques in G. A clique C is a subset $C \subseteq V$ such that $(i, j) \in E$ for all $i, j \in C$. A clique is *maximal* if it is not contained within any other clique. For example, in Figure 3.12 (B), (1) $\{4, 5\}$ is a maximal-clique; (2) $\{4\}$ is a clique; (3) $\{1, 2\}$ is a clique; (4) $\{1, 2, 3\}$ is a maximal-clique. Since we construct the graph to reflect the structure of the factors in ϕ, graph separation implies independence and vice versa.

The graphical structure of a Markov random field may be used to factorize the joint distribution elements of ϕ into a normalized product of strictly positive, real-valued potential functions, derived from the notion of conditional independence. Each potential function operates in a subset of the random variables represented by vertices in G. The joint distribution of a Markov random field is defined by,

$$p(x) = \frac{1}{Z} \prod_{c \in C} \varphi_C(x_C) \tag{3.16}$$

where C is the set of maximal cliques in the graph. $\varphi_C(x_C)$ is a potential function (a positive, but otherwise arbitrary, real-valued function) on the clique x_C, and Z is the normalization factor, defined as:

$$Z = \sum_x \prod_{c \in C} \varphi_C(x_C). \tag{3.17}$$

In theory the structure of graph G may be arbitrary, provided it represents the conditional independencies in the label sequences being modeled. However, when modeling sequences, the

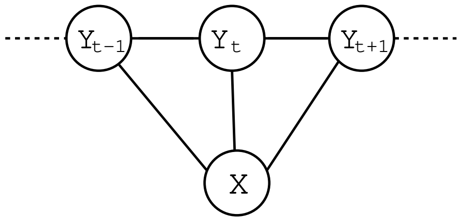


Figure 3.13. Conditional Random Fields: the hidden nodes can depend on observations at any time step, thus relaxing the independence assumptions required by HMMs.

simplest and most common graph structure encountered is that in which the nodes corresponding to elements form a simple first-order chain, as illustrated in Figure 3.13, that is called linear-chain conditional random fields (CRFs). Next, we will focus our discussion on CRFs and their extensions.

3.3.1 Conditional Random Fields and Extensions

As a specific case of undirected graphical models, conditional random fields (CRFs), shown in Figure 3.13, were introduced originally by [74] for modeling sequences. Recently, there has been an explosion of interest in CRFs, with successful applications including text processing [124, 103, 117], bio-informatics [114, 77], and computer vision [71, 54].

The underlying idea of CRFs is that of defining a conditional probability distribution over label sequences given a particular observation sequence, rather than a joint distribution over both label and observation sequences in HMMs. The primary advantage of CRFs over hidden Markov models is their conditional nature, resulting in the relaxation of the independence assumptions required by HMMs in order to ensure tractable inference. Additionally, CRFs avoid the label bias problem: the transitions leaving a given state compete only against each other, rather than against all other transitions in the model [74] – a weakness exhibited by maximum entropy Markov models (MEMMs) and other conditional Markov models based on directed graphical models.

Let $X = \{x_1, x_2, ...x_T\}$ be the observed input data sequence. Let Y be a set of states, each of which associated with a label and $\{y_1, y_2, ...y_T\}$ be a sequence of states. Linear-chain CRFs thus define the conditional probability of a state sequence given an input sequence to be

$$P(Y|X) = \frac{1}{Z_o} \exp(\sum_{t=1}^{T} F_\theta(y_t, y_{t-1}, X)), \tag{3.18}$$

where Z_o is a normalization factor over all state sequences. CRFs are thus a function of exponenti-

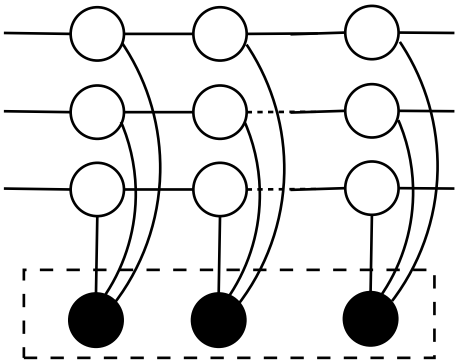


Figure 3.14. Dynamic Conditional Random Fields, which has linear chains of labels with connections between co-temporal labels. Note that the dashed line indicates that the hidden nodes can depend on observations at any time step.

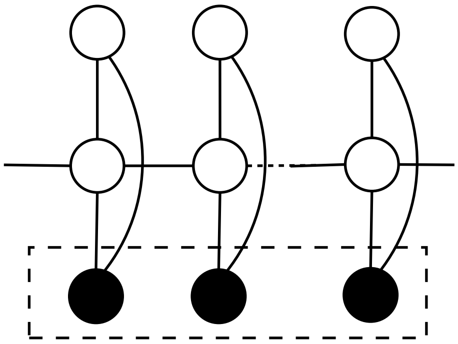


Figure 3.15. Hidden Random Fields (HRF), an undirected version of the IOHMM. The Markov assumption is made regarding the latent nodes and each label node is conditionally independent of all other nodes given its associated latent node. Note that the dashed line indicates that the hidden nodes can depend on observations at any time step.

ated feature functions F_θ, computed in terms of weighted sums over the features of the cliques. In particular, it is often decomposed as follows,

$$F_\theta(y_t, y_{t-1}, X) = \sum_j \lambda_j t_j(y_{t-1}, y_t, X, t) + \sum_k \mu_k s_k(y_t, X, t)), \tag{3.19}$$

where $t_j(y_{t-1}, y_t, X, t)$ is a transition feature function of the entire observation sequence and the labels at positions t and $t-1$ in the label sequence; $s_k(y_t, X, t)$ is a state feature function of the label at position t and the observation sequence; and λ_j and μ_k are parameters to be estimated from training data.

The form of a CRF, as given in Equation 3.18, is heavily motivated by the principle of maximum entropy – a framework for estimating probability distributions from a set of training data.

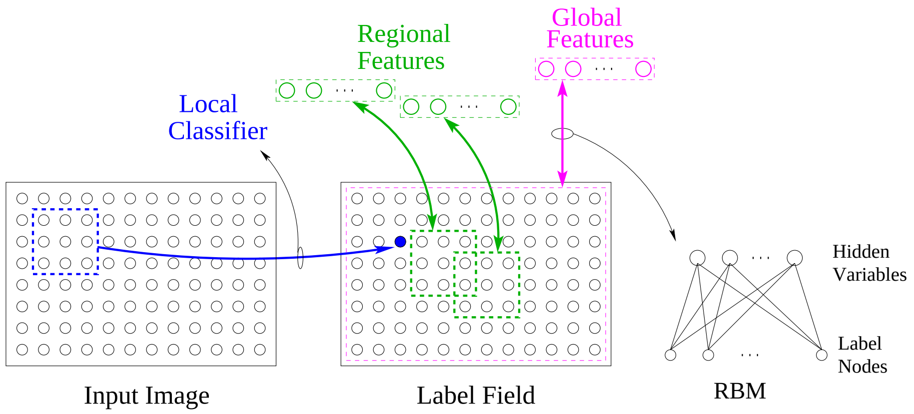


Figure 3.16. Graphical model representation of restricted Boltzmann machines - Conditional Random Fields (RBM-CRF). The local classifier maps image regions to label variables, while the hidden variables corresponding to regional and global features form an undirected model with the label variables. Note that features and labels are fully interconnected, with no intra-layer connections (restricted Boltzmann machine). This figure is from (54).

The *entropy* of a probability distribution is a measure of uncertainty and is maximized when the distribution in question is as uniform as possible. The principle of maximum entropy asserts that the only probability distribution that can justifiably be constructed from incomplete information, such as finite training data, is that which has maximum entropy subject to a set of constraints representing the information available. Any other distribution will involve unwarranted assumptions [62].

Several special cases of conditional random fields are of particular interest.

First, *dynamic conditional random fields* (D-CRF) [123], shown in Figure 3.14, are sequence models which allow multiple labels at each time step, rather than single label as in linear-chain CRFs.

Second, *relational Markov networks* [124] is a type of general CRFs in which the graphical structure and parameter *typing* are determined by an SQL-like (Structured Query Language) syntax.

Third, *hidden random fields* (HRF), shown in Figure 3.15, was introduced by Kakade, Teh and Roweis [65], and can be viewed as an undirected version of the Input-output HMM. The Markov assumption is made regarding the latent nodes and each label node is conditionally independent of all other nodes given its associated latent node.

Last, *restricted Boltzmann machines - Conditional Random Fields* (RBM-CRF), as shown in Figure 3.16, which uses restricted Boltzmann machines (RBMs) [119] to represent the features, was proposed by He, Zemel and Carreira-Perpinan [54] in the context of multi-scale CRFs for image labeling.

3.4 Graphical Model Computations

The three problems for standard HMMs also play central roles in the computation concerning general graphical models.

- **Problem 1 – Likelihood computation:** Given the graphical model M parameterized by θ, how do we compute the likelihood of a sequence:

  $$P(O_1^T|\theta), \tag{3.20}$$

  where $O_1^T = \{o_1, o_2, \cdots, o_T\}$. It can be solved by inference algorithms: exact inference, such as the junction tree algorithm, and approximation inference, such as variational methods [48] and Markov Chain Monte Carlo (MCMC) methods [50].

- **Problem 2 – Decoding:** Given the graphical model M parameterized by θ, how do we choose a state sequence $S = \{s_1, s_2, \cdots, s_T\}$ so that $P(O, S|\theta)$ is maximized.

  $$S^* = \arg\max_S P(O, S|\theta). \tag{3.21}$$

  This is essentially an optimization problem, which can be solved by dynamic programming Viterbi algorithm [125].

- **Problem 3 – Parameter learning:** Given the graphical model M parameterized by θ, including the model structure, how do we best estimate the parameters θ, so that $P(O|\theta)$ is optimized given observation sequences O:

  $$\theta^* = \arg\max_\theta \prod_{i=1}^{N} P(O(i)|\theta). \tag{3.22}$$

  where N is the number of observation sequences. It can be solved using gradient ascent [117], Expectation-Maximization (EM) [31], or generalized Expectation-Maximization (GEM) [93] algorithms.

The algorithmic solutions to those three problems constitute a great amount of research in the field of machine learning. The detailed discussion of the computation for general graphical models is beyond the scope of this thesis. Readers could refer to contents in [38].

3.5 Summary

This chapter briefly reviewed the theory of probabilistic graphical models, and presented various directed and undirected graphical models. We discussed connections between various models that had previously been considered quite different, thus promoting the viewpoint that graphical models provide a unified and useful framework for a large class of problems involving probabilistic inference.

In the next chapters, we will focus on how to develop probabilistic graphical models to solve our three problems – group action modeling, dominance modeling, unusual event modeling. We first introduce the multi-layer framework for group action recognition.

Chapter 4

Group Action Modeling: Recognition

In this chapter, we address the problem of recognizing sequences of human interaction patterns in meetings, with the goal of structuring them in semantic terms. The investigated patterns are inherently group-based (defined by the individual activities of meeting participants, and their interplay), and multimodal (as captured by cameras and microphones). By defining a proper set of individual actions, group actions can be modeled as a two-layer process, one that models basic individual activities from low-level audio-visual features, and another one that models the interactions. We propose a two-layer Hidden Markov Model (HMM) framework that implements such concept in a principled manner, and that has advantages over previous works. First, by decomposing the problem hierarchically, learning is performed on low-dimensional observation spaces, which results in simpler models. Second, our framework is easier to interpret, as both individual and group actions have a clear meaning, and thus easier to improve. Third, different HMM models can be used in each layer, to better reflect the nature of each subproblem. Our framework is general and extensible, and we illustrate it with a set of 14 group actions, using a public five-hour meeting corpus. Experiments and comparison with a single-layer HMM baseline system show its validity.

4.1 Introduction

Devising computational frameworks to automatically infer human behavior from sensors constitutes an open problem in many domains. Moving beyond the person-centered paradigm [120], recent work has started to explore multi-person scenarios, where not only individual but also group actions or interactions become relevant [44, 59, 99, 10].

Group activity plays a key role in meetings [126, 87], and this is documented by a significant amount of work in social psychology [83]. Viewed as a whole, a group shares information, engages in discussions, and makes decisions, proceeding through diverse communication phases both in single meetings and during the course of a long-term teamwork [83]. Recognizing group actions is therefore useful for browsing and retrieval purposes [126, 80], e.g., to structure a meeting into a sequence of high-level items.

Interaction in meetings is inherently group-based [83] and multimodal [69]. In the first place, we can view a meeting as a continuous sequence of mutually exclusive group actions taken from an exhaustive set [80, 33]. Each of these group actions involves multiple simultaneous participants, and is thus implicitly constrained by the actions of the individuals. In the second place, as the principal modality in meetings, speech has recently been studied in the context of interaction modeling [56, 131, 33]. However, work analyzing the benefits of modeling individual and group actions using multiple modalities has been limited [10, 80, 81, 100], despite the fact that actions in meetings, both at the individual (e.g., note-taking or talking), and at the group level (e.g. dictating) are often defined by the joint occurrence of specific audio and visual patterns.

In this chapter, we present a two-layer HMM framework for group action recognition in meetings. The fundamental idea is that, by defining an adequate set of individual actions, we can decompose the group action recognition problem into two levels, from individual to group actions. Both layers use HMMs or extensions. The goal of the lower layer is to recognize individual actions of participants using low-level audio-visual (AV) features. The output of this layer provides the input to the second layer, which models interactions. Individual actions naturally constitute the link between the low-level audio-visual features and high-level group actions. Similarly to continuous automatic speech recognition, we perform group action recognition directly on the data sequence, deriving the segmentation of group actions in the process. Our approach is general, extensible, and brings improvement over previous work, which reflects on the results obtained on a public meeting corpus, for a set of 14 group actions based on multimodal turn-taking patterns.

This chapter is organized as follows. Section 4.2 introduces the multi-layer framework. Sections 4.3 and 4.4 describe the meeting data, and the feature extraction process. Experiments and discussion are presented in Section 4.5. Conclusions are drawn in Section 4.6.

4.2 Group Action Recognition

In this section, we first introduce our computational framework. We then apply it to a specific set of individual and group actions. Finally, we describe some specific details.

4.2.1 Framework Overview

Our framework is based on the use of Hidden Markov Models (HMMs) and some of their extensions. HMMs have been used with success for numerous sequence recognition tasks, including speech recognition [106]. HMMs introduce a hidden state variable and factorize the joint distribution of a sequence of observations and the state using two simpler distributions, namely emission and transition distributions. Such factorization yields efficient training algorithms such as the Expectation-Maximization algorithm (EM) [31] which can be used to select the set of parameters to maximize the likelihood of several observation sequences. In our work, we use Gaussian Mixture Models (GMMs) to represent the emission distribution.

The success of HMMs applied to sequences of actions is based on a careful design of sub-models (distributions) corresponding to lexical units (phonemes, words, letters, actions). Given a training set of observation sequences representing meetings for which we know the corresponding labeling (but not necessarily the precise alignment), we create a new HMM for each sequence as the concatenation of sub-model HMMs corresponding to the sequence of actions. This new HMM can then be trained using EM and will have the effect of adapting each sub-model HMM accordingly. When a new sequence of observation features of a meeting becomes available, the objective is to obtain the optimal sequence of sub-model HMMs (representing actions) that could have generated the given observation sequence. An approximation of this can be done efficiently using the well-known Viterbi algorithm [125]. This process therefore leads to the recognition of actions directly on the

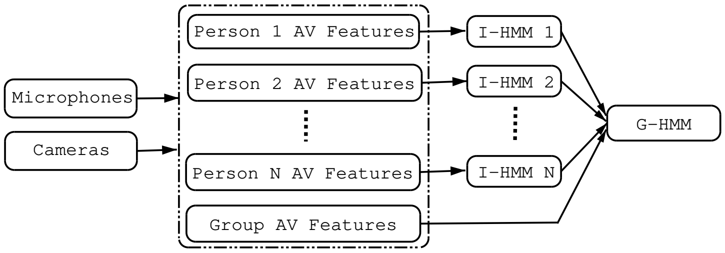


Figure 4.1. The two-layer framework applied to meeting action recognition: the lower layer recognizes individual actions of participants using low-level audio-visual (AV) features. The output of this layer provides the input to the second layer, which models interactions. Individual actions naturally constitute the link between the low-level audio-visual features and high-level group actions.

data sequence, generating the action boundaries in the process.

In our framework, we distinguish group actions (which belong to the whole set of participants) from individual actions (belonging to specific persons). Our ultimate goal is the recognition of group activity, and so individual actions should act as the bridge between group actions and low-level features, thus decomposing the problem in stages. The definition of both action sets is thus clearly intertwined.

Let I-HMM denote the lower recognition layer (individual action), and G-HMM denote the upper layer (group action). I-HMM receives as input AV features extracted from each participant, and outputs recognition results, either as soft or hard decisions (Section 4.2.4). In turn, G-HMM receives as input the output from I-HMM, and a set of *group features*, directly extracted from the raw streams, which are not associated to any particular individual. In our framework, each layer is trained independently, and can be substituted by any of the HMM variants that might capture better the characteristics of the data, more specifically asynchrony [11], or different noise conditions [36] between the audio and visual streams. Our approach is summarized in Figure 4.1. The training procedure is described in later sections.

Compared with a single-layer HMM, the layered approach has the following advantages, some of which were previously pointed out by [97]: (1) a single-layer HMM is defined on a possibly large observation space, which might face the problem of over-fitting with limited training data. It is important to notice that the amount of training data becomes an issue in meetings where data labeling is not a cheap task. In contrast, the layers in our approach are defined over small-dimensional observation spaces, resulting in more stable performance in cases of limited amount of training data. (2) The I-HMMs are person-independent, and in practice can be trained with much more data from different persons, as each meeting provides multiple individual streams of training data. Better generalization performance can then be expected. (3) The G-HMMs are less sensitive to slight changes in the low-level features because their observations are the outputs of the individual action recognizers, which are expected to be well trained. (4) The two layers are trained independently. Thus, we can explore different HMM combination systems. In particular, we can replace the baseline I-HMMs with models that are more suitable for multi-modal asynchronous data sequences, with the goal of gaining understanding of the nature of the data (Section 4.2.3). The framework thus becomes simpler to understand, and amenable to improvements at each separate level. (5) The framework is general and extensible to recognize new group actions defined in the future.

4.2.2 Definition of Actions

As an implementation of the proposed framework, we define a set of group actions and individual actions in this section. On one hand, a set of group actions is defined based on *multi-modal turn-taking patterns* [81]. A solid body of work in social psychology has confirmed that, in the context of group discussions, speaker turn patterns convey a rich amount of information about the dynamics of the group and the individual behaviour of its members, including trends of influence, dominance, and interest [83, 102, 40]. While speaking turns are obviously described mainly by audio information, significant information also exists in non-verbal cues. Work in the literature has studied how participants coordinate speaking turns via an ensemble of multimodal cues, such as gaze, speech back-channels, changes in posture, etc. [102, 95]. From a different perspective, recognizing multi-modal group turn-taking is also useful for meeting structuring, for access and retrieval purposes.

The list of group actions is defined in Table 4.1. Note that we consider a "monologue" or a "presentation" as a group action, because we define it as the joint occurrence of several individual patterns (e.g., one person speaks while the others listen to her). For meeting browsing and indexing, it might be also desirable to know which specific participant is doing a monologue in the meeting. Therefore, we further divide the "monologue" action into "monologue1", "monologue2", etc., according to the number of participants. In a similar way, we divide the "monologue+note-taking" action into "monolgue1+note-taking", "monologue2+note-taking", and so on. Thus, for a four-participant meeting, a set of $N_G = 14$ group actions has been defined as: N_G = { *discussion, monologue1, monologue1 + note-taking, monologue2, monologue2 + note-taking, monologue3, mono-*

Table 4.1. Description of group actions

Discussion	most participants engaged in conversations
Monologue	one participant speaking continuously without interruption
Monologue+ Note-taking	one participant speaking continuously others taking notes
Note-taking	most participants taking notes
Presentation	one participant presenting using the projector screen
Presentation+ Note-taking	one participant presenting using projector screen, others taking notes
White-board	one participant speaking using the white-board
White-board+ Note-taking	one participant speaking using white-board, others taking notes

logue3 + note-taking, monologue4, monologue4 + note-taking, note-taking, presentation, presentation + note-taking, whiteboard, whiteboard + note-taking }. These group actions are multimodal, and commonly found in meetings. For modeling purposes, they are assumed to define a partition (i.e., the action set is non-overlapping and exhaustive). This set is richer compared to the one that we defined in [81], as it includes simultaneous occurrence of actions, like *"monologue+note-taking"* which could occur during real situations, like dictating or minute-taking. The group actions we defined here can be easily described by combinations of a proper set of individual actions defined in the following. Our framework is general, and other type of group actions could be defined. Note that high-level group actions in semantic terms (e.g. agreement / disagreement) would certainly require language-based features [56].

On the other hand, we define a small set of $N_I = 3$ multimodal individual actions which, as stated earlier, will help bridge the gap between group actions and low-level AV features. The list appears in Table 4.2. While the list of potentially interesting individual actions in meetings is large, our ultimate goal is recognition of the group-level actions defined in Table 4.1. It is interesting to note that, although at first glance one would not think of *"speaking"* or *"writing"* as multimodal, joint sound and visual patterns do occur in these cases and are useful in recognition, as the results in later sections confirm.

Finally, meeting rooms can be equipped with white-boards or projector screens which are shared by the group. Extracting features from these group devices also helps recognize group actions. They constitute the group features described in the previous subsection. Their detailed description will

Table 4.2. Description of individual actions

Speaking	one participant speaking
Writing	one participant taking notes
Idle	one participant neither speaking nor writing

Table 4.3. Relationships between group actions, individual actions and group features. Symbol "$\star$" indicates that the white-board or projector screen are in use when the corresponding group action takes place. Symbol "/" indicates that the number of participants for the corresponding action is not certain. The numbers (0,1,...) indicate the number of involved meeting participants in the group action

Group Actions	Individual Actions			Group Features	
	speaking	writing	idle	white-board	projector
discussion	>2	/	/		
monologue	1	0	/		
monologue+note-taking	1	>=1	/		
note-taking	0	>2	0		
presentation	1	0	/		$\star$
presentation+note-taking	1	>=1	/		$\star$
white-board	1	0	/	$\star$	
white-board+note-taking	1	>=1	/	$\star$	

be presented in section 4.4.

The logical relations between individual actions, group actions, and group features are summarized in Table 4.3. The group actions can be seen as combinations of individual actions plus states of group devices. For example, *"presentation + note-taking"* can be decomposed into *"speaking"* by one individual, with more than one *"writing"* participant, while the group device of *projector screen* is in use. Needless to say, our approach is not rule-based, but Table 4.3 is useful to conceptually relate the two layers.

In the next sections, we present some details about the architecture of our framework. To facilitate description, we first define the following symbols:

- $\mathbf{O}^a$: a sequence of audio-only feature vectors.
- $\mathbf{O}^v$: a sequence of visual-only feature vectors.
- $\mathbf{O}^{a+v}$: a sequence of concatenated audio-visual feature vectors.
- $\mathbf{o}_{1:t} \triangleq \mathbf{o}_1, \mathbf{o}_2, ..., \mathbf{o}_t$: a sequence (audio, visual, or audio-visual stream) up to time t.
- q_t: the HMM state at time t

4.2.3 Individual Action Models

We investigate three models for the lower-layer I-HMM, each of which attempts to model specific properties of the data. The investigated models are:

Early Integration HMM (Early Int.), where a basic HMM [106] is trained on combined AV features. This method involves aligning and synchronizing the AV features to form one concatenated set of features which is then treated as a single stream of data. The concatenation simply defines the audio-visual feature space as the cartesian product of the audio and video feature spaces, creating vectors which first contain the components of the audio feature vector, followed by the components of the video feature vector. Early integration selects the set of parameters θ_i^* of the model corresponding to action i that maximizes the likelihood of L audio-visual observation sequences as follows:

$$\theta_i^* = \arg\max_{\theta_i} \prod_{l=1}^{L} P(\mathbf{O}_l^{a+v}|\theta_i). \tag{4.1}$$

Audio-visual Multi-Stream HMM (MS-HMM), which combines the audio-only and visual-only streams. Each stream is modeled independently. $\theta_i^* = (\theta_{i,a}^*, \theta_{i,v}^*)$ are the best model parameters for action i to maximize the likelihood of audio-only and visual-only sequences respectively.

$$\theta_{i,a}^* = \arg\max_{\theta_{i,a}} \prod_{l=1}^{L} P(\mathbf{O}_l^{a}|\theta_{i,a}). \tag{4.2}$$

$$\theta_{i,v}^* = \arg\max_{\theta_{i,v}} \prod_{l=1}^{L} P(\mathbf{O}_l^{v}|\theta_{i,v}). \tag{4.3}$$

For testing, we create a special HMM by recombing all the single-stream HMM likelihoods at various specific temporal points. Depending on these recombination points, various solutions appear. When the models are recombined after each state, the underlying system is equivalent to making the hypothesis that all streams are state-synchronous and independent of each other given the state. A more powerful recombination strategy enables some form asynchrony between the states of each stream: one could consider an HMM in which states could include all possible combinations of the single-stream HMM states. However, this strategy quickly becomes intractable since the total number of states of this model would be exponential in the number of streams. We use the first combination strategy in our experiments.

The final classification is based on the fusion of the outputs of both modalities by estimating their joint occurrence [36], as follows:

$$P(\mathbf{O}_l^{a+v}|q_t) = P(\mathbf{O}_l^a|q_t, \theta_{i,a})^{\omega} P(\mathbf{O}_l^v|q_t, \theta_{i,v})^{(1-\omega)}, \tag{4.4}$$

where the weighting factor $\omega (0 \leq \omega \leq 1)$ represents the relative reliability of the two modalities.

Audio-visual Asynchronous HMM (A-HMM), which also combines audio-only and visual-only streams, by learning the joint distribution of pairs of sequences when these sequences are not synchronized and are not of the same length or rate [11]. This situation could occur in the meeting scenario at the group level when, for instance, an individual starts playing her role before the rest of the group. A similar situation could happen at the individual level between the audio and visual streams. For instance, it is known that the movements of the face are not synchronized with the actually uttered speech of a person [79]. Furthermore, in a conversational setting, a person tends to move before taking a turn, and often stops gesticulating before finishing speaking as a turn-yielding signal [35]. Being able to stretch some streams with respect to others at specific points could thus yield performance improvement The A-HMM for action i models the joint distribution of the two streams by maximizing the likelihood of L observation sequences as follows:

$$\theta_i^* = \arg\max_{\theta_i} \prod_{l=1}^{L} P(\mathbf{O}_l^a, \mathbf{O}_l^v|\theta_i). \tag{4.5}$$

Furthermore, while normal HMM optimization techniques integrate the likelihood of the data over all possible values of the hidden variable (which is the value of the state at each time step), asynchronous HMMs also integrate this likelihood over all possible alignments between observation sequences, adding a new hidden variable $\tau_t = s$ meaning that observation $\mathbf{o}_t^a$ is aligned with observation $\mathbf{o}_s^v$. With the hidden variable τ_t and using several reasonable independence assumptions, [11] can factor the joint likelihood of the data and the hidden variables into several simple conditional distributions, which makes the model tractable using the EM algorithm. The Viterbi algorithm can be used to obtain the optimal state sequence as well as the alignment between the two sequences.

4.2.4 Linking the Two Layers

Obviously, a mechanism to link the two HMM layers has to be specified. There are two approaches to do so, based on different I-HMM outputs. Let $a^t = (a_1^t, ..., a_{N_I}^t) \in \mathbb{R}^{N_I}$ denote a vector in a continuous space of dimension equal to the number of individual actions, which indicates the degree of confidence in the recognition of each individual action at time t for a sequence $\mathbf{o}_{1:t}$.

The first approach directly outputs the probability P_k^t for each individual action model $M_k, k = 1, ..., N_I$, as input feature vector to G-HMM, $a_k^t = P_k^t$ for all k. We refer to it as *soft decision*.

In soft decision, we define P_k^t as the posterior probability of model M_k given a sequence $\mathbf{o}_{1:t}$. It can be computed in two steps. In the first step, we compute the probability of having generated the sequence and being in the state i at time t. We denote this probability as $\rho(i,t)$. For different I-HMMs, the probability $\rho(i,t)$ is computed in different ways.

- *Early integration normal HMM*:

$$\rho(i,t) = P(\mathbf{o}_{1:t}, q_t = i), \tag{4.6}$$

  which is equivalent to the forward variable $\alpha(i,t) \triangleq P(\mathbf{o}_{1:t}, q_t = i)$ in the standard Baum-Welch algorithm [106]. $\mathbf{o}_{1:t}$ could be audio-only, visual-only or audio-visual stream.

- *Multi-stream HMM:* $\rho(i,t)$ is calculated as follows,

$$\rho(i,t) = P(\mathbf{o}_{1:t}^a, \mathbf{o}_{1:t}^v, q_t = i) \tag{4.7}$$

$$\tag{4.8}$$

  which is estimated in the multi-stream context by

$$\rho(i,t) = P(\mathbf{o}_{1:t}^a, q_t = i)^\omega P(\mathbf{o}_{1:t}^v, q_t = i)^{1-\omega}, \tag{4.9}$$

  where $\mathbf{o}_{1:t}^a$ is the audio-only sequence and $\mathbf{o}_{1:t}^v$ is the visual-only sequence. ω is the weighting factor defined in Equation (4.4).

- *Asynchronous HMM:* $\rho(i,t)$ is calculated as follows,

$$\rho(i,t) = \sum_{s=t-\triangle t}^{t+\triangle t} P(\mathbf{o}_{1:t}^{a}, \mathbf{o}_{1:s}^{v}, q_t = i, \tau_t = s), \tag{4.10}$$

where $\triangle t$ is the size of a sliding window centered at current time t. The variable $\tau_t = s$ can be seen as the alignment between sequence $\mathbf{o}_{1:t}^{a}$ (audio-only stream) and $\mathbf{o}_{1:s}^{v}$ (visual-only stream).

In the second step, we normalize the probability $\rho(i,t)$ for all states of all the models in order to obtain the posterior for each model. The probability $P(q_t = i|\mathbf{o}_{1:t})$ of state i given a sequence $\mathbf{o}_{1:t}$ is

$$\begin{aligned}
P(q_t = i|\mathbf{o}_{1:t}) &= \frac{P(q_t = i, \mathbf{o}_{1:t})}{P(\mathbf{o}_{1:t})} &(4.11)\\
&= \frac{P(q_t = i, \mathbf{o}_{1:t})}{\sum_{j=1}^{N_S} P(q_t = j, \mathbf{o}_{1:t})} &(4.12)\\
&= \frac{\rho(i,t)}{\sum_{j=1}^{N_S} \rho(j,t)}, &(4.13)
\end{aligned}$$

where N_s is the number of all states for all models.

With this, the probability P_k^t of model M_k given a sequence $\mathbf{o}_{1:t}$ is then computed as

$$\begin{aligned}
P_k^t &= \sum_{i \in M_k} P(q_t = i|\mathbf{o}_{1:t}) &(4.14)\\
&= \sum_{i \in M_k} \frac{\rho(i,t)}{\sum_{j=1}^{N_S} \rho(j,t)}, &(4.15)
\end{aligned}$$

where i is the state in model M_k, which is a subset of the states of all models. The probability P_k^t of model M_k is the sum of the probabilities of all states in model M_k.

In the second approach, the individual action model with the highest probability outputs a value of 1, while all other models output a zero value. The vector a^t generated in this way is used as input to G-HMM. We refer to it as *hard decision*.

We concatenate the individual recognition vectors from all participants, together with the group-level features, into a $(N_I \times N_P + N_{GF})$-dimensional vector (where N_P is the number of participants, and N_{GF} is the dimension of the group features) as observations to G-HMM for group action recognition.

4.2.5 Group Action Models

We investigate two models for the group action layer: one is the standard HMM, and another is conditional random fields (CRFs). We have described basic ideas of conditional random fields in Section 3.3.1.

Recently, there has been an explosion of interest in conditional random fields, as a successful alternative to HMMs. Conditional random fields are theoretically more powerful, and have the following advantages over HMMs:

- An HMM is a *generative* model, assigning a joint probability to paired observation and label (or hidden state) sequences. Thus, the HMM parameters are typically trained to maximize the joint likelihood of training samples. CRF is a *conditional* model that is trained to maximum the posterior probability of the label sequence given the observation sequence.

- HMMs must make very strict independence assumptions on the observations given the current state for computational tractability. CRFs relax such assumption. CRFs have the great flexibility to include a wide variety of arbitrary, non-independent features as input.

4.3 Meeting Database

We used the publicly available meeting corpus first described in [80], which was collected in a meeting room equipped with synchronized multi-channel audio and video recorders (publicly available at `http://mmm.idiap.ch/`). The sensors include three fixed cameras and twelve microphones [86]. Two cameras have an upper-body, frontal view of two participants including part of the table. A third wide-view camera captures the projector screen and white-board. Audio was recorded using lapel microphones for all participants, and an eight-microphone array placed in the center of the table. The complex nature of the audio-visual information present in meetings will be better appreciated by looking directly at the above website. A snapshot of the three camera views, and the visual feature extraction is shown in Figure 4.2. The corpus consists of 59 short meetings of five-minute average duration, with four participants per meeting. The group action structure was scripted before recording, so part of the group actions labels we define were already available as part of the corpus. However, we needed to relabel the rest of the group actions (e.g. *monologues* into

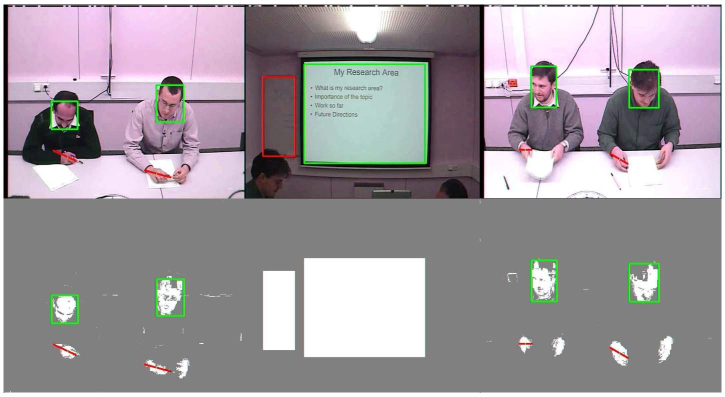


Figure 4.2. Multi-camera meeting room and visual feature extraction

either *monologues* or *monologues+note-taking*), and to label the entire corpus in terms of individual actions. All ground-truth was produced using *Anvil*, a publicly available video annotation tool (`http://www.dfki.de/~kipp/anvil/`).

4.4 Feature Extraction

In this section, we describe the process to extract the two types of AV features: person-specific AV features and group-level AV features. The former are extracted from individual participants. The latter are extracted from the whiteboard and projector screen regions.

4.4.1 Person-Specific AV Features

Person-specific visual features were extracted from the cameras that have a close view of the participants. Person-specific audio features were extracted from the lapel microphones attached to each person, and from the microphone array. The complete set of features is listed in Table 4.4.

Person-specific visual features. For each video frame, the raw image is converted to a skin-color likelihood image, using a 5-component skin-color Gaussian mixture model (GMM). We use the chromatic color space, known to be less variant to the skin color of different people [136]. The chromatic colors are defined by a normalization process: $r = \frac{R}{R+G+B}, g = \frac{G}{R+G+B}$. Skin pixels were then classified based on thresholding of the skin likelihood. A morphological postprocessing step was performed to remove noise. The skin-color likelihood image is the input to a connected-component algorithm (flood filling) that extracts blobs. All blobs whose areas are smaller than a given threshold were removed. We use 2-D blob features to represent each participant in the meeting, assuming that the extracted blobs correspond to human faces and hands. First, we use a multi-view face detector to verify blobs corresponding to the face. The blob with the highest confidence output by the face detector is recognized as the face. Among the remaining blobs, the one that has the rightmost centroid horizontal position is identified as the right hand (we only extracted features from the right hands since the participants in the corpus are predominately right-handed). For each person, the detected face blob is represented by its vertical centroid position and eccentricity [120]. The hand blob is represented by its horizontal centroid position, eccentricity, and angle. Additionally, the motion magnitude for head and right hand are also extracted and summed into one single feature.

Person-specific audio features. Using the microphone array and the lapels, we extracted two types of person-specific audio features. On one hand, speech activity was estimated at four seated locations, from the microphone array waveforms. The seated locations are expressed as 3-D vectors in Cartesian coordinates, measured with respect to the microphone array in our meeting room. These vectors correspond to the location where people are typically seated. One measure was computed per seat location. The speech activity measure coming from each seated location was the SRP-PHAT (Steered Response Power-Phase Transform) measure, an increasingly popular technique used for acoustic source localization due to its suitability for reverberant environments [32]. SRP-PHAT is a continuous value that indicates the speech activity at a particular location. On the other hand, three acoustic features were estimated from each lapel waveform: energy, pitch and speaking rate. We computed these features on speech segments, setting a value of zero on silence segments. Speech segments were detected using the microphone array, because it is well suited for multiparty speech. We used the SIFT algorithm [78] to extract pitch, and a combination of estimators [88] to extract speaking rate.

4.4.2 Group AV Features

Group AV features were extracted from the white-board and projector screen regions. Given the constrained topology of a real meeting room, most people will naturally tend to occupy the same regions when making a presentation or using the whiteboard. The features are listed in Table 4.4.

Table 4.4. Audio-visual feature list

		Description
Person-Specific Features	Audio	SRP-PHAT from each seat
		speech relative pitch
		speech energy
		speech rate
	Visual	head vertical centroid
		head eccentricity
		right hand horizontal centroid
		right hand angle
		right hand eccentricity
		head and hand motion
Group Features	Audio	SRP-PHAT from white-board
		SRP-PHAT from projector screen
	Visual	mean difference from white-board
		mean difference from projector screen

Group visual features. These were extracted from the camera that looks towards the white-board and projector screen area. We first get difference images between a reference background image and the image at each time, in the white-board and projector screen regions (Figure 4.2). On these difference images, we use the average intensity over a grid of 16×16 blocks as features.

Group audio features. These are SRP-PHAT features extracted using the microphone array from two locations corresponding to the white-board and projector screen.

4.5 Experiments

In this section, we first describe the measures used to evaluate our results, and then present results for both individual action recognition and group action recognition.

4.5.1 Performance Measures

We use the *action error rate* (AER) and the *frame error rate* (FER) as measures to evaluate the results of group action recognition and individual action recognition, respectively.

AER is equivalent to the word error rate widely used in speech recognition, and is defined as the minimum sum of insertions (Ins: symbols that were not present in a ground truth sequence, but were decoded in the recognized sequence), deletions (Del: symbols that were present in a ground truth sequence, but were not decoded in the recognized sequence), and substitution (Sub: symbols that were present in a ground truth sequence, but were decoded as a different symbol in the recognized sequence), divided by the total number of actions in the ground-truth,

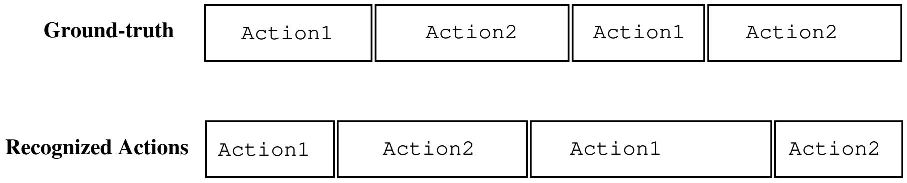


Figure 4.3. AER is not a meaningful assessment for small number of actions.

AER $= \frac{\text{Sub+Del+Ins}}{\text{total actions}} \times 100\%$. This is obtained by applying a dynamic programming technique comparing the obtained sequence of labels with the expected one. It is also often called the "edit distance" or the "Levenshtein distance".

For group action recognition, we have $N_G = 14$ possible actions which in many cases have no clear-cut temporal boundaries. Furthermore, at least five actions occur in each meeting in the corpus. We believe that AER is thus a good measure to evaluate group action recognition, as we are more interested in the recognition of the correct action sequence rather than the precise time alignment of the recognized action segments.

However, AER overlooks the time alignment between recognized and target action segments. For individual action recognition, there are only $N_I = 3$ possible actions. Furthermore, some streams (participants) in the corpus consist of only two individual actions (e.g., a person who talks only once during the course of a meeting). AER might not provide a meaningful assessment in such cases. As shown in Figure 4.3, AER equals zero because the recognized actions and the ground-truth actions have the same sequential order. But obviously, the result in Figure 4.3 is not perfect. Therefore, it is necessary to verify that the temporal alignment of the recognized actions with another measure, especially for the case in which the total number of actions is small.

In this view, we adopt FER as the performance measure for individual action recognition. FER is defined as one minus the ratio between the number of correctly recognized frames and the number of total frames, FER $= (1 - \frac{\text{correct frames}}{\text{total frames}}) \times 100\%$. This measure reflects well the accuracy of the boundaries (begin and end time) of the recognized actions, compared to manually labeled action boundaries.

With limited number of training and testing actions, results are likely to vary due to the random initialization of the training procedure (based on Expectation-Maximization [106]). For this reason, and to assess consistency in the results, we report the mean and standard deviation (STD) for AER and FER, computed over 10 runs with random initialization of the EM procedure.

Finally, we also use confusion matrices, whose rows and columns index the recognized and ground-truth actions, respectively. The element c_{ij} of the confusion matrix corresponds to either the percentage (for individual actions) or the instances (for group actions) of action j recognized as action i. The confusion matrix for group actions is based on AER, so there are substitution, insertion, and deletion errors. For individual actions, there are neither insertions nor deletions because the performance measure is FER.

4.5.2 Experimental Protocol

For both individual and group action recognition, we use 6-fold cross-validation on the training set to select the values of the model parameters that are not estimated as part of the EM algorithm. In a HMM/GMM architecture, these include the number of states per action, and the number of components (Gaussians) per state. In 6-fold cross-validation, we divided the data into 6 subsets of approximately equal size. We then train the models six times with different parameter configurations, each time leaving out one of the subsets from training, and using only the omitted subset to compute the corresponding performance measure (FER for individual actions, AER for group actions). The parameters resulting in the best overall performance were selected, and used to re-train the models on the whole training set.

For group actions, as described in [80], two disjoint sets of eight people each, whose identities were known, were used to construct the training and test sets. Each meeting was recorded using a randomly chosen 4-person combination within each of the sets. With this choice, no person appears in both the training and the test set. For individual actions, the original 8-people set in the training set was further split into two disjoint subsets at each time during the cross-validation procedure. One of these subsets was used to extract the streams belonging to the training set. The other subset was used to create the validation set. With this choice, we ensure that the data extracted from the same person is not used to both train and validate the individual action models.

From the 59 meetings, 30 are used as training data, and the remaining 29 are used for testing. The number of frames (N_F) and number of actions (N_A) for individual action and group action in the different data sets are summarized in Table 4.5. The number of individual actions is much larger than that of group actions. There are two reasons. First, for individual action recognition,

Table 4.5. Number of frames (N_F) and number of actions (N_A) in different data sets.

Individual Actions	train		test	
	N_F	N_A	N_F	N_A
speaking	35028	1088	33747	897
writing	15803	363	27365	390
idle	127569	1426	112488	1349
total	178400	2877	173600	2636
Group Actions	**train**		**test**	
	N_F	N_A	N_F	N_A
discussion	17760	48	14450	49
monologue	7615	26	7585	26
monologue + note-taking	6260	17	6695	23
note-taking	640	6	320	3
presentation	3170	6	3345	9
presentation + note-taking	3455	5	3865	9
white-board	2155	5	265	1
white-board + note-taking	3545	11	6875	19
total	44600	124	43400	139

there are four participants for each meeting. Therefore, there are $30 \times 4 = 120$ streams for training and $29 \times 4 = 116$ streams for testing. Second, the duration of individual actions is typically shorter than that of group actions.

4.5.3 Individual Action Recognition

The three methods described in Section 4.2.3 were tested for individual action recognition.

- **Early integration (Early Int.),** trained on three feature sets: audio-only, visual-only. and audio-visual.
- **Audio-visual multi-stream HMM (MS-HMM),** combining individual audio and visual streams. Audio and visual streams are modeled independently. The final classification is based on the fusion of the outputs of both modalities by estimating their joint occurrence (Section 4.2.3).
- **Audio-visual asynchronous HMM (A-HMM),** combining individual audio and visual streams by learning the joint distribution of pairs of sequences when these sequences are not synchronized (Section 4.2.3).

Multi-stream HMMs allow us to give different weights to different modalities (see Equation 4.4). Following the discussion presented in [81], we use (0.8,0.2) to weight the audio and visual modalities, respectively. For asynchronous HMM, the allowed asynchrony ranges from $\pm 2.2s$ using simple reasonable a priori knowledge.

Table 4.6. Results of individual action recognition

Method	Features	FER (%)	STD
Early Int.	Visual	34.17	3.64
	Audio	23.48	2.70
	Audio-visual	9.98	2.65
MS-HMM	Audio-visual	8.58	1.76
A-HMM	Audio-visual	7.42	1.13

The summary of the results for all the individual action recognition models is presented in Table 4.6, in terms of FER mean and standard deviation, obtained over 10 runs (as described earlier, each run starts with a random initialization of the EM training procedure).

From Table 4.6, we observe that all methods using AV features produced less than 10% FER, which is about 15% absolute improvement over using audio-only features, and about 25% absolute improvement over using visual-only features. Asynchronous HMM produced the best result. Given that the total number of frames is over $43,000$, the improvement using asynchronous HMM over the other HMM methods is statistically significant with a confidence level above 99%, using a standard proportion test [49]. The improvement suggests that there exist asynchronous effects between the audio and visual modalities. Additionally, we tested the MS-HMM system with an equal-weight scheme (0.5, 0.5). The performance decreased compared to the MS-HMM with larger weight on audio (0.8 and 0.2, see earlier discussion). This is not surprising given the predominant role of audio in the defined actions.

The confusion matrices for visual-only, audio-only, and audio-visual streams, corresponding to a randomly chosen single run, are shown in Tables 4.7, 4.8, and 4.9, respectively. We can see that *"speaking"* is well detected using audio-only features, and that *"writing"* is well detected using visual-only features. Using audio-visual features, both *"speaking"* and *"writing"* are generally well detected. Using AV features, *"writing"* tends to get confused with *"idle"*, which in turn is the action with the highest FER. This is likely due to the catch-all role that this action plays. In practice, *"idle"* includes all other possible AV patterns, (e.g. pointing, laughing, etc.), which makes its modeling more difficult, compared with the other two well-defined actions.

In order to empirically investigate asynchronous effects in the individual actions, we performed forced alignment decoding on the audio-only and visual-only streams independently. A similar approach was taken to establish empirical evidence for asynchrony in multi-band automatic speech recognition in [85]. The decoder in each stream was constrained by the ground-truth individual

Table 4.7. Confusion matrix of recognized individual actions (using visual-only features) Rows: recognized actions. Columns: ground-truth

	Speaking	Writing	Idle
Speaking	51.92%	3.00%	8.22%
Writing	45.87%	85.93%	34.65%
Idle	2.21%	11.07%	57.13%

Table 4.8. Confusion matrix of recognized individual actions (using audio-only features) Rows: recognized actions. Columns: ground-truth

	Speaking	Writing	Idle
Speaking	91.74%	1.26%	1.78%
Writing	1.16%	35.23%	22.10%
Idle	7.10%	63.51%	76.12%

action sequence, and so the output action sequences differ only in their temporal boundaries. We calculated the time misalignment (start-time difference of corresponding actions) between the two sequences. Actions having absolute misalignments larger than $5s$ were discarded, as the misalignments were more likely caused by recognition errors, rather than asynchronous effects. Figure 4.5.3 shows the resulting histogram of misalignments, assumed due to asynchronous effects, for these individual actions. The histogram can be approximated by a Gaussian distribution, with a mean of $-0.13s$ (effectively zero, as misalignments happened in both directions) and a standard deviation of 2.05. More than 80% of the individual actions are distributed in the range of $\pm 2.2s$ (defined at the beginning of this section), while there are 17% individual actions without any asynchronous effects ($P(t = 0) = 17\%$). This suggests that, for most individual actions having evidence in both streams, allowing asynchrony between streams should more accurately model the data.

4.5.4 Group Action Recognition

Using the outputs from I-HMM and the group-level features, concatenated as described in Section 4.2.4, we investigated a number of cases for recognition of group actions listed as follows.

Table 4.9. Confusion matrix of recognized individual actions (using AV features) Rows: recognized actions. Columns: ground-truth

	Speaking	Writing	Idle
Speaking	94.23%	2.12%	4.73%
Writing	1.03%	89.60%	10.89%
Idle	4.74%	8.28%	84.38%

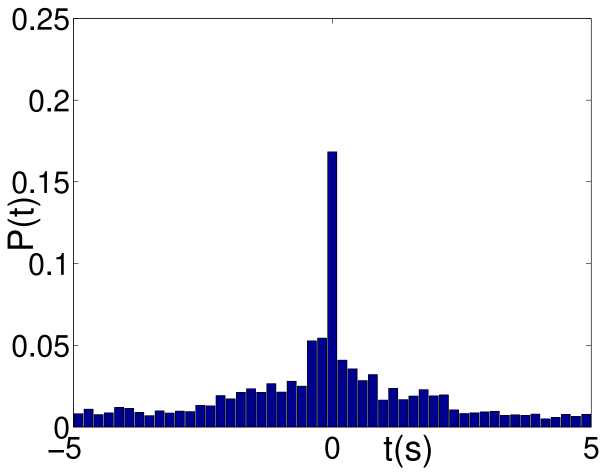


Figure 4.4. Histogram of asynchronous effects of individual actions

- **Early integration, visual-only, soft decision**. A normal HMM is trained using the combination of the results of the I-HMM trained on visual-only features, and the visual group features. The soft decision criteria is used.

- **Early integration, audio-only, soft decision**. Same as above, but replacing visual-only by audio-only information.

- **Early integration, AV, hard decision**. Same as above, but replacing visual-only by audio-visual information. The hard decision criteria is used.

- **Early integration, AV, soft decision**. Same as above, but changing the criteria to link two HMM layers.

- **Multi-stream, AV, hard decision**, using the multi-stream HMM approach as I-HMM. The hard decision criteria is used.

- **Multi-stream, AV, soft decision**. Same as above, but changing the criteria to link two HMM layers.

- **Asynchronous HMM, AV, hard decision**. We use the asynchronous HMM for individual action layer and audio-visual features. The hard decision criteria is used.

- **Asynchronous HMM, AV, soft decision**. Same as above, but changing the criteria to link two HMM layers.

Table 4.10. Results of group action recognition

Method			AER (%)	STD
Single-layer HMM	Visual		48.20	3.78
	Audio		36.70	4.12
	Audio-visual		23.74	2.97
Multi-layer HMM	Visual		42.45	2.85
	Audio		32.37	2.10
	Early Int.	hard	17.98	2.75
		soft	16.55	1.40
	MS-HMM	hard	17.27	2.01
		soft	15.83	1.61
	A-HMM	hard	17.85	2.87
		soft	15.11	1.48
	HMM + CRFs		15.01	1.43

- **Early integration HMM, AV, hard decision, conditional random fields**. We use early integration HMM trained on audio-visual features as individual action layer. The output state sequence resulting from Viterbi decoding serves as the input to the group action layer implemented by conditional random fields (CRFs).

As baseline methods for comparison, we tested single-layer HMMs, using low-level audio-only, visual-only, and AV features as observations [80], and trained by cross-validation following the same experimental protocol. The results appear in Table 4.10, in terms of AER mean and standard deviation over 10 runs.

We observe from Table 4.10 that the use of AV features outperformed the use of single modalities for both single-layer and multi-layer methods. This result supports the hypothesis that the group actions we defined are inherently multimodal. Furthermore, the best two-layer HMM method (A-HMM) using AV features improved the performance by over 8% compared to the AV single-layer HMM. Additionally, the standard deviation for the two-layer approach is half the baseline's, which suggests that our approach might be more robust to variations in initialization, given the fact that each HMM stage in our approach is trained using an observation space of relatively low dimension. Regarding hard vs. soft decision, soft decision produced a slightly better result, although not statistically significant given the number of group actions. However, the standard deviation using soft-decision is again around half the corresponding one using hard-decision. The overall best result so far we could get is using the *HMM + CRFs* framework. This result confirms the advantage of CRFs over HMMs, as already shown by previous work on many sequence segmentation and labeling tasks [124, 103, 117, 114, 77, 71, 54].

Table 4.11. Confusion matrix of recognized group actions for single-layer HMM using audio-visual features. Rows: recognized actions. Columns: ground-truth

	D	M1	M1+N	M2	M2+N	M3	M3+N	M4	M4+N	N	P	P+N	W	W+N	Del
D	45														1
M1	2	6	3												
M1+N			3												
M2				6	1										
M2+N				2	3										1
M3						2									1
M3+N						3	7								
M4								2							
M4+N								3	5						
N										2					1
P											6	5			1
P+N											1	3			
W	1					1					1		1	2	
W+N												1		17	
Ins					1		1								

To further analyze results, we provide the confusion matrices for single-layer HMM using AV features, and two-layer HMM using AV, soft-decision and asynchronous HMM in Tables 4.11 and 4.12, respectively. We showed discussion (D), monologue ($M1 \cdots M4$), monologue+note-taking (M1+N, $\cdots$, M4+N), note-taking (N), presentation (P), presentation+note-taking (P+N), white-board (W), and white-board+note-taking (W+N). Empty cells represent zero values. It is evident that the two-layer method greatly reduced the number of errors, compared with the single-layer method. For both matrices, we see that most substitution errors come from confusions between actions with and without note-taking. This might be mainly because several instances of *"writing"* could not be reliably detected as individual actions, as mentioned in the previous subsection. There are several *"presentation"* actions confused with *"white-board"*, which might be because some speakers moved around the white-board and projector-screen regions during a presentation. On the other hand, *"discussion"* and *"note-taking"* actions can be recognized reasonably well.

4.6 Conclusions

In this chapter, meetings were defined as sequences of multi-modal group actions. We addressed the problem of group action recognition, proposing a multi-layer framework to decompose it into two layers. The first layer maps low-level audio-visual features into individual actions. The second layer

Table 4.12. Confusion matrix of recognized group actions for two-layer HMM (using asynchronous HMM with soft decision). Rows: recognized actions. Columns: ground-truth

	D	M1	M1+N	M2	M2+N	M3	M3+N	M4	M4+N	N	P	P+N	W	W+N	Del
D	44														1
M1	2	6	2												
M1+N			4												
M2				7											
M2+N				1	5										
M3						5									1
M3+N						1	6								
M4								4							
M4+N								1	5						
N							1			3					
P											6				
P+N												8			
W	2										2	1	1	1	
W+N											1			18	
Ins				1								1	1		

uses results from the first layer as input to recognize group actions. Experiments demonstrate the effectiveness of the framework to recognize a set of 14 multimodal turn-taking actions, compared to a baseline, single-layer HMM system.

In reality, meetings are not restricted to pre-defined action sets. Furthermore, high-level group actions in meetings can be ambiguous (and expensive) to label. In this view, modeling group actions with unsupervised approaches, which find “action structure” in either individual meetings or whole meeting collections, without the need for labeled data or previous knowledge of the actions, become very attractive options. This is the subject of the next chapter.

Chapter 5

Group Action Modeling: Clustering

In this chapter, we address the problem of group action clustering using the two-layer HMM framework described in Chapter 4. The goal is to create one cluster for each group action, where the number of group actions and the action boundaries are unknown a priori. The results show that the use of multiple modalities and the layered framework are advantageous, compared to various baseline methods.

5.1 Introduction

A meeting can be seen as proceeding through phases, where a group disseminates information, discusses, and makes decisions [83]. In this view, a simple model can thus be used to define a meeting as a continuous sequence of group actions (i.e., involving multiple simultaneous participants) chosen from one or more pre-defined action dictionaries, which is well suited for supervised learning, as we did in Chapter 4, as long as the action dictionaries are well defined. This implies that the actions comprising each dictionary should be mutually exclusive, exhaustive, and unambiguous to human observers, at least to a degree for which manually labeled data for supervised learning can be reliably generated.

In reality, however, meetings are not restricted to pre-defined action sets. Furthermore, high-level group actions in meetings can be ambiguous (and expensive) to label. Roughly speaking, the degree of ambiguity correlates with the actions' level of semantic meaning. Basic actions like writing or speaking can be clearly identified, group actions like discussions are more ambiguous, and high-level actions like information sharing might be very difficult to label reliably, which could seriously challenge supervised methods.

In this view, modeling high-level group actions with unsupervised approaches, which find "action structure" in either individual meetings or whole collections, without the need for labeled data or previous knowledge of the actions, become very attractive options [137, 141], especially given the vast amount of data that is generated in many real cases. Given adequate features, clustering an individual meeting could partition it into action-consistent segments. Clustering an entire collection could further find action-consistent clusters across meetings. Additionally, unsupervised methods could naturally deal with variations (e.g. in the number of participants) that would otherwise need to be modeled explicitly in supervised methods.

In this chapter, we apply the layered HMM framework for group action clustering in meetings, as an alternative to fully supervised methodologies. In our view, our methodology constitutes an attractive option for analysis of high-level group actions in meetings, due to its potential to deal with actions that would otherwise be difficult to pre-define and/or expensive to label.

The multi-layer framework, action lexicon, audio-visual feature extraction process and individual action models have been described in Chapter 4. The rest of this chapter is organized as follows. Section 5.2 introduces group action clustering. Section 5.3 presents experiments and discussion. Concluding remarks are provided in Section 5.4.

5.2 Group Action Clustering

For group action clustering, we employ an agglomerative clustering algorithm, recently proposed in the speech community for speaker clustering [3], and that has shown good performance for such a task. The algorithm is based on an ergodic HMM framework with a minimum duration constraint, where the number of clusters and segmentation boundaries are unknown a priori. Each state of the HMM represents a cluster having several identical states in cascade in order to impose the minimum duration constraint. A three-cluster case is illustrated in Figure 5.1.

The HMM clustering algorithm can be summarized as follows:

1. **Initialization:** Start by over-clustering, i.e. clustering the data into a number of clusters

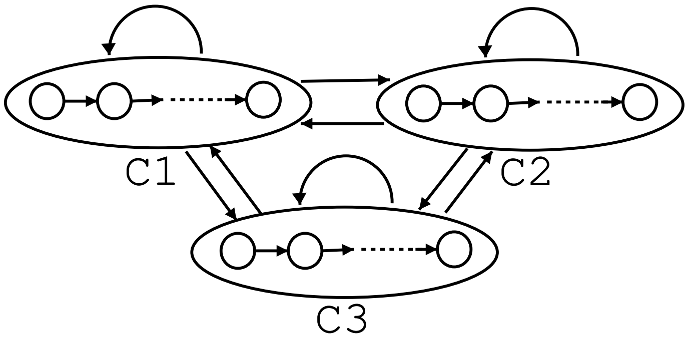


Figure 5.1. The ergodic HMM topology with minimum duration constraint.

larger than the hypothesized number of actions. The probability density function of each cluster is represented by a Gaussian Mixture Model (GMM) and the parameters of this GMM are estimated using the expectation maximization (EM) algorithm. The initialization for each distribution is done using K-means.

2. **Segmentation:** Obtain the segmentation automatically using the Viterbi algorithm [106] on the current HMM topology and parameters.

3. **Training:** Re-estimate the parameters of all clusters based on this segmentation.

4. **Merging:** Search for the best candidate pair of clusters for merging based on the criterion as follows [3]:

   - Let M_a and M_b represent the number of parameters (Gaussian components) in the PDFs of these two clusters (D_a and D_b) respectively.
   - Let us hypothesize a new cluster having data $D = D_a \bigcup D_b$ with a PDF modeled by a GMM parameterized by θ with $M_a + M_b$ number of Gaussian components.
   - Given the above conditions, a pair of clusters (D_a and D_b) becomes a candidate for merging if the following is true:

$$\log p(D|\theta) \geq \log p(D_a|\theta_a) + \log p(D_b|\theta_b). \tag{5.1}$$

The segmentation-training-merging process is iterated until no more cluster pairs satisfy the merging criterion.

The HMM clustering algorithm has a number of advantages [3]. First, the final number of clusters is decided automatically using a robust merging criterion. Secondly, instead of making local threshold-based decisions, the HMM clustering algorithm produces a global segmentation of the meeting video without using any pre-defined threshold, which is optimal in the maximum likelihood sense, while avoiding the need for development data. Thirdly, the clustering algorithm can be applied directly on the data sequences, deriving the segmentation in the process without assumptions regarding the number of clusters and their boundaries.

The clustering algorithm can be applied to one individual meeting, as well as to a complete meeting collection, with a minor difference. When clustering a collection, the features for all meetings are concatenated. However, the inter-meeting boundaries are known a priori, so this particular knowledge is used as part of the clustering process.

5.3 Experiments and Results

In this section, we describe our experiments and results. The meeting data set and audio-visual feature extraction process have been described in Chapter 4. Since clustering is a different task from recognition, we first present the performance measures used to evaluate our results. We then present results for group action clustering and discuss our findings.

5.3.1 Performance Measures

Two measures (*action error rate* and *frame error rate*) were proposed to evaluate results of supervised continuous group action recognition in [33, 81]. However, these measures cannot be used for unsupervised group action clustering because the labels of the clusters are unknown. Instead, we use three measures used in the field of speaker clustering to evaluate our results: *average cluster purity* (acp), *average action purity* (aap) and overall evaluation criterion K [2, 3]. These measures are explained below. First we define:

- n_{ij}: total number of frames in cluster i by action j
- $n_{\bullet j}$: total number of frames of action j
- $n_{i\bullet}$: total number of frames in cluster i
- N_a: total number of actions

- N_c: total number of clusters
- N: total number of frames

The purity of a cluster $p_{i\bullet}$ and the *acp* are defined as

$$p_{i\bullet} = \sum_{j=1}^{N_a} \frac{n_{ij}^2}{n_{i\bullet}^2}, \tag{5.2}$$

$$acp = \frac{1}{N} \sum_{i=1}^{N_c} (p_{i\bullet} \times n_{i\bullet}). \tag{5.3}$$

Similarly, the action purity $p_{\bullet j}$ and the *aap* are given by

$$p_{\bullet j} = \sum_{i=1}^{N_c} \frac{n_{ij}^2}{n_{\bullet j}^2}, \tag{5.4}$$

$$aap = \frac{1}{N} \sum_{j=1}^{N_a} (p_{\bullet j} \times n_{\bullet j}). \tag{5.5}$$

The *acp* gives a measure of how well a cluster is limited to only one action, while the *aap* gives a measure of how well one action is limited to only one cluster. In the ideal case, *i.e.* one cluster for each group action, the value of both *average action purity (aap)* and *average cluster purity (acp)* is one: $acp = 1, aap = 1$.

However, from only *acp* or *aap* taken separately, it is hard to evaluate the overall performance because *acp* can achieve a high value with more clusters than really required, and *aap* can achieve a high value with less clusters. In the extreme case, *acp=1* if a cluster has only one frame and *aap=1* if there is only one cluster for the whole meeting. In order to facilitate comparison between systems, an overall evaluation criterion K is defined as follows, where larger K indicates better overall performance.

$$K = \sqrt{acp \times aap}. \tag{5.6}$$

As a percentage, the average criterion K is around 70% for the robust speaker clustering algorithm described in [2].

Table 5.1. Clustering results for individual meetings

Method	N_c *mean*	N_c *std*	*aap* (%)	*acp* (%)	*K* (%)
two-layer HMM					
Visual	6.20	2.19	41.4	77.0	56.8
Audio	3.10	1.12	71.3	56.1	63.7
Early Int.	3.59	0.95	69.5	71.3	70.1
MS-HMM	4.17	1.13	72.7	70.8	71.8
A-HMM	3.51	0.78	78.6	70.0	73.8
Baseline: single-layer HMM					
Visual	8.72	2.17	33.6	76.1	50.6
Audio	3.03	1.94	61.1	57.8	58.6
AV	4.10	1.35	68.8	64.2	65.7
Baseline: true number of clusters ($N_c = N_a$)					
B_1	3.93	0.73	64.3	60.1	62.1
B_2			78.4	70.9	74.1
$B_2 - 1$	2.93	0.73	83.5	62.7	71.8
$B_2 + 1$	4.93	0.73	72.6	70.9	71.1

5.3.2 Results and Discussion

To test our approach, we investigated the following combinations of modalities and models for the lower layer,

- **Early integration, visual-only**. The clustering algorithm was applied on the concatenation of the results produced by an early integration *I-HMM* trained on visual-only features, and the visual group features.
- **Early integration, audio-only**. Same as above, but replacing visual-only by audio-only information.
- **Early integration, AV**. Same as above, but using AV data.
- **Multi-stream, AV**. Same as above, but using the *MS-HMM* in individual action layer.
- **Asynchronous, AV**. Same as above, but using the *A-HMM*.

Additionally, to analyze the benefit of the layered approach, we investigated a number of single-layer clustering schemes, which use the same clustering algorithm directly applied on the low-level features (visual, audio, and AV).

The performance regarding model selection (i.e., determination of the number of clusters) was also studied. We define two baseline systems based on K-means (B_1), and HMM clustering (B_2)

respectively, which model an "ideal" case, in which the final number of clusters is exactly the same as the number of group actions (as indicated by the ground-truth). For these systems, the model used for the lower layer was *A-HMM*, as it produced the best performance for the two-layer method (see discussion below).

Finally, we investigated two clustering cases. In the first case, we cluster group actions for each meeting. Usually, the number of group actions within one meeting is less than the complete set of eight actions. In the second case, we cluster the whole test meeting collection, which produces a segmentation for each meeting where segments belonging to the same cluster get consistent labels across the corpus. In this case, there are eight group actions.

Parameter Selection. For the individual action layer, parameters were selected by six-fold cross-validation, splitting the training set into training and validation subsets and selecting the values of the hyper-parameters that maximize K (see Equation 5.6) in the validation sets. For the group action layer, we obtained results by varying the number of initial clusters (10-30), the number of Gaussians (5-10), and the minimum duration of each cluster (15-30s). In Tables 5.1-5.2, the results for the number of clusters *(N_c)* are shown in terms of mean and standard deviation. We report mean values for average action purity *(aap)* and cluster purity *(acp)*, and for the overall criterion (K). The results can be summarized as follows.

Single- vs. multi-modality and single- vs. two-layer HMM. For both the single- and the two-layer cases, the use of AV features produced better results than using only one modality. Audio-only features were more discriminant than video-only, which is not surprising given the type of group actions we addressed. We noticed that methods using audio features got high *aap* and low *acp* while methods using video features showed the opposite trend. This is because, according to the ground-truth, the number of clusters (N_c) was usually underestimated using audio, while over-estimated using visual features. Audio-only features thus seem to be described better by simpler models, while visual-only features describe a more complex cluster structure. Additionally, the layered approach outperformed the single-layer method under the same conditions (using one or multiple modalities, and when clustering individual meetings or the whole data set). Given the large total number of frames ($> 43,000$), these improvements are statistically significant, which confirms the effectiveness of the layered approach, and the multimodal nature of group actions in meetings.

Comparison between I-HMM methods. The analysis of the performance of the various *I-HMMs* for individual action recognition is described in detail in [139], but overall, the asynchronous HMM produced the best results. Regarding group action clustering, although multi-stream HMM improved over early integration, the asynchronous HMM also produced the best results among all HMM systems for the two meeting clustering cases. This indicates that the probability-based features obtained from this model were more discriminative, and suggests the presence of asynchrony between the audio-visual streams for individual actions. In Tables 5.1-5.2, both *acp* and *asp* of *A-HMM* are above 70%. This means that more than 70% of all group actions are in the right clusters, while more than 70% of all clusters are composed of data from the same group action.

Comparison with "ideal" baseline systems. The layered method using AV features outperformed the K-means baseline (B_1), while performed slightly worse than HMM clustering baseline (B_2). We can also see that with a slight increase/decrease of the number of clusters, the performance of this baseline system decreased. (In Tables 5.1-5.2, "$B_2 - 1$" and "$B_2 + 1$" denote the baseline system, in which we deliberately increase or decrease the number of clusters by 1.) Interestingly, the best two-layer HMM method outperforms these two cases, which somewhat suggests that our approach is not too far from the "ideal" case.

Single meeting vs. entire meeting collection. The results of clustering the whole collection are slightly worse than the results of clustering single meetings for the multimodal layered models (between 1.6-2.0%); the degradation is more pronounced for the single-modality approaches. This decrease in the clustering quality could be explained by the larger variation in the data (the number of meeting participants in the test set taken as a whole is 10), but mainly by the increasing possibility of overlap between different group actions in the feature space, due to the larger number of actions. Note however that clustering the whole corpus generates consistent action labels across meetings; this important benefit was traded by the decrease in performance.

Model selection. For both individual meeting clustering and whole collection clustering, the methods using AV features obtained a number of clusters closer to the true number of actions. For the first case, there are 3.93 group actions on average in the ground-truth. The average number of clusters found using AV features ranges from 3.51 to 4.17, which is close to the true number (Table 5.1). For the second case, there are 8 group actions in the ground-truth. The two-layer AV systems *MS-HMM* and *A-HMM* both converged around 7 clusters (Table 5.2), which is in good accordance

Table 5.2. Clustering results for meeting collection

Method	N_c		*aap (%)*	*acp (%)*	*K (%)*
	mean	*std*			
two-layer HMM					
Visual	11.67	2.16	31.0	47.2	38.2
Audio	3.50	2.65	77.7	41.4	56.7
Early Int.	10.60	1.93	70.9	65.7	68.3
MS-HMM	7.28	1.41	74.8	65.2	69.8
A-HMM	7.10	1.70	74.0	70.5	72.2
Baseline: single-layer HMM					
Visual	16.33	4.08	20.2	46.3	30.6
Audio	3.16	2.40	76.1	33.6	50.6
AV	6.73	2.51	64.3	60.1	62.1
Baseline: true number of clusters ($N_c = N_a$)					
B_1	8	0	47.3	51.1	49.2
B_2			71.5	73.8	72.6
$B_2 - 1$	7	0	74.5	67.1	70.7
$B_2 + 1$	9	0	63.9	78.5	70.8

with the true number, although slightly underestimated.

To evaluate the quality of the clustering results, we display the found clusters and ground-truth actions in Fig.5.2, for the best 13 test meetings ranked by decreasing order, based on the criterion K (the symbol $M_{\#}$ is the meeting index in the test set). Dashed-line rectangles denote automatic clusters (with labels $\{1, 2, ...\}$), which are compared against the ground-truth actions denoted by solid-line rectangles, showing *discussion* (D), *monologue* (M), *monologue + note-taking* (MN), *note-taking* (N), *presentation* (P), *presentation + note-taking* (PN), *white-board* (W) and *white-board + note-taking* (WN). The left and right columns of Fig.5.2 show the results of clustering individual meetings and the entire meeting collection, respectively. For both cases, we can see that for meetings with large overall criterion K, the obtained alignments between clusters and actions are better. The results degrade with decreasing K. Notice that, for the case of clustering the meeting collection (Fig.5.2: right-column), cluster labels are consistent across meetings. For example, most clusters with label "3" correspond to "MN" (*monologue + note-taking*) group action, and clusters with label"1" often correspond to the "D" (*discussion*) action.

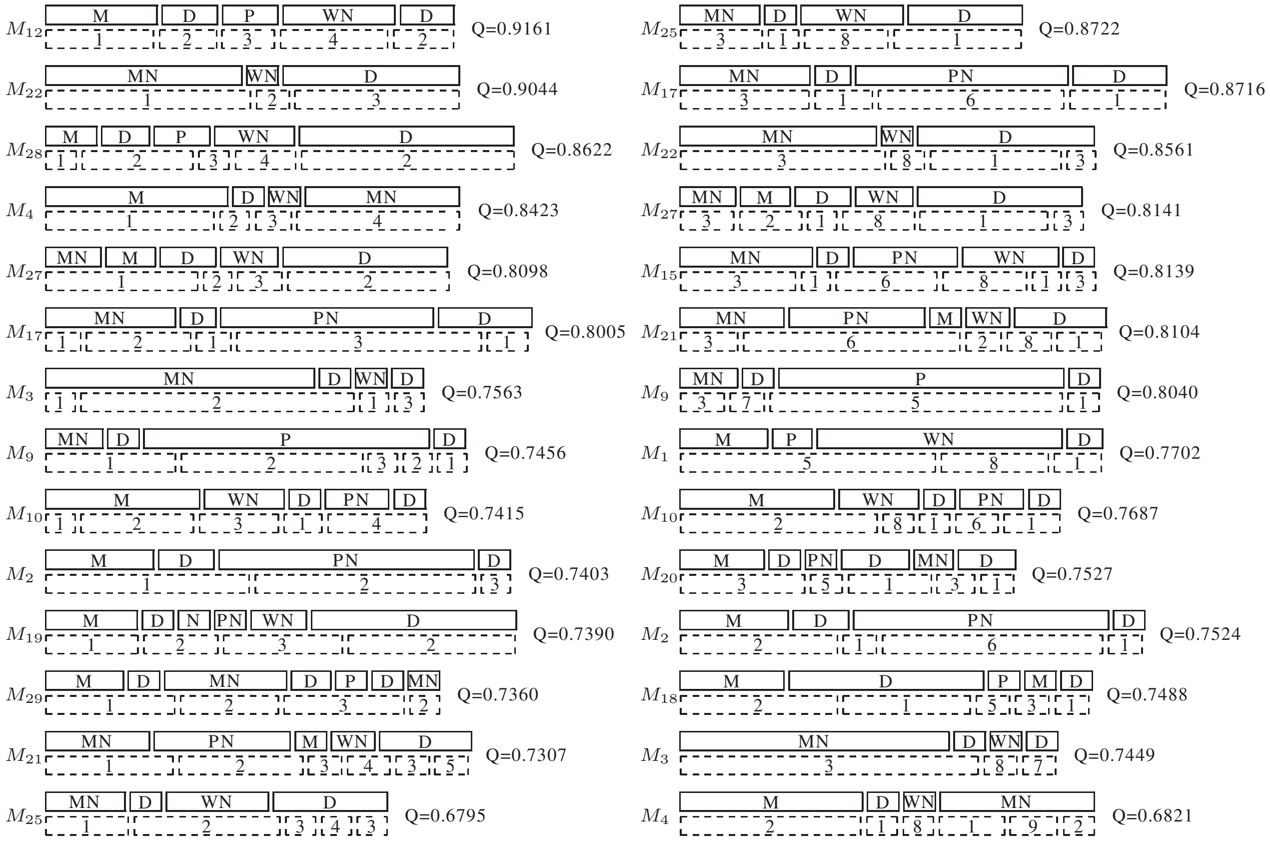


Figure 5.2. Results of clustering individual meetings (left column), and entire meeting collection (right column). Clustering an individual meeting could partition it into action-consistent segments. Clustering an entire collection could further find action-consistent clusters across meetings.

5.4 Conclusions

In this chapter, we addressed the problem of group action clustering using the layered HMM framework. The first layer maps low-level audio-visual features into probability-based, individual-action features. The second layer groups such features into clusters, which correspond reasonably well to group actions.

In meetings, some people seem particularly capable of driving the conversation and dominating its outcome. These people, skilled at establishing the leadership, have the largest influence on the group decisions, and often shift the focus of the meeting when they speak. In the next chapter, we will describe a novel dynamic Bayesian network that quantitively determines how much influence each participant has on the outcome of the meeting.

Chapter 6

Dominance Modeling

In this chapter, we present a model that learns the influence of interacting Markov chains within a team. The proposed model is a dynamic Bayesian network (DBN) with a two-level structure: individual-level and group-level. The individual level models actions of each player, and the group-level models actions of the team as a whole. Unlike existing multi-stream models, the influence of each player on the team, which can also be interpreted as reliability in the case of data fusion, is jointly learned with the rest of the model parameters in a principled manner using the Expectation-Maximization (EM) algorithm. Experiments on two data sets: synthetic multi-player game, and multi-party meetings show the effectiveness of the proposed model. In the first experiment, we demonstrate that our model can learn reasonable influence values for players in various synthetic multi-player games. In the second experiment, we learn the influence of each participant in meetings using acoustic and language features. We show that the learned influence distributions are in good accordance with the influence distributions from human judgements.

6.1 Introduction

In multi-agent systems, individuals within a group coordinate and interact to achieve a goal. For instance, consider a basketball game where a team of players with different roles, such as attack and defense, collaborate and interact to win the game. Each player performs a set of individual actions, evolving based on their own dynamics. A group of players interact to form a team. Actions of the team and its players are strongly correlated, and different players have different influence on the team. Taking another example, in conversational settings, some people seem particularly capable of driving the conversation and dominating its outcome. These people, skilled at establishing the leadership, have the largest influence on the group decisions, and often shift the focus of the meeting when they speak [39]. This problem of determining how much influence one person has on others has been studied in the context of multi-party conversations [9] and wearable computing [24].

In this chapter, we propose a novel dynamic Bayesian network, that we call *the team-player influence model* for multi-stream sequence modeling. The proposed model has a two-level structure: In the first level, we model actions of individual players. In the second one, we model team actions as a whole. A fundamental aspect of our model is that it explicitly learns the influence of individual players on the team. The learned influence values have different semantic meanings for different applications. We illustrate this on two datasets: synthetic multi-player games (Section 6.5), and multi-party meetings (Section 6.6) for different purposes (1) a multi-player game dataset with the goal to determine the influence of players in games, and (2) a meeting dataset aiming at determine the influence of participants in multi-party meetings.

This chapter is organized as follows. Section 6.2 introduces the team-player influence model. Section 6.3 reviews related models. Section 6.4 discusses implementation issues. Section 6.5 presents results on multi-player games. Section 6.6 presents results on a meeting corpus. Section 6.7 provides concluding remarks.

6.2 The Team-Player Influence Model

The proposed model, called the team-player influence model, is a dynamic Bayesian network (DBN) with a two-level structure: the *player* level and the *team* level (Figure 6.1). The player level represents the actions of individual players, evolving based on their own Markovian dynamics (Figure 6.1 (a)). The team level represents group-level actions (the action belongs to the team as a whole, not to a particular player). In Figure 6.1 (b), the arrows up (from players to team) represent the influence of the individual actions on the group actions, and the arrows down (from team to players) represent the influence of the group actions on the individual actions.

Let O^i and S^i denote the observation and state of the i^{th} player respectively, and S^G denotes the team state. For N players, and observation sequences of identical length T, according to Figure 6.1, the joint distribution of our model is given by

$$P(S,O) = \prod_{i=1}^{N} P(S_1^i) \cdot \prod_{t=1}^{T}\prod_{i=1}^{N} P(O_t^i|S_t^i) \cdot \prod_{t=1}^{T} P(S_t^G|S_t^1 \cdots S_t^N) \cdot \prod_{t=2}^{T}\prod_{i=1}^{N} P(S_t^i|S_{t-1}^i, S_{t-1}^G). \tag{6.1}$$

Regarding the player level, we model the actions of each individual with a first-order Markov model (Figure 6.1 (a)) with one observation variable O^i and one state variable S^i. Furthermore, to capture the dynamics of all the players interacting as a team, we add a hidden variable S^G (team state), which is responsible to model the group-level actions. Different from individual player states that have their own Markovian dynamics, the team state is not directly influenced by its previous state. S^G could be seen as the aggregate behavior of the individuals, yet it provides a useful level of description beyond individual actions. There are two kinds of relationships between the team and players:

1. The team state at time t influences the players' states at the next time (down arrow in Figure 6.1 (b)). In other words, the state of the i^{th} player at time $t+1$ depends on its previous state as well as on the team state, i.e., $P(S_{t+1}^i|S_t^i, S_t^G)$.

2. The team state at time t is influenced by all the players' states at the current time (up arrow in Figure 6.1 (b)), resulting in a conditional state transition distribution $P(S_t^G|S_t^1 \cdots S_t^N)$.

We add one hidden variable Q in the model, to switch parents for S^G. The idea of switching parent (also called Bayesian multi-nets [16] explained in Chapter 3) is as follows: a variable -S^G in this case- has a set of parents $\{Q, S^1 \cdots S^N\}$ (Figure 6.1(c)). Q is the switching parent that determines which of the other parents to use, conditioned on the current value of the switching parent. $\{S^1 \cdots S^N\}$ are the conditional parents. In Figure 6.1(c), Q switches the parents of S^G

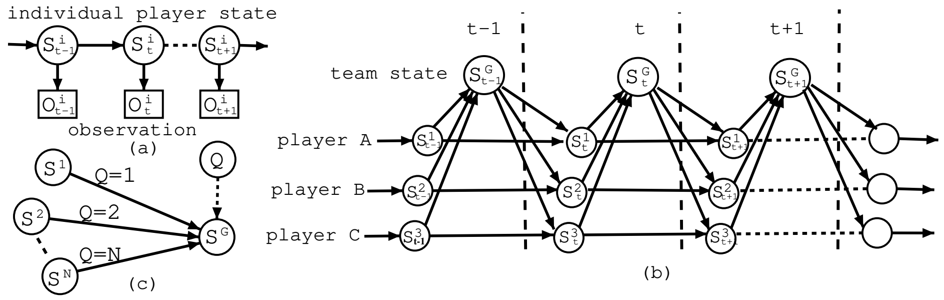


Figure 6.1. (a) Markov Model for individual player. (b) The team-player influence model (for simplicity, we omit the observation variables of individual Markov chains, and the switching parent variable Q). (c) Switching parents. Q is called a switching parent of S^G, and $\{S^1 \cdots S^N\}$ are conditional parents of S^G. When $Q = i$, S^i is the only parent of S^G.

among $\{S^1 \cdots S^N\}$, corresponding to the distribution,

$$P(S_t^G|S_t^1 \cdots S_t^N) = \sum_{i=1}^{N} P(S_t^G, Q = i|S_t^1 \cdots S_t^N) \tag{6.2}$$

$$= \sum_{i=1}^{N} P(Q = i|S_t^1 \cdots S_t^N) P(S_t^G|S_t^i \cdots S_t^N, Q = i) \tag{6.3}$$

$$= \sum_{i=1}^{N} P(Q = i) P(S_t^G|S_t^i) = \sum_{i=1}^{N} \alpha_i P(S_t^G|S_t^i). \tag{6.4}$$

From Equation 6.3 to Equation 6.4, we made two assumptions: (i) Q is independent of $\{S^1 \cdots S^N\}$; and (ii) when $Q = i$, S_t^G only depends on S_t^i. The distribution over the switching-parent variable $P(Q)$ essentially describes how much influence or contribution the state transitions of the player variables have on the state transitions of the team variable. We refer to $\alpha_i = P(Q = i)$ as the influence value of the i^{th} player. Obviously, $\sum_{i=1}^{N} \alpha_i = 1$, *i.e.*, the sum of contributions of all players equals 1.

A fundamental aspect of our model is that the α_i values are automatically learned from data (see Section 6.4). The learned α_i values have different semantic meanings for different applications. We illustrate this on three applications: synthetic multi-player games and multi-party meetings (see sections 6.5, 6.6). For synthetic multi-player games, α_i indicates whether player i plays a leading (or following) role in games. In multi-party meetings, α_i represents the influence of each participant in meetings.

It is also worthwhile to discuss more about the meaning of the team state S^G. This hidden variable models the team state, and acts as a "bottleneck" node that reduces the complexity of the model from a combinatorial number of pairwise links (that would model full inter-player interactions) to a linear number. In the generative model, at each time-step the team state is emitted by only one player, chosen based on the influence prior. This is an operation that might not be computed by pair-wise interactions without enumerating all of them. In addition, each player is further affected by the team state. Since the model was trained in an unsupervised manner, there is no guarantee that this hidden variable will have a direct semantic interpretation. The variable takes values selected to maximize the overall likelihood.

6.3 Related Models

The proposed team-player influence model is related to a number of models, namely mixed-memory Markov model (MMM) [115, 68], coupled HMM (CHMM) [98], influence model [5, 9, 24], dynamical systems trees (DSTs) [60], layered HMM [97, 138], and multi-stream HMM [36]. MMMs decompose a complex model into mixtures of simpler ones, for example, a K-order Markov model, into mixtures of first-order models: $P(S_t|S_{t-1}S_{t-2}\cdots S_{t-K}) = \sum_{i=1}^{K}\alpha_i P(S_t|S_{t-i})$. The CHMM models interactions of multiple Markov chains by directly linking the current state of one stream with the previous states of all the streams (including itself): $P(S_t^i|S_{t-1}^1 S_{t-1}^2\cdots S_{t-1}^N)$. However, the model becomes computationally intractable for more than two streams. The influence model [5, 9, 24] simplifies the state transition distribution of the CHMM into a convex combination of pairwise conditional distributions, i.e., $P(S_t^i|S_{t-1}^1 S_{t-1}^2\cdots S_{t-1}^N) = \sum_{j=1}^{N}\alpha_{ji}P(S_t^i|S_{t-1}^j)$. We can see that influence model and MMM take the same strategy to reduce complex models with large state spaces to a combination of simpler ones with smaller state spaces. In [9, 24], the influence model was used to analyze speaking patterns in conversations (i.e., turn-taking) to determine how much influence one participant has on others. In such model, α_{ji} is regarded as the influence of the j^{th} player on the i^{th} player.

All these models, however, limit themselves to modeling the interactions between individual players, *i.e.,* the influence of *one player on another player*. The proposed team-player influence model extends these models by using the group-level variable S^G that allows to model the influence between *all the players and the team*: $P(S_t^G|S_t^1 S_t^2\cdots S_t^N) = \sum_{i=1}^{N}\alpha_i P(S_t^G|S_t^i)$, and additionally conditioning the dynamics of each player on the team state: $P(S_{t+1}^i|S_t^i, S_t^G)$.

DSTs [60] have a tree structure that models interacting processes through the parent hidden Markov chains. There are two differences between DSTs and our model: (1) In DSTs, the parent chain has its own Markovian dynamics, while the team state of our model is not directly influenced by the previous team state. Thus, our model captures the emergent phenomena in which the group action is "nothing more" than the aggregate behaviors of individuals, yet it provides a useful level of representation beyond individual actions. (2) The influence between players and team in our model is "bi-directional" (up and down arrows in Figure 6.1(b)). In DSTs, the influence between child and parent chains is "uni-directional": parent chains can influence child chains, while child chains cannot influence their parent chains.

The layered HMMs were proposed to model multimodal office activities [97], and multimodal group actions in meetings (see Chapter 4). The basic idea of layered HMMs is to decompose the complex action recognition problem into layered architectures. Each layer, which is trained independently, uses ergodic HMMs or extensions. The output of the lower layer provides the input to the upper layer. Similar to the layered HMMs, the team-player influence model has a two-level structure: *players* and *team*. Different from layered HMMs, the team-player influence model can nicely capture interactions between the team and players as (1) the team state influences the players' states at the next time, and (2) the team state is influenced by all the players' states at the current time. Furthermore, the team-player model is trained jointly, while each layer of the layered HMM is trained independently.

6.4 Implementation Issues

For simplicity, let us assume that all player variables have the same number of states N_S, and the team variable has N_G possible states. The joint log probability is then given by,

$$\begin{aligned}\log P(S,O) = &\underbrace{\sum_{i=1}^{N}\sum_{j=1}^{N_S} z_{j,1}^{i}\cdot\log P(S_1^i=j)}_{initial\ probability} + \underbrace{\sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{j=1}^{N_S} z_{j,t}^{i}\cdot\log P(O_t^i|S_t^i=j)}_{emission\ probability}\\ &+\underbrace{\sum_{t=2}^{T}\sum_{i=1}^{N}\sum_{j=1}^{N_S}\sum_{k=1}^{N_S}\sum_{g=1}^{N_G} z_{j,t}^{i}\cdot z_{k,t-1}^{i}\cdot z_{g,t-1}^{G}\cdot\log P(S_t^i=j|S_{t-1}^i=k,S_{t-1}^G=g)}_{group\ influence\ on\ individual\ transition}\\ &+\underbrace{\sum_{t=1}^{T}\sum_{k=1}^{N_S}\sum_{g=1}^{N_G} z_{g,t}^{G}\cdot z_{k,t}^{i}\cdot\log\{\sum_{i=1}^{N}\alpha_i P(S_t^G=g|S_t^i=k)\}}_{individual\ influence\ on\ group},\end{aligned} \tag{6.5}$$

where the indicator variable $z_{j,t}=1$ if $S_t=j$, otherwise $z_{j,t}=0$. We can see that the model has complexity $O(T\cdot N\cdot N_G\cdot N_S^2)$. For example, typical values used in our experiments are $T=2000, N_S=10, N_G=5, N=4$, thus a total of 10^6 operations is required, which is still tractable. An Expectation Maximization (EM) algorithm can be applied where the E-step estimates the expectation of the indicate variable $z_{j,t}$, while the M-step maximizes the Equation 6.5.

We implemented our model using the Graphical Models Toolkit (GMTK) [17], a DBN system for speech, language, and time series data. Specifically, we used the switching parents feature of GMTK, which greatly facilitates the implementation of the two-level model to learn the influence values using the Expectation Maximization (EM) algorithm. Since EM is sensitive to local maxima, good initialization is very important. One strategy we have found useful is that we first train individual action models (Figure 6.1 (a)). Then we use the trained emission distribution from the individual action model to initialize the emission distribution of the team-player influence model. In this way, the two-level model can be trained more simply.

In the next sections, we study the performance of our model on both synthetic data (several multi-player games) and real data (a corpus of multi-party meetings). The two datasets illustrate the typical applications of our model. In the first experiment, we demonstrate that our model can learn reasonable influence values for players in various synthetic multi-player games. In the second experiment, we learn the influence of each participant in meetings using acoustic and language

Figure 6.2. A snapshot of the multi-player games: four players move along the pathes labeled in the map based on some predefined rules. Some are leading players, and some are following players. A follower tries to catch the leader by following the leader's direction. Initial positions and speeds of players are randomly generated.

features. We show that the learned influence distributions are in good accordance with the influence distributions from human judgements.

6.5 Experiments on Synthetic Data

6.5.1 The Task

We first test our model on multi-player synthetic games, in which four players (labeled A-D) move along a number of predetermined paths manually labeled in a map (Figure 6.2), based on the following rules:

- `Game I`: Player A moves randomly. Player B, C and D are meticulously following player A.
- `Game II`: Player A moves randomly. Player B is meticulously following player A. Player C moves randomly. Player D is meticulously following player C.
- `Game III`: All four players, A, B, C and D, move randomly.

A follower moves randomly until it lies on the same path as its target, and after that it tries to reach the target by following the target's direction. The initial positions and speeds of players are randomly generated. The observation of an individual player is its motion trajectory in the form of a sequence of positions, $(x_1, y_1), (x_2, y_2) \cdots (x_t, y_t)$, each of which belongs to one of 20 predetermined paths in the map. Therefore, we set $N_S = 20$. The number of team states is set to $N_G = 5$. In

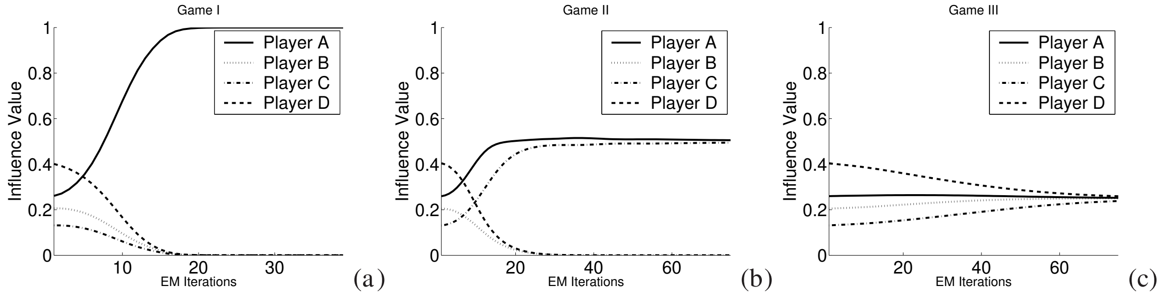


Figure 6.3. Influence values, α_i in Equation 6.4, with respect to the EM iterations in different games. (a) The final learned influence value for the leading player A is almost 1, while the influence values for the other three players are almost 0. (b) The learned influence values for both leading player A and C are close to 0.5, and the influence values for following player B and D are close to 0. (c) The learned influence values are equally around 0.25, since players A, B, C, D move randomly.

experiments, we found that the final results were not sensitive to the specific number of team states for this dataset in a range from 3 to 10 states. The length of each game sequence is $T = 2000$ frames. EM iterations were stopped once the relative difference in the global log likelihood was less than 2%.

6.5.2 Results and Discussions

Figure 6.3 shows the learned influence value for each of the four players in the different games with respect to the number of EM iterations. We can see that for `Game I`, player A is the leading player based on the defined rules. The final learned influence value for player A is almost 1, while the influence for the other three players are almost 0. For `Game II`, player A and player C are both leaders based on the defined rules. The learned influence values for player A and C are indeed close to 0.5, which indicates they have similar influence on the team. For `Game III`, the four players are moving randomly, and the learned influence values are around 0.25, which indicates that all players have similar influence on the team. The results on these toy data suggest that our model is capable of learning sensible values for $\{\alpha_i\}$, in good agreement with the concept of influence we have described before.

6.6 Experiments on Meeting Data

6.6.1 The Task

As a second application of the team-player influence model, we investigate the influence of participants in meetings. Status, dominance, and influence are important concepts in social psychology for which our model could be particularly suitable in a (dynamic) conversational setting [39]. In this section, we first describe how we manually collected influence judgements and the performance measure we used. We then report our results using audio and language features, compared with simple baseline methods. The meeting dataset has already been described in Chapter 4.

6.6.2 Manually Labeling Influence Values and Performance Measure

The manual annotation of influence of meeting participants is to some degree a subjective task, as a definite ground-truth does not exist. In our case, each meeting was labeled by three independent annotators who had no access to any information about the participants (e.g. job titles and names). This was enforced to avoid any bias based on prior knowledge of the meeting participants (e.g. a student would probably assign a large influence value to his supervisor). After watching an entire meeting, the three annotators were asked to assign a probability-based value (ranging from 0 to 1, all adding up to 1) to meeting participants, which indicated their influence in the meeting (Figure 6.5(b-d)). From the three annotations, we computed the pairwise Kappa statistics [26], a commonly used measure for inter-rate agreement, defined as

$$K = \frac{P(A) - P(E)}{1 - P(E)} \tag{6.6}$$

where $P(A)$ is the observed agreement, and $P(E)$ is the chance agreement. The obtained pairwise Kappa ranges between 0.68 and 0.72, which demonstrates a good agreement among the different annotators. We estimated the ground-truth influence values by averaging the results from the three annotators (Figure 6.5(a)).

We use the Kullback-Leibler (KL) divergence to evaluate the results. For the j^{th} meeting, given an automatically determined influence distribution $\tilde{P}_j(Q)$, and the ground truth influence distribution $P_j(Q)$, the KL divergence is given by: $D(\tilde{P}_j \| P_j) = \sum_{i=1}^{N} \tilde{P}_j(Q=i) \log_2 \frac{\tilde{P}_j(Q=i)}{P_j(Q=i)}$, where N is the

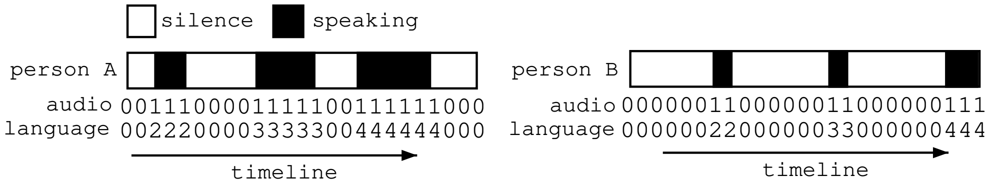


Figure 6.4. Illustration of state sequences using audio and language features respectively. Using audio, there are two states: speaking and silence. Using language, the number of states equals to the number of PLSA topics plus one silence state.

number of participants. The smaller $D(\tilde{P}_j \| P_j)$, the better the performance (if $\tilde{P}_j = P_j \Rightarrow D(\tilde{P}_j \| P_j) = 0$). Note that the KL divergence is not symmetric. We can calculate the average KL divergence for all the meetings: $D = \frac{1}{M}\sum_{j=1}^{M} D(\tilde{P}_j \| P_j)$, where M is the number of meetings.

6.6.3 Audio and Language Features

We first extract audio features useful to detect speaking turns in conversations. We compute the SRP-PHAT measure using the signals from a 8-microphone array [81], which is a continuous value indicating the speech activity from a particular participant. We use a Gaussian emission probability, and set $N_S = 2$, each state corresponding to speaking and non-speaking (silence), respectively (Figure 6.4).

Additionally, language features were extracted from manual transcripts. After removing stop words, the meeting corpus contains 2175 unique terms. We then employed *probabilistic latent semantic analysis* (PLSA) [57], which is a language model that projects documents from the high-dimensional bag-of-words space into a topic-based space of lower dimension. Each dimension in this new space represents a "topic", and each document is represented as a mixture of topics. In our case, a document corresponds to one speech utterance $(t_s, t_e, w_1 w_2 \cdots w_k)$, where t_s is the start time, t_e is the end time, and $w_1 w_2 \cdots w_k$ is a sequence of words. PLSA is thus used as a feature extractor that could potentially capture "topic turns" in meetings.

We embedded PLSA into our model by treating the states of individual players as instances of PLSA topics. Therefore, the PLSA model determines the emission probability in Equation 6.5, and is trained independently of the team-player influence model. We assign each segment to the aspect with maximal probability, similar to the strategy used in the aspect HMM [18]. We repeat the PLSA topic within the same utterance ($t_s \le t \le t_e$). The topic for the silence segments was set to 0 (Figure

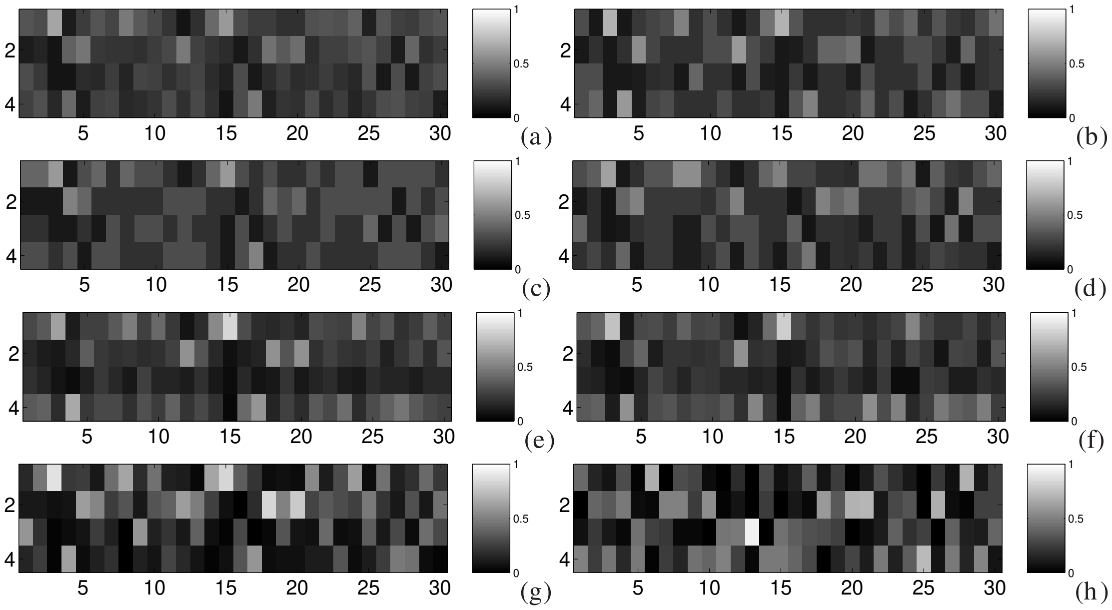


Figure 6.5. Influence values of the four participants (the y-axis direction in each figure (a) - (h)) for the 30 meetings (the x-axis direction in each figure (a) - (h)). The gray-scale bar indicates the influence values ranging from 0 (dark) to 1 (bright). Figures (a) to (h) correspond to: (a) ground-truth (average of the three human annotations: A_1, A_2, A_3). (b) A_1 : human annotation 1, (c) A_2 : human annotation 2. (d) A_3 : human annotation 3. (e) Our model + language. (f) Our model + audio. (g) Speaking-length. (h) Randomization.

6.4). We can see that using audio-only features can be seen as a special case of using language features, by using only one topic in the PLSA model (i.e., all utterances belong to the same topic). The number of PLSA topics was set to 10 ($N_S = 10$), and N_G was set to 5 ($N_G = 5$). Experimentally, we found that the final results were not sensitive to the specific number of states for this dataset in a reasonable range. But a more thorough evaluation of the effect of the choice of these parameters would be needed. In our experiments, we stop EM training once the relative difference in the global log-likelihood between two consecutive iterations was less than 2%.

6.6.4 Results and Discussions

We compare our model with a method based on the speaking length (how much time each of the participants speaks). In this case, the influence value of a meeting participant is defined to be proportional to his speaking length: $P(Q = i) = L_i / \sum_{i=1}^{N} L_i$, where L_i is the speaking length of participant i. As a second baseline model, we randomly generated 1000 combinations of influence values (under the constraint that the sum of the four values equals 1), and report the average

Table 6.1. Results of different methods on meetings ("model" denotes the team-player influence model).

Method	KL
model + Language	0.106
model + Audio	0.135
Speaking length	0.226
Randomization	0.863

Table 6.2. Results of human annotation on meetings.

Human Annotation	KL
A_i vs. A_j	0.090
A_i vs. $\overline{A_i}$	0.053
A_i vs. GT	0.037

performance.

The results on the separate test set are shown in Table 6.1 and Figure 6.5 (e-h). We can see that the performance of the three methods: model + language, model + audio, and speaking-length (Figure 6.5 (e-g)) are significantly better than that of the randomization method (Figure 6.5 (h)). Using language features with our model achieves the best performance, but it is optimistically obtained by the fact that manual transcripts were used to train the PLSA model. Our model (using either audio or language features) outperforms the speaking-length based method, which suggests that the learned influence distributions are in better accordance with the influence distributions from human judgements. As shown in Figure 6.4, using audio features can be seen as a special case of using language features. We use language features to capture "topic turns" by factorizing the two states: "speaking, silence" into more states: "topic1, topic2, ..., silence". We can see that the result using language features is better than that using audio features. In other words, compared with "speaking turns", "topic turns" improves the performance of our model to learn the influence of participants in meetings.

It is interesting to look at the KL divergence between any pair of the three human annotations (A_i vs. A_j), any one against the average of the others (A_i vs. $\overline{A_i}$), and any one against the ground-truth (A_i vs. GT). The average results are shown in Table 6.2. We can see that the result of "A_i vs. GT" is the best, which is reasonable since "GT" is the average of A_1, A_2, and A_3. Figure 6.6 shows the histogram of KL divergence between any pair of human annotations for the 30 meetings. The histogram has a distribution of $\mu = 0.09, \sigma = 0.11$. We can see that the results of our model

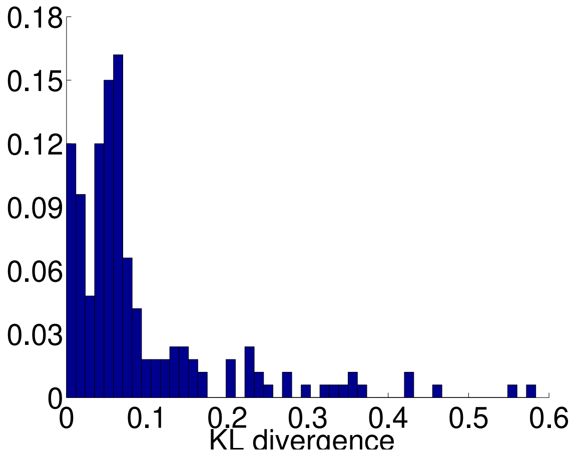


Figure 6.6. Histogram of KL divergence between any pair of the human annotations (A_i vs. A_j) for the 30 meetings. The histogram has a distribution of $\mu = 0.09, \sigma = 0.11$.

(language: 0.106, audio: 0.135) are very close to the mean ($\mu = 0.09$), which indicates that our model is comparable to human performance.

With our model, we can calculate the cumulative influence of each meeting participant over time. Figure 6.7 shows the cumulative influence values for segments with increasing length: 10 seconds, 20 seconds, ..., 5 minutes. We can see that the cumulative influence is related to the meeting agenda: The meeting starts with the monologue of person1 (monologue1). The influence of person1 is almost 1, while the influences of the other persons are nearly 0. When the four participants are involved in a discussion, the influence of person1 decreases, and the influences of the other three people increase. The influence of person4 increases quickly during monologue4. The final influence of participants becomes relatively stable in the second discussion.

6.7 Conclusions

In this chapter, we presented the team-player influence model for dominance modeling in meetings. The individual level models actions of individual players and the group-level models the group as a whole. Our model uses the team state as a way of modeling interactions among participants, while keeping a tractable representation. The proposed model explicitly learns the influence of individual players on the team. Experiments on both synthetic multi-player games and multi-party meetings showed the effectiveness of the proposed model. In the first experiment, we demonstrated that our

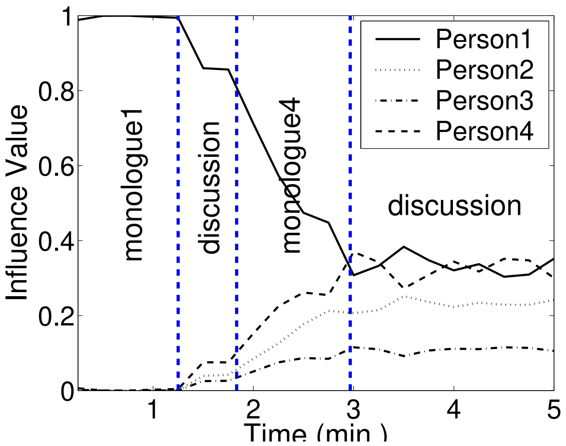


Figure 6.7. Evolution of cumulative influence over time in a 5 minute meeting. The dotted vertical lines indicate the predefined meeting agenda. The meeting starts with the monologue of person1 (monologue1). The influence of person1 is almost 1, while the influences of the other persons are nearly 0. When the four participants are involved in a discussion, the influence of person1 decreases, and the influences of the other three people increase. The influence of person4 increases quickly during monologue4. The final influence of participants becomes relatively stable in the second discussion.

model can learn reasonable influence values for players in various synthetic multi-player games. In the second experiment, we investigated the influence of participants in meetings. We showed that the learned influence distributions are in good accordance with the influence distributions estimated by human judgements.

In some event detection applications, events of interest occur over a relatively small proportion of the total time: e.g. alarm generation in surveillance systems, and extractive summarization of raw video events. We call this kind of event: *unusual event*. In the next chapter, we address the problem of unusual event modeling.

Chapter 7

Unusual Event Modeling

In this chapter, we address the problem of temporal unusual event detection. Unusual events are characterized by a number of features (rarity, unexpectedness, and relevance) that limit the application of traditional supervised model-based approaches. We propose a semi-supervised adapted Hidden Markov Model (HMM) framework, in which usual event models are first learned from a large amount of (commonly available) training data, while unusual event models are learned by Bayesian adaptation in an unsupervised manner. The proposed framework has an iterative structure, which adapts a new unusual event model at each iteration. We show that such a framework can address problems due to the scarcity of training data and the difficulty in pre-defining unusual events. Experiments on audio, visual, and audio-visual data streams illustrate its effectiveness, compared with both supervised and unsupervised baseline methods.

7.1 Introduction

In some event detection applications, events of interest occur over a relatively small proportion of the total time: e.g. alarm generation in surveillance systems, and extractive summarization of raw video events. The automatic detection of temporal events that are relevant, but whose occurrence rate is either expected to be very low or cannot be anticipated at all, constitutes a problem which has recently attracted attention in computer vision and multimodal processing under an umbrella of names (abnormal, unusual, or rare events) [121, 141, 21]. In this thesis, we employ the term *un-*

usual event, which we define as events with the following properties: (1) they seldom occur (rarity); (2) they may not have been thought of in advance (unexpectedness); and (3) they are relevant for a particular task (relevance).

It is clear from such a definition that unusual event detection entails a number of challenges. The rarity of an unusual event means that collecting sufficient training data for supervised learning will often be infeasible, necessitating methods for learning from small numbers of examples. In addition, more than one type of unusual event may occur in a given data sequence, where the event types can be expected to differ markedly from one another. This implies that training a single model to capture all unusual events will generally be infeasible, further exacerbating the problem of learning from limited data. As well as such modeling problems due to rarity, the unexpectedness of unusual events means that defining a complete event lexicon will not be possible in general, especially considering the genre- and task-dependent nature of event relevance.

Most existing works on event detection have been designed to work for specific events, with well-defined models and prior expert knowledge, and are therefore ill-posed for handling unusual events. Alternatives to these approaches, addressing some of the issues related to unusual events, have been proposed recently [121, 141, 21]. However, the problem remains unsolved.

We propose a framework for unusual event detection. Our approach is motivated by the observation that, while it is unrealistic to obtain a large training data set for unusual events, it is conversely possible to do so for usual events, allowing the creation of a well-estimated model of usual events. In order to overcome the scarcity of training material for unusual events, we propose the use of Bayesian adaptation techniques [110], which adapt a usual event model to produce a number of unusual event models in an unsupervised manner. The proposed framework can thus be considered as a semi-supervised learning technique.

In our framework, a new unusual event model is derived from the usual event model at each step of an iterative process via Bayesian adaptation. Temporal dependencies are modeled using HMMs, which have recently shown good performance for unsupervised learning [3]. We objectively evaluate our algorithm on a number of audio, visual, and audio-visual data streams, each generated by a separate source, and containing different events. With relatively simple audio-visual features, and compared to both supervised and unsupervised baseline systems, our framework produces encouraging results.

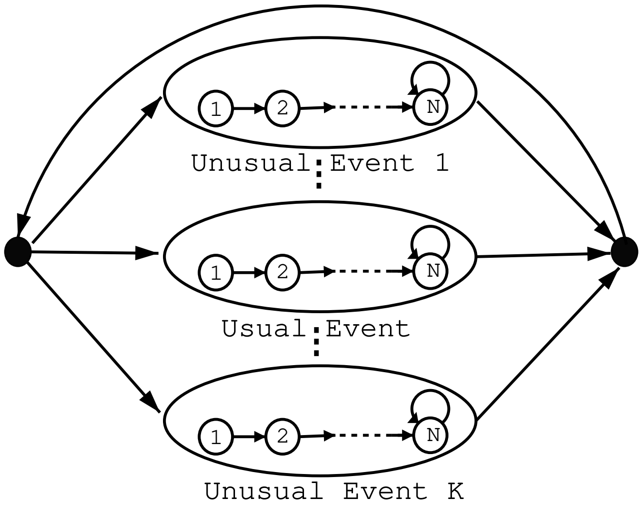


Figure 7.1. HMM topology for the proposed framework

This chapter is organized as follows. We introduce the proposed framework in Section 7.2. In Section 7.3, we present experimental results and discuss our findings. We conclude the chapter in Section 7.4.

7.2 The Iterative Adapted HMMs

In this section, we first introduce our computational framework. We then describe the implementation details.

7.2.1 Framework Overview

As shown in Figures 7.1 and 7.3, our framework is a hierarchical structure based on an ergodic K-class Hidden Markov Model (HMM) (K is the number of unusual event states plus one usual event state), where each state is a sub-HMM with minimum duration constraint. The central state represents usual events, while the others represent unusual events. All states can reach (or be reached from) other states in one step, and every state can transmit to itself.

Our method starts by having only one state representing usual events (Figure 7.2, step 0). It is normally easy to collect a large number of training samples for usual events, thus obtaining a well-estimated model for usual events. A set of parameters θ^* of the usual-event HMM model is

0. **Training the general model**
   A general usual event model is estimated with
   a large number of training samples.
1. **Outlier detection**
   Slice the test sequence into fixed length segments.
   The segment with the lowest likelihood given the
   general model is identified as outlier.
2. **Adaptation**
   A new unusual event model is adapted from the general
   usual event model using the detected outlier.
   The usual event model is adapted from the general
   usual event model using the other segments.
3. **Viterbi decoding**
   Given a new HMM topology (with one more state),
   the test sequences are decoded using Viterbi
   algorithm to determine the boundary of events.
4. **Outlier detection**
   Identify a new outlier, which has the smallest
   likelihood given the adapted usual event model.
5. Repeat step 2, 3, 4
6. **Stop**
   Stop the process after the given number of iterations.

Figure 7.2. Iterative adapted HMM

learned by maximizing the likelihood of observation sequences $\{X_1, X_2, ..., X_M\}$ as follows:

$$\theta^* = \arg\max_\theta \prod_{j=1}^{M} P(X_j|\theta). \tag{7.1}$$

The probability density function of each HMM state is assumed to be a Gaussian Mixture Model (GMM). We use the standard Expectation-Maximization (EM) algorithm [31] to estimate the GMM parameters. In the E-step, a segmentation of the training samples is obtained to maximize the likelihood of the data, given the parameters of the GMMs. This is followed by an M-step, where the parameters of the GMMs are re-estimated based on this segmentation. This creates a general usual event model.

Given the well-estimated usual event model and an unseen test sequence, we first slice the test sequence into fixed length segments overlapping with each other. This is done by moving a sliding window over the whole sequence. The choice of the sliding window size corresponds to the minimum duration constraint in the HMM framework. Given the usual event model, the likelihood of each segment is then calculated. The segment with the lowest likelihood value is identified as an outlier (Figure 7.2, step 1). The outlier is expected to represent one specific unusual event and could be used to train an unusual event model. However, one single outlier is obviously insufficient to give a good estimate of the model parameters for unusual events. In order to overcome the lack of training material, we propose the use of model adaptation techniques, such as Maximum a Posteriori (MAP) [110], where we adapt the already well-estimated usual event model to a particular unusual event model using the detected outlier, i.e, we start from the usual event model, and move towards an unusual event model in some constrained way (see Section 7.2.2 for implementation details). The original usual event model is trained using a large number of samples, which generally means that it yields Gaussians with relatively large variances. In order to make the model better suited for test sequences, the original usual event model is also adapted with the other segments (except for the detected outlier), using the same adaptation technique for the unusual event model (Figure 7.2, step 2).

Given the new unusual and usual event models, both adapted from the general usual event model, the HMM topology is changed with one more state. Hence the current HMM has 2 states, one representing the usual events and one representing the first detected unusual event. The Viterbi algorithm is then used to find the best possible state sequence which could have emitted the observation sequence, according to the maximum likelihood (ML) criterion (Figure 7.2, step 3). Transition points, which define new segments, are detected using the current HMM topology and parameters. A new outlier is now identified by sorting the likelihood of all segments given the usual event model (Figure 7.2, step 4). The detected outlier provides material for building another unusual event model, which is also adapted from usual event model. At the same time, both the unusual and usual event models are adapted using the detected unusual / usual event samples respectively. The process repeats until we obtain the desired number of unusual events. At each iteration, all usual / unusual event models are adapted from the parent node (see Figure 7.3), and a new unusual event model is derived from the usual event model via Bayesian adaptation. The number of iterations thus corresponds to the number of unusual event models, as well as the number of states in the HMM topology.

As shown in Figure 7.3, the proposed framework has a top-down hierarchical structure. Initially, there is only one node in the tree, representing the usual event model. At the first iteration, two

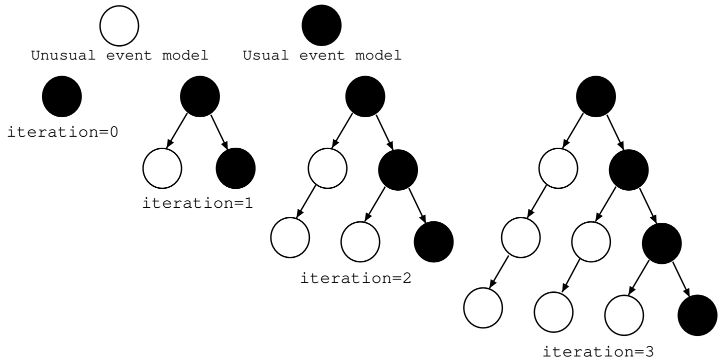


Figure 7.3. Illustration of the algorithm flow. At each iteration, two leaf nodes, one representing usual events and the other one representing unusual events, are split from the parent usual event node; A leaf node representing an unusual event is also adapted from the parent unusual event node.

new leaf nodes are split from the upper parent node: one representing usual events and the other one representing unusual events. At the second iteration, there are three leaf nodes in the tree: two for unusual events and one for usual events. The tree grows in a top-down fashion until we reach the desired number of iterations. The proposed algorithm is summarized in Figure 7.2.

Compared with previous work on unusual event detection, our framework has a number of advantages. Most existing techniques using supervised learning for event detection require manually labeling a large number of training samples. As our approach is semi-unsupervised, it does not need explicitly labeled unusual event data, facilitating initial training of the system and hence application to new conditions. Furthermore, we derive both unusual event and usual event models from a general usual event model via adaptation techniques in an online manner, thus allowing for a faster model training. In addition, the minimum duration constraint for temporal events can be easily imposed in the HMM framework by simply changing the number of cascaded states within each class.

In the next subsection, we give more details on the used adaptation techniques.

7.2.2 MAP Adaptation

Several adaptation techniques have been proposed for GMM-based HMMs, such as Gaussian clustering, Maximum Likelihood Linear Regression (MLLR) [75] and Maximum a posteriori (MAP) adaptation (also known as Bayesian adaptation) [110]. These techniques have been widely used in tasks such as speaker and face verification [110, 20]. In these cases, a general world model of speakers / faces is trained and then adapted to the particular speaker / face. In our case, we train a general usual event model and then use MAP to adapt both unusual and usual event models.

According to the MAP principle, we select parameters θ^* such that they maximize the posterior probability density, that is:

$$\theta^* = \arg\max_\theta P(\theta|X) = \arg\max_\theta P(X|\theta) \cdot P(\theta), \tag{7.2}$$

where $P(X|\theta)$ is the data likelihood and $P(\theta)$ is the prior distribution. When using MAP adaptation, different parameters can be chosen to be adapted [110]. In [110, 20], the parameters that are adapted are the Gaussian means, while the mixture weights and standard deviations are kept fixed and equal to their corresponding value in the world model. In our case we adapt all the parameters. The reason to adapt the weights is that we model events (either usual or unusual) with different components in the mixture model. When only one specific event is present, it is expected that the weights of the other components will be adapted to zero (or a relatively small value). We also adapt the variances in order to move from the general model, which may have larger covariance matrix, to a specific model, with smaller variance, focusing on one particular event in the test sequence.

Following [110], there are two steps in adaptation. First, estimates of the statistics of the training data are computed for each component of the old model. We use $\{w_i^{new}, \mu_i^{new}, \sigma_i^{2new}\}$ to represent the weight, mean and variance for component i in the new model, respectively. These parameters are estimated by ML, using the well-known equations [15],

$$w_i^{new} = \frac{1}{M}\sum_{j=1}^{M} P(i|x_j,\theta), \tag{7.3}$$

$$\mu_i^{new} = \frac{\sum_{j=1}^{M} x_j P(i|x_j,\theta)}{\sum_{j=1}^{M} P(i|x_j,\theta)}, \tag{7.4}$$

$$\sigma_i^{2new} = \frac{\sum_{j=1}^{M} P(i|x_j,\theta)(x_j-\mu_i^{new})(x_j-\mu_i^{new})^T}{\sum_{j=1}^{M} P(i|x_j,\theta)}, \tag{7.5}$$

where M is the number of data examples.

In the second step, the parameters of a mixture i are adapted using the following set of update equations [110].

$$\hat{w}_i = \alpha \cdot w_i^{old} + (1 - \alpha) \cdot w_i^{new}, \tag{7.6}$$

$$\hat{\mu}_i = \alpha \cdot \mu_i^{old} + (1 - \alpha) \cdot \mu_i^{new}, \tag{7.7}$$

$$\hat{\sigma}_i = \alpha \cdot (\sigma_i^{old} + (\hat{\mu}_i - \mu_i^{old})(\hat{\mu}_i - \mu_i^{old})^T) + (1 - \alpha) \cdot (\sigma_i^{new} + (\hat{\mu}_i - \mu_i^{new})(\hat{\mu}_i - \mu_i^{new})^T), \tag{7.8}$$

where $\{\hat{w}_i, \hat{\mu}_i, \hat{\sigma}_i^2\}$ are weight, mean and variance of the adapted model in component i, $\{w_i^{old}, \mu_i^{old}, \sigma_i^{2old}\}$ are the corresponding parameters in the old component i respectively, and α is a weighting factor to control the balance between old model and new estimates. The smaller the value of α, the more contribution the new data makes to the adapted model.

7.3 Experiments and Results

In this section, we first introduce the performance measures and baseline systems we used to evaluate our results. Then we illustrate the effectiveness of the proposed framework using audio, visual and audio-visual events.

7.3.1 Performance Measures

The problem of unusual event detection is a two-class classification problem (unusual events *vs.* usual events), with two types of errors: a *false alarm* (FA), when the method accepts an usual event sample (frame), and a *false rejection* (FR), when the method rejects an unusual event sample. The performance of the unusual event detection method can be measured in terms of two error rates: the *false alarm rate* (FAR), and the *false rejection rate* (FRR), defined as follows:

$$\text{FAR} = \frac{\text{number of FAs}}{\text{number of usual event samples}} \times 100\%, \tag{7.9}$$

$$\text{FRR} = \frac{\text{number of FRs}}{\text{number of unusual event samples}} \times 100\%. \tag{7.10}$$

The performance for an ideal event detection algorithm should have low values of both FAR and FRR. We also use the *half-total error rate* (HTER), which combines FAR and FRR into a single measure: HTER $= \frac{\text{FAR}+\text{FRR}}{2}$.

7.3.2 Baseline Systems

To evaluate the results, we compare the proposed semi-supervised framework with the following baseline systems.

- **Supervised HMM:** Two standard HMM models, one for usual events and one for unusual events, are trained using manually labeled training data according to Equation 7.1. For testing, the event boundary is obtained by applying Viterbi decoding on the sequences.

  For supervised HMM, we test two cases. In the first case, we train usual and unusual event models using a large (sufficient) number of samples, referred to as *supervised-1*. In the second case, referred to as *supervised-2*, around 10% of the unusual event training samples from the first case are used to train the unusual event HMM. The purpose of *supervised-2* is to investigate the case where there is only a small number of unusual event training samples.

- **Unsupervised HMM:** The second baseline system is an agglomerative HMM-based clustering algorithm, recently proposed for speaker clustering [3], and that has shown good performance. The unsupervised HMM clustering algorithm starts by over-clustering, i.e. clustering the data into a large number of clusters. Then it searches for the best candidate pair of clusters for merging based on the criterion described in [3]. The merging process is iterated until there are only two clusters left, one assumed to correspond to usual events, and another one for unusual events. We assume that the cluster with the largest number of samples represents usual events, and the other cluster represents unusual events. This model is referred to as *unsupervised*.

For both the proposed approach and the baseline methods, all parameters are selected to minimize *half-total error rate* (HTER) criterion on a validation data set.

7.3.3 Results on Audio Events

For the first experiment, we used a data set of audio events obtained through a sound search engine: `http://www.findsounds.com/types.html`. The purpose of this experiment is to have a

Table 7.1. Audio events data. Number of frames for various methods (*NA:* Not Applicable).

method	train set		test set	
	usual	unusual	usual	unusual
our approach	90000	*NA*	72750	2250
supervised-1	90000	20000		
supervised-2	90000	2000		
unsupervised	*NA*	*NA*		

controlled setup for evaluation of our algorithm. We first selected 60 minutes of audio data containing only 'speaking' events. We then manually mixed it with other interesting audio events, namely 'applause', 'cheer', and 'laugh' events. The length of each concatenated segment is random. 'Speaking' is labeled as usual event, while all the other events are considered unusual. The minimum duration for audio events is two seconds.

We extracted Mel-Frequency Cepstral Coefficients (MFCCs) features for this task. MFCC are short-term spectral-based features and have been widely used in speech recognition [106] and audio event classification. We extracted 12 MFCC coefficients from the original audio signal using a sliding window of 40ms at fixed intervals of 20ms. The number of training and testing frames for the different methods is shown in Table 7.1. Note that there is no need for unusual event training data for our approach. For the unsupervised HMM, there is no need for training data. The percentage of frames for unusual events in the test sequence is around 3%.

Figure 7.4(a) shows the performance of the proposed approach with respect to the number of iterations. We observe that FRR always decreases while FAR continually increases with the increase of the number of iterations. This is because our approach derives a new unusual event modal from the usual event model via Bayesian adaptation at each iteration. With the increase of unusual event models, more unusual events can be detected, while more usual events are falsely accepted as unusual events.

Figure 7.4(b) shows the performance comparison between the proposed approach and baseline systems in terms of HTER. We can see that the supervised HMM with sufficient amount of training data gives the best performance. The proposed approach improves the performance, compared to the *supervised-2* and *unsupervised* baselines when the number of iterations is adequately closer. The results show that the benefit of using the proposed approach is not performance improvement when sufficient training data is available, but rather its effectiveness when there are not enough

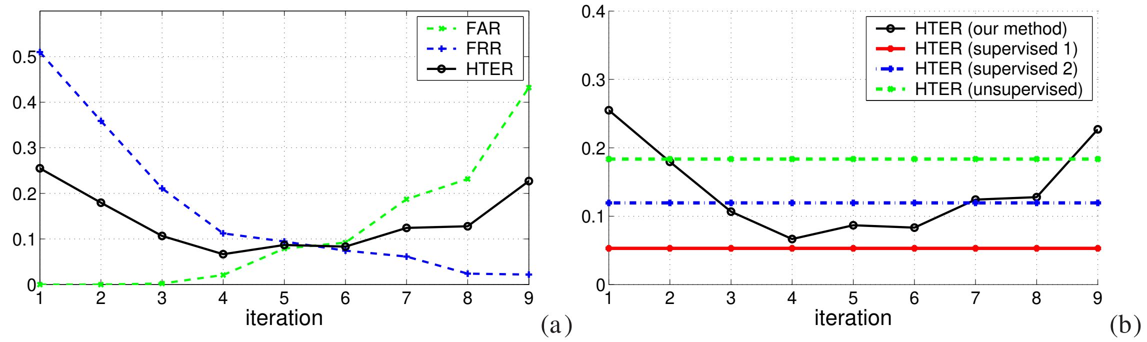


Figure 7.4. Results for audio unusual event detection. The X-axis represents the number of iterations in our approach.

training samples for unusual events. The best result of our approach is obtained at 4 iterations (HTER $= 6.65\%$), slightly worse than *supervise-1* (HTER $= 5.29\%$), showing the effectiveness of our approach given that it does not need any unusual event training data.

7.3.4 Results on Visual Events

The visual data we investigate is a 30-minute long poker game video, containing 26 different events and originally manually labeled and used in [141]. Seven cheating related events, including 'hiding a card', 'exchanging cards', 'passing cards under table', etc., are categorized as unusual events (see Figure 7.6). Other events such as 'playing cards', 'drinking water', and 'scratching', are considered as usual events. The minimum duration for these visual events is 15 frames.

The number of training and testing frames for different methods is shown in Table 7.2. While we chose this visual task to show application on an existing data set, we note that the percentage of frames of unusual events in the test sequence is about 17%, which does not correspond very well to the assumption of *rarity* made by our model. The unusual event testing data for the *supervised-1* method is much smaller, compared to other methods. This is because we use a larger number of unusual event frames (1320) for training, and we are left with a small number of unusual event frames (195) for testing. To deal with this problem, we repeat experiments for *supervised-1* ten times by randomly splitting total unusual events into two parts: one with 1320 frames for training, and the other one with 195 frames for testing. We report the mean results of the ten runs. Note also that the amount of training data for the unusual model (1320 frames) is smaller than the previous experiments.

Table 7.2. Video events data. Number of frames for various methods (NA: Not Applicable).

method	train set		test set	
	usual	unusual	usual	unusual
our approach	9000	NA	7387	1515
supervised-1	9000	1320		195
supervised-2	9000	300		1215
unsupervised	NA	NA		1515

We extract motion and color features from moving blocks of each frame in the video in a similar way as in [141]. We start with a static background image. We detect the moving objects using background subtraction. We then superimpose a 6×6 grid on the detected motion mask. We first compute a motion histogram. In each tile of the grid, we calculate the total number of motion pixels, and these features are concatenated to form a $6 \times 6 = 36$ dimension of feature vector to describe the motion in the current frame. In a similar way, we can compute the color histogram for the moving objects in chromatic color space (defined by $r = \frac{R}{R+G+B}, g = \frac{G}{R+G+B}$). We concatenate the motion histogram and the color histogram into a $108 = 36 + 2 \times 36$ dimensional feature vector. To reduce the feature space dimension and for feature decorrelation, we apply a Principal Component Analysis (PCA) to transform the 108-dimensional features into 36-dimensional features.

The results are shown in Figure 7.5. Overall, this is a more difficult task. We observe the similar trend of FAR and FRR as in audio event detection, with respect to the number of iterations in our approach. The best result of our approach is obtained with 4 iterations, although the values of HTER are relatively stable between 4 iterations and 7 iterations. We come to similar conclusions as for the audio event detection, that is, the supervised approach with sufficient training samples provides the best performance, while the proposed framework is better than the other baseline systems. Note that the supervised approach with small number of training samples performs worse than the unsupervised approach.

7.3.5 Results on Audio-Visual Events

We also apply our framework to audio-visual unusual event detection using the ICCV'03 recorded presentation videos, publicly available: `http://www.robots.ox.ac.uk/~awf/iccv03videos`. Each presentation video is about 20 minutes in length with 25 frames per second. We define a set of multimodal unusual events, including 'speaker showing demo, audience applause', 'speaker playing

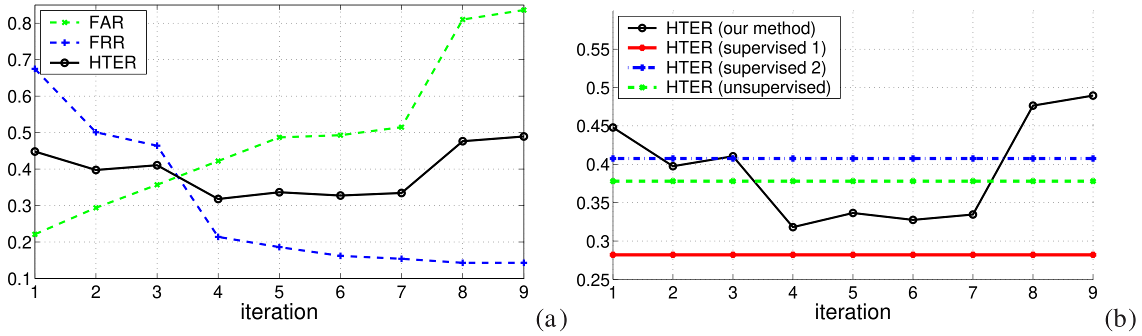


Figure 7.5. Results of visual unusual events detection.

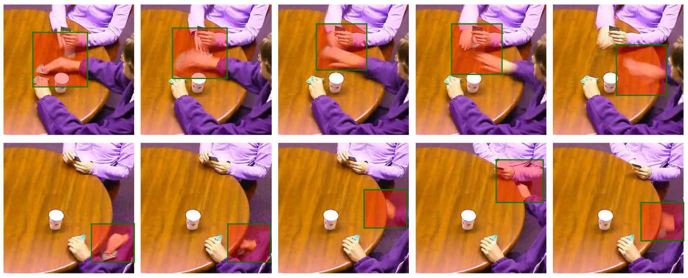
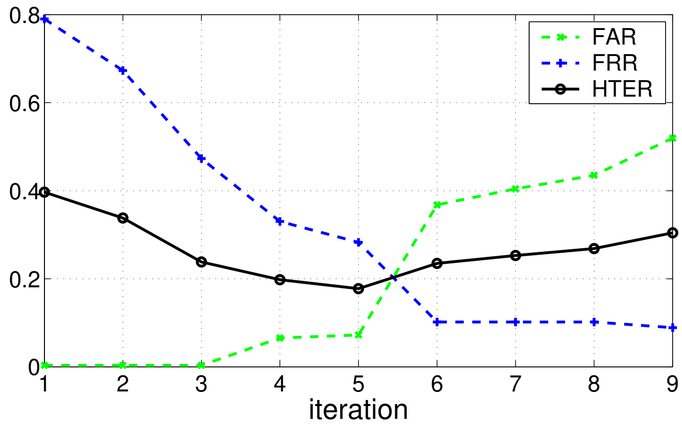


Figure 7.6. Top: Visual event of 'exchanging cards'; Bottom: Visual event of 'passing cards under table'

video, audience laugh', and 'speaker interrupted by audience's questions'. Note that since some unusual events in the presentation setting cannot be defined before watching the entire database, the unusual events list we define here should be regarded as a small subset.

A set of audio-visual features were extracted. For audio features, we used the same features as in Section 7.3.3. For visual features, we extracted a motion histogram from each frame of the video, computed in a similar way to Section 7.3.4. Audio and visual features were then concatenated.

Since the occurrence of unusual events is rare, manually labeling a large amount of samples is impractical, highlighting the need for semi-supervised or unsupervised approaches. Due to the lack of sufficient annotated training data for the supervised baselines, we only report results of our approach. Two presentation videos are used for training to build the general usual event model. We then apply our framework to a third meeting for unusual event detection. We labeled the events by hand to obtain a ground truth in the three videos. The results are shown in Figure 7.7. We observe that, with the increase of iterations, FRR decreases while FAR increases, which means that more unusual events are detected, but at the cost of falsely accepting more usual events as unusual events. The best result of our approach is obtained when the number of iterations is 5.



(a)

Figure 7.7. Results of our approach in terms of FAR, FRR and HTER.

Table 7.3. Overall the best results

Events	Method	FAR %	FRR %	HTER %
audio	our method	2.09	11.2	6.65
	supervised 1	3.97	6.62	5.29
	supervised-2	11.8	12.6	12.2
	unsupervised	12.5	24.2	18.3
visual	our method	42.2	21.4	31.8
	supervised-1	26.8	29.6	28.2
	supervised-2	41.3	40.2	40.7
	unsupervised	40.1	35.5	37.8
audio-visual	our approach	7.20	28.2	17.7

7.3.6 Overall Discussion

Table 7.3 summarizes overall results of audio, visual and audio-visual unusual event detection. For the proposed approach, the results correspond to the iteration with the minimum HTER. For both audio and visual unusual event detection, we can see that supervised HMM well-trained with sufficient data achieves the best performance while the proposed approach performs better than the other baseline systems.

As a well-known rule-of-thumb, the number of training samples needed for a well-trained model is directly related to the model complexity (the number of model parameters). The penalty for training with insufficient data is over-fitting, *i.e.* poor generalization capability. Both our approach and the baseline methods are based on HMMs for usual and unusual events modeling and hence have similar model complexity.

For the proposed approach, we currently do not determine the optimal number of iterations. As shown in Figures 7.4, 7.5 and 7.7, finding the optimal number of iterations is a trade-off between FAR and FRR. Some applications require more unusual events detected thus need more iterations. Otherwise, we might stop iterations at the early stages if fewer false alarms are expected. Automatic model selection is a difficult problem that we are studying, in particular with the Bayesian Information Criterion (BIC) [116]. In our approach, there is one additional state in the HMM topology at each iteration, which results in an increase of both the number of model parameters and the likelihood of a test sequence. BIC could be used to handle the trade-off between model complexity and data likelihood.

We also note that feature selection is a critical issue in unusual event detection, particularly when using a semi- or unsupervised approach. The nature of the events found by the system will necessarily relate to the nature of discrimination provided by the features. In the above experiments, while the audio features seem to allow such discrimination, ongoing research should include investigation of different visual features.

Finally, regarding the three properties we used to define an unusual event (rarity, unexpectedness, and relevance), our method aims at accounting for the first two (one could argue that unexpectedness is a feature of some rare events). Relevance is a task-dependent property, whose incorporation in our work would require human intervention.

7.4 Conclusions

In this chapter, we presented a semi-supervised adapted HMM framework for unusual event detection. The proposed framework is well suited for cases in which collecting sufficient unusual event training data is impractical and unusual events cannot be defined in advance. With relatively simple audio-visual features, and compared to both supervised and unsupervised baseline systems, our framework produces encouraging results.

Chapter 8

Conclusion

8.1 Summary of Achievements

In this thesis, we have developed a number of computational frameworks for human interaction modeling from sensor information. We used the more general probabilistic graphical model framework, which provides for families of probability distributions to be modeled in a concise and intuitive manner. The major contributions of this thesis can be broken down into three computational models (*i.e.* multi-layer HMM, team-player influence model, and semi-supervised adapted HMM) and three related tasks (*i.e.* group action modeling, influence modeling, and unusual event modeling). The following sections presents brief summary of specific contributions.

Part I (Chapter 4 – Chapter 5) provided a multi-layer HMM framework which decomposes the problem of human interaction modeling into *individual action* and *group action* stages, thus simplifying the complexity of the task. By defining a proper set of individual actions, group actions can be modeled as a two-layer process, one that models basic individual activities from low-level audio-visual features, and another one that models the interactions. We proposed a two-layer Hidden Markov Model (HMM) framework that implements such concept in a principled manner, and that has advantages over previous works. First, by decomposing the problem hierarchically, learning is performed on low-dimensional observation spaces, which results in simpler models. Second, our framework is easier to interpret, as both individual and group actions have a clear meaning, and thus easier to improve. Third, different HMM models can be used in each layer, to better reflect the nature of each subproblem. This framework is general and extensible, and experiments and comparison with a single-layer HMM baseline system showed its validity.

In Part II (Chapter 6), we proposed a model that learns the influence of interacting Markov chains within a team. The proposed model is a dynamic Bayesian network (DBN) with a two-level structure: individual-level and group-level. The individual level models actions of each player, and the group-level models actions of the team as a whole. Unlike existing multi-stream models, the influence of each player on the team, which can also be interpreted as reliability in the case of data fusion, is jointly learned with the rest of the model parameters in a principled manner using the Expectation-Maximization (EM) algorithm. Experiments on synthetic multi-player game and multi-party meetings showed the effectiveness of the proposed model. In the first experiment, we demonstrated that our model could learn reasonable influence values for players in simple synthetic multi-player games. In the second experiment, we learnt the influence of each participant in meetings using acoustic and language features. We show that the learned influence distributions are in good accordance with the influence distributions from human judgements.

In Part III (Chapter 7), we addressed the problem of temporal unusual event detection. Unusual events are characterized by a number of features that limit the application of traditional supervised model-based approaches. We proposed a semi-supervised adapted Hidden Markov Model (HMM) framework, in which usual event models are first learned from a large amount of (commonly available) training data, while unusual event models are learned by Bayesian adaptation in an unsupervised manner. The proposed framework has an iterative structure, which adapts a new unusual event model at each iteration. We showed that such a framework could address problems due to the scarcity of training data and the difficulty in pre-defining unusual events. Experiments on audio, visual, and audio-visual data streams illustrated its effectiveness, as compared to both supervised and unsupervised baseline methods.

8.2 Directions to Explore

Without being exhaustive, we identify the following areas of potential future work.

- **Multi-stream Modeling**

  Multi-stream sequence processing is a very challenging problem in machine learning. Some open issues are discussed in [12]. For instance, most solutions that model the joint probability of the streams need in general exponential resources with respect to the number of streams, the number of states of each underlying Markov chain, or the size of each stream, thus quickly become intractable. There are two directions for this problem. First, we could decompose the complex problem of multi-stream modeling into stages. Second, we could derive approximate inference techniques (such as variational Bayesian methods [6]) to make complex models tractable.

- **Group Action Modeling**

  - **Defining a richer set of group actions:** The group action lexicon we currently defined is small and thus limited. A direction of future work could be the definition of a rich set of group actions. For that purpose, incorporating semantic language features would be essential for such task.
  - **Incorporation of a language model for actions:** Being successfully used in continuous speech recognition, a language model is the way in which words are naturally used together, according to the grammar and spelling rules of a language. A line of future research could be the investigation of a language model for group actions in the meeting scenarios.
  - **Group action recognition in other applications:** Meetings are just a particular case of where group actions emerge. Group activity recognition and discovery play an important role in other applications, such as surveillance, monitoring, and human-computer interaction. Applying the models proposed in this thesis to other scenarios could be a direction for future research.

- **Dominance Modeling**

- **Modeling:** The team-player influence model could be studied and extended in a number of ways. On the modeling side, we still need to gain deeper understanding of the "meaning" of the hidden team variable, which acts as a bottleneck node that reduces the complexity of the model from a combinatorial number of pairwise links to a linear number, but which, as discussed in Chapter 6, has no clear semantic interpretation when the model is trained in an unsupervised fashion. A second important issue is the fact that influence is a parameter that might have fluctuations over time. Investigating this issue could be especially attractive for analyzing group activity from a long-term perspective.
- **Application:** On the application side, a deeper study of the effect of automatically extracted multimodal features, *e.g.* speaking length with gaze, on the estimation of influence in meetings is necessary.
- **Evaluation:** As discussed in Chapter 6, detecting influence of meeting participants is to some degree a subjective task, as a unique ground-truth and a perfect evaluation measure does not exist. We used Kullback-Leibler (KL) divergence to evaluate the results reported in this thesis. One could investigate the use of alternative evaluation measures.

• **Unusual Event Detection**

- **Model Selection:** For the proposed semi-supervised adapted HMM framework, we currently do not determine the optimal number of iterations. Finding the optimal number of iterations is essentially a *model selection* problem. Some applications might require more unusual events being detected, and thus need more iterations in the proposed algorithm. Otherwise, we might stop iterations at the early stages if fewer false alarms are expected. Automatic model selection is a difficult problem in machine learning. There are various methods that can be used for model section, such as *simple validation*, *cross-validation*, and Bayesian Information Criterion (BIC) [116] methods. In particular, BIC could be a promising method because in our semi-supervised adapted HMMs, there is one additional state in the topology at each iteration, which results in an increase of both the number of model parameters and the likelihood of a test sequence. BIC could be used to handle the trade-off between model complexity and data likelihood. The investigation of the optimal model structure remains as a future research issue.

– **The unbalanced data:** Many applications face the problem of unbalanced data. That is, the number of samples for some classes is far less (or more) than samples of the other classes. The problem of unusual event detection belongs to this category. There is relatively little work in machine learning that explicitly addresses the problem. In the future, a research issue might be the investigation of novel machine learning algorithms that can address this data-unbalanced problem in a principled way.

Bibliography

[1] M. Al–Hames, G. Rigoll. A Multi-Modal Mixed-State Dynamic Bayesian Network for Robust Meeting Event Recognition from Disturbed Data. In *Proc. 6th International Conference on Multimedia and Expo (ICME)*, Amsterdam, July, 2005.

[2] J. Ajmera, H. Bourlard, I. Lapidot, and I. McCowan. Unknown-multiple speaker clustering using HMM. In *ICSLP*, Colorado, 2002.

[3] J. Ajmera, and C. Wooters. A robust speaker clustering algorithm. In *IEEE Automatic Speech Recognition Understanding Workshop*, 2003.

[4] T. Allen. Architecture and communication among product development engineers. In *Sloan School of Magement, MIT*, 1997.

[5] C. Asavathiratham. The influence model: A tractable representation for the dynamics of networked Markov chains. *Dept. of EECS, MIT, Cambridge*, 2000.

[6] H. Attias. Inferring parameters and structure of latent variable models by variational bayes. In *Proceedings of the 15th Annual Conference on Uncertainty in Artificial Intelligence (UAI-99)*, San Francisco, CA, 1999. Morgan Kaufmann.

[7] R. F. Bales. *Interaction Process Analysis: A method for the study of small groups.* Addison-Wesley, 1951.

[8] R.F. Bales, F.L. Strodbeck, T.M. Mills, and M.E. Roseborough. Channels of communication in small groups. *American Sociological Review*, 16:461–468, 1951.

[9] S. Basu, T. Choudhury, B. Clarkson, and A Pentland. Learning human interactions with the influence model. Technical Report 539, MIT Media Laboratory, June 2001.

[10] S. Basu, T. Choudhury, B. Clarkson, and A. Pentland. Towards measuring human interactions in conversational settings. In *Proc. IEEE CVPR Workshop on Cues in Communication*, Kawai, Dec. 2001.

[11] S. Bengio. An asynchronous hidden Markov model for audio-visual speech recognition. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems, NIPS 15*. MIT Press, 2003.

[12] S. Bengio and H. Bourlard. Multi channel sequence processing. In *In J. Winkler, M. Niranjan, and N. Lawrence, editors, Deterministic and Statistical Methods in Machine Learning: First International Workshop, Lecture Notes in Artificial Intelligence*, number volume LNAI 3635, Springer-Verlag.

[13] Y. Bengio and P. Frasconi. An input/output HMM architecture. *In Advancesin Neural Information ProcessingSystems*, page 427–434, 1995.

[14] J. Berger, S.J. Rosenholtz, and M. Zelditch Jr. Status organizing processes. *Annual Review of Sociology*, 6:479–508, 1980.

[15] J. Bilmes. A gentle tutorial of the EM algirthm and its application to parameter estimation for Gaussian mixture and hidden Markov models. In *ICSI-TR-97-021, U.C. Berkeley*, 1997.

[16] J. Bilmes. Dynamic Bayesian multinets. *Proc. of the 16th conf. on Uncertainty in Artificial Intelligence*, 2000.

[17] J. Bilmes and G. Zweig. The graphical models toolkit: An open source software system for speech and time series processing. *Proc. ICASSP*, vol. 4:3916–3919, 2002.

[18] D. Blei and P. Moreno. Topic segmentation with an aspect hidden Markov model. *Proc. of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 343–348, 2001.

[19] H. Buxton and S. Gong. Advanced visual surveillance using Bayesian networks. *In International Conference on Computer Vision*, 1995.

[20] F. Cardinaux, C. Sanderson, and S. Bengio. Adapted generative models for face verification. In *IEEE, Face and Gesture*, 2004.

[21] M.T. Chan, A. Hoogs, J. Schmiederer, and M. Perterson. Detecting rare events in video using semantic primitives with HMM. In *In Proc. ICPR*.

[22] P Chang, M Han, and Y Gong. Highlight detection and classification of baseball game video with hidden Markov models. In *IEEE ICIP*, New York, Sept. 2002.

[23] S. Chiappa and S. Bengio. HMM and IOHMM modeling of EEG rhythms for asynchronous BCI systems. In *European Symposium on Artificial Neural Networks ESANN*, 2004.

[24] T. Choudhury and S. Basu. Modeling conversational dynamics as a mixed memory Markov process. *Proc. of Intl. Conference on Neural Information and Processing Systems (NIPS)*, 2004.

[25] T. Choudhury and A. Pentland. Sensing and modeling human networks using the sociometer. In *in Proceedings of the International Conference on Wearable Computing*, 2003.

[26] J.A. Cohen. A coefficient of agreement for nominal scales. In *Educ Psych Meas*, number 20.

[27] J. L. Crowley. Social perception. In *ACM Queue vol. 4, no. 6*, San Francisco, CA, July 2006.

[28] R. Cutler, Y. Rui, A. Gupta, JJ Cadiz, I. Tashev, L. He, A. Colburn, Z. Zhang, Z. Liu, and S. Silverberg. Distributed meetings: A meeting capture and broadcasting system. In *Proc. ACM Int. Conf. on Multimedia*, 2002.

[29] A. Dielmann and S. Renals. Dynamic Bayesian networks for meeting structuring. In *Proc. IEEE ICASSP*, 2004.

[30] I. Mian D. Haussler, A. Krogh and K. Sjolander. Protein modeling using hidden Markov models: analysis of globins. In *Proceedings of the Hawaii International Conference on System Sciences*, volume 1, 1993.

[31] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(B):1–38, 1977.

[32] J. DiBiase, H. Silverman, and M. Brandstein. Robust localization in reverberant rooms. In M. Brandstein and D. Ward, editors, *Microphone Arrays*, chapter 8, pages 157–180. Springer, 2001.

[33] A. Dielmann and S. Renals. Dynamic Bayesian networks for meeting structuring. *In Proc. IEEE ICASSP*, 2004.

[34] M. Dirst and A. Weigend. Baroque forecasting: on completing JS Bachs last fugue. in A. Weigend and N. gershenfeld (Eds.), *time series prediction: forecasting the future and understanding the past.* In *Addison-Wesley, Reading, MA*, 1993.

[35] S. Duncan. Some signals and rules for taking speaker turns in conversation. In *Journal of Personality and Social Psychology*, number 23:2.

[36] S. Dupont and J. Luettin. Audio-visual speech modeling for continuous speech recognition. *IEEE Transactions on Multimedia*, 2(3):141–151, September 2000.

[37] N. Eagle and A. Pentland. Reality mining: Sensing complex social systems. In *Journal of Personal and Ubiquitous Computing*, 2005.

[38] M.I. Jordan (Ed.). Learning in graphical models. In *Cambridge MA, MIT Press*, 1999.

[39] S. L. Ellyson and J. F. Dovidio editors. Power, dominance, and nonverbal behavior. In *Springer-Verlag*, 1985.

[40] N. Fay, S. Garrod, and J. Carletta. Group discussion as interactive dialogue or serial monologue: The influence of group size. *Psychological Science*, 11(6):487–492, 2000.

[41] S. Fine, Y. Singer, and N. Tishby. The hierarchical hidden Markov model: Analysis and applications. In *Machine Learning*, pages 32–41, 1998.

[42] J. Forbes, T. Huang, K. Kanazawa, and S. Russell. The batmobile: towards a Bayesian automated taxi. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, 1995.

[43] B. Frey and N. Jojic. Transformed component analysis: Joint estimation of spatial transformations and image components. In *ICCV*, pages 1190–1196, Sep. 1999.

[44] A. Galata, N. Johnson, and D. Hogg. Learning behavior models of human activities. *In British Machine Vision Conference*, 1999.

[45] J. L. Gauvain and C.-H. Lee. Maximum a Posteriori estimation for multivariate Gaussian mixture observation of Markov chains. In *IEEE Transactions on Speech Audio Processing*, volume 2, pages 291–298, April 1994.

[46] D.M. Gavrila. The visual analysis of human movement: A survey. In *Computer Vision and Image Understanding: CVIU*, number 73, pages 82–98, 1999.

[47] Z. Ghahramani and M. I. Jordan. Factorial hidden Markov models. *In Machine Learning*, 1997.

[48] Zoubin Ghahramani and Geoffrey E. Hinton. Variational learning for switching state-space models. In *Neural Computation*, pages 963 – 996, 1998.

[49] D. Gibbon, R. Moore, and R. Winksi. *Handbook of Standards and Resources for Spoken Language Systems*. Mouton de Gruyter, 1997.

[50] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter. Markov chain monte carlo in practice. In *Chapman & Hall, London*, 1996.

[51] G. Goetsch and D. McFarland. Models of the distribution of acts in small discussion groups. *Social Psychology Quarterly*, 43(2):173–183, 1980.

[52] S. Gong and T. Xiang. Recognition of group activities using a dynamic probabilistic network. In *Proc. IEEE ICCV*, Oct. 2003.

[53] W. E. L. Grimson, C. Stauffer, R. Romano, and L. Lee. Using adaptive tracking to classify and monitor activities in a site. In *Proc. of CVPR*, pages 22–29, 1998.

[54] X. He, R. Zemel, and M. Carreira-Perpinan. Multiscale conditional random fields for image labelling. In *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[55] H. Hermansky, D. Ellis, and S. Sharma. TANDEM connectionist feature extraction for conventional HMM systems. In *Proc. of ICASSP*, Istanbul, June 2000.

[56] D. Hillard, M. Ostendorf, and E. Shriberg. Detection of agreement vs. disagreement in meetings: Training with unlabeled data. In *Proc. HLT-NAACL Conference*, Edmonton, May 2003.

[57] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. In *Machine Learning, 42:177–196*, 2001.

[58] S. Hongeng, F. Bremond, and R. Nevatia. Bayesian framework for video surveillance application. *In International Conference on Pattern Recognition*, 2000.

[59] S. Hongeng and R. Nevatia. Multi-agent event recognition. In *Proc. IEEE ICCV*, Vancouver, July 2001.

[60] A. Howard and T. Jebara. Dynamical systems trees. In *Uncertainty in Artificial Intelligence (UAI)*, July, 2004.

[61] V. Hozjan and Z. Kacic. Improved emotion recognition with large set of statistical features. In *Proc. Eurospeech*, Geneva.

[62] E. T. Jaynes. Information theory and statistical mechanics. In *The Physical Review*.

[63] Finn V. Jensen and Frank Jensen. Optimal junction tree. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, pages 360 – 366, 1994.

[64] A. Just, O. Bernier, and S. Marcel. HMM and IOHMM for the recognition of mono- and bi-manual 3D hand gestures. In *IDIAP-RR 04–39*, 2004.

[65] Sham Kakade, Yee Whye Teh, and Sam Roweis. An alternative objective function for Markovian fields. In *In International Conference on Machine Learning 19 (ICML 02).*

[66] L. Kennedy and D. Ellis. Pitch-based emphasis detection for characterization of meeting recordings. In *Proc. ASRU*, Virgin Islands, Dec. 2003.

[67] Hamed Ketabdar, Jithendra Vepa, Samy Bengio, and Herve Bourlard. Developing and enhancing posterior based speech recognition systems. In *Proceedings of Interspeech*, 2005.

[68] K. Kirchhoff, S. Parandekar, and J. Bilmes. Mixed-memory Markov models for automatic language identification. *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, 2000.

[69] R. Krauss, C. Garlock, P. Bricker, and L. McMahon. The role of audible and visible back-channel responses in interpersonal communication. *Journal of Personality and Social Psychology*, 35(7):523–529, 1977.

[70] F. Kubala. Rough'n'ready: a meeting recorder and browser. *ACM Computing Surveys*, (31), 1999.

[71] S. Kumar and M. Hebert. Discriminative fields for modeling spatial dependencies in natural images. In *In Neural Information Processing Systems 16. MIT Press, Cambridge, MA*, 2003.

[72] J. Kwon and K. Murphy. Modeling freeway traffic with coupled HMMs. *Technical report, University of California at Berkeley*, May 2000.

[73] O. Kwon, K. Chan, J. Hao, and T. Lee. Emotion recognition by speech signals. In *In* Proc. Eurospeech*, Geneva*.

[74] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning (ICML-2001)*, 2001.

[75] C. Leggetter and C. WoodlandP. Flexible speaker adaptation using maximum likelihood linear regression. 1995.

[76] Y. Li and H.Y. Shum. Learning dynamic audio-visual mapping with input-output hidden Markov models. In *ACCV*, 2002.

[77] Y .Liu, J. Carbonell, P. Weigele, and V. Gopalakrishnan. segmentation conditional random fields (SCRFs): A new approach for protein fold recognition. In *In ACM International conference on Research in Computational Molecular Biology (RECOMB05)*, 2005.

[78] J. D. Markel. The SIFT algorithm for fundamental frequency estimation. *IEEE Transactions on Audio and Electroacoustics*, 20:367–377, 1972.

[79] D. W. Massaro and D. G. Stork. Speech recognition and sensory integration. In *American Scientist*, number 86:3.

[80] I. McCowan, S. Bengio, D. Gatica-Perez, G. Lathoud, F. Monay, D. Moore, P. Wellner, and H. Bourlard. Modeling human interactions in meetings. In *Proc. IEEE ICASSP*, Hong Kong, April 2003.

[81] I. McCowan, D. Gatica-Perez, S. Bengio, G. Lathoud, M. Barnard, and D. Zhang. Automatic analysis of multimodal group actions in meetings. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, volume 27(3), pages 305–317, 2005.

[82] J. McGrath and D. Kravitz. Group research. *Annual Review of Psychology*, 33:195–230, 1982.

[83] J. E. McGrath. *Groups: Interaction and Performance.* Prentice-Hall, 1984.

[84] G. Medioni, I. Cohen, F. Bremond, S. Hongeng, and R. Nevatia. Event detection and analysis from video streams. In *In IEEE Transactions on Pattern Analysis and Machine Intelligence*, number Vol. 23(8).

[85] N. Mirghafori and N. Morgan. Transmissions and transitions: A study of two common assumptions in multi-band ASR. In *Proc. ICASSP*, Seattle, 1998.

[86] D. Moore. The IDIAP smart meeting room. IDIAP-COM 07, IDIAP, 2002.

[87] N. Morgan, D. Baron, J. Edwards, D. Ellis, D. Gelbart, A. Janin, T. Pfau, E. Shriberg, and A. Stolcke. The meeting project at ICSI. In *Proc. of the Human Language Technology Conference*, San Diego, CA, March 2001.

[88] N. Morgan and E. Fosler-Lussier. Combining multiple estimators of speaking rate. In *Proc. of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP-98)*, 1998.

[89] A. Morris, A. Hagen, H. Glotin, and H. Bourlard. Multi-stream adaptive evidence combination for noise robust asr. In *Speech Communication*, 2001.

[90] S. Mota and R. Picard. Automated posture analysis for detecting learner's interest level. In *In* Proc. CVPR Workshop on Computer Vision and Pattern Recognition for Human Computer Interaction (CVPR–HCI)*, Madison Wisconsin*.

[91] K. Murphy. Dynamic Bayesian networks: Representation, inference and learning. *Ph.D. dissertation, UC Berkeley*, 2002.

[92] A. Nadas. Estimation of probabilities in the language model of the IBM speech recognition system. In *IEEE Transactions on ASSP*, volume 32, pages 859–861, 1984.

[93] R. M. Neal and G. E. Hinton. A view of the EM algorithm that justies incremental, sparse, and other variants. In *Learning in Graphical Models*.

[94] A. Nefian, L. Liang, X. Pi, L. Xiaoxiang, C. Mao, and K. Murphy. A coupled HMM for audio-visual speech recognition. *ICASSP*, 2002.

[95] D. Novick, B. Hansen, and K. Ward. Coordinating turn-taking with gaze. In *Proc. of the 1996 International Conference on Spoken Language Processing (ICSLP-96)*, 1996.

[96] R. Ofsche and M. Lee. Status, deference, influence and convenient rationalization: An application of two-process theory. Working Papers in Two-Process Theory 3, Department of Sociology, University of California, 191.

[97] N. Oliver, E. Horvitz, and A. Garg. Layered representations for learning and inferring office activity from multiple sensory channels. In *Proc. ICMI*, Pittsburgh, Oct. 2002.

[98] N. Oliver, B. Rosario, and A. Pentland. Graphical models for recognizing human interactions. *Proc. of Intl. Conference on Neural Information and Processing Systems NIPS 98.*, pages 343–348, 1998.

[99] N. Oliver, B. Rosario, and A. Pentland. A Bayesian computer vision system for modeling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8), Aug. 2000.

[100] E. Padilha and J. Carletta. Nonverbal behaviours improve a simulation of small group discussion. In *Proc. First Int. Nordic Symposium of Multi-modal Communication*, Copenhagen, Sep. 2003.

[101] E. Padilha and J. C. Carletta. A simulation of small group discussion. In *EDILOG*, 2002.

[102] K. C. H. Parker. Speaking turns in small group interaction: A context-sensitive event sequence model. *Journal of Personality and Social Psychology*, 54(6):965–971, 1988.

[103] F. Peng and A. McCallum. Accurate information extraction from research papers using conditional random fields. In *In Proceedings of Human Language Technology Conference and North American Chapter of the Association for Computational Linguistics (HLT-NAACL04)*, 2004.

[104] A. Pentland. Smart rooms. In *Scientific American*, volume vol. 274, pages 68–76, 1996.

[105] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. In *Proc. of IEEE*, volume 77, pages 257–286, 1989.

[106] L. R. Rabiner and B.-H. Juang. *Fundamentals of Speech Recognition*. Prentice-Hall, 1993.

[107] D. Reidsma, R. Rienks, and N. Jovanovic. Meeting modelling in the context of multimodal research. In *Proc. of the Workshop on Machine Learning and Multimodal Interaction*, 2004.

[108] S. Reiter, B. Schuller, and G. Rigoll. Segmentation and Recognition of Meeting Events using a Two-Layered HMM and a combined MLP-HMM Approach. In *Proc. 7th International Conference on Multimedia and Expo (ICME)*, Toronto, July, 2005.

[109] S. Renals and D. Ellis. Audio information access from meeting rooms. In *Proc. IEEE ICASSP 2003*, 2003.

[110] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn. Speaker verification using adapted Gaussian mixture models. In *Digital Signal Processing*, number 10, pages 1–3, 2000.

[111] R. Rienks and D. Heylen. Automatic dominance detection in meetings using easily detectable features. In *2nd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms*, Edinburgh, Scotland, 2005. Springer Verlag.

[112] R. Rienks, D. Zhang, D. Gatica-Perez, and W. Post. Detection and application of influence rankings in small group meetings. In *In the Eighth International Conference on Multimodal Interfaces (ICMI'06)*, 2006.

[113] Y. Rui, A. Gupta, and A. Acero. Automatically extracting highlights for TV baseball programs. In *Proc. ACM Multimedia*.

[114] K. Sato and Y. Sakakibara. Rna secondary structural alignment with conditional random fields. In *Bioinformatics*, page 237C242, 2005.

[115] L. K. Saul and M. I. Jordan. Mixed memory Markov models: Decomposing complex stochastic processes as mixtures of simpler ones. *Machine Learning*, 37(1):75–87, 1999.

[116] G. Schwarz. Estimating the dimension of a model. In *The Annals of Statistics*, number Vol. 6.

[117] F. Sha and F. Pereira. Shallow parsing with conditional random fields. In *In Proceedings of HLT-NAACL*, page 213–220, 2003.

[118] M. Skounakis, M. Craven, and S. Ray. Hierarchical hidden Markov models for information extraction. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, 2003.

[119] P. Smolensky. Information processing in dynamical systems: Foundations of harmony theory. In *In D. E. Rumelhart and J. L. McClelland, editors, Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1: Foundations. MIT Press*, 1986.

[120] T. Starner and A. Pentland. Visual recognition of American sign language using HMMs. In *Proc. Int. Work. on AFGR*, Zurich, 1995.

[121] C. Stauffer, W. Eric, and L. Grimson. Learning patterns of activity using real-time tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22 , Issue 8 (August 2000) pages: 747–757 August 2000.

[122] R. Stiefelhagen. Tracking focus of attention in meetings. In *IEEE International Conference on Multimodal Interfaces*, Pittsburgh, PA, 2002.

[123] C. Sutton, K. Rohanimanesh, and A. McCallum. Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. In *In Proceedings of the Twenty-First International Conference on Machine Learning (ICML)*, 2004.

[124] B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *In Eighteenth Conference on Uncertainty in Artificial Intelligence (UAI02)*, 2002.

[125] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, pages 260–269, 1967.

[126] A. Waibel, M. Bett, F. Metze, K. Ries, T. Schaaf, T. Schultz, H. Soltau, H. Yu, and K. Zechner. Advances in automatic meeting record creation and access. *in Proc. IEEE ICASSP*, May 2001.

[127] A. Waibel, T. Schultz, M. Bett, R. Malkin, I. Rogina, R. Stiefelhagen, and J. Yang. SMaRT:the Smart Meeting Room Task at ISL. In *Proc. IEEE ICASSP 2003*, 2003.

[128] J Wang, C Xu, E.S. Chng, and Q Tian. Sports highlight detection from keyword sequences using HMM. In *IEEE ICME*, Taiwan, June 2004.

[129] P. Wellner, M. Flynn, and M. Guillemot. Browsing recorded meetings with FERRET. In *MLMI04. Springer-Verlag*, 2004.

[130] C. Wojek, K. Nickel, and R. Stiefelhagen. Activity Recognition and Room Level Tracking in an Office Environment. In *IEEE Int. Conference on Multisensor Fusion and Integration for Intelligent Systems – MFI06*, Sept. 2006.

[131] B. Wrede and E. Shriberg. The relationship between dialogue acts and hot spots in meetings. In *Proc. ASRU*, Virgin Islands, Dec. 2003.

[132] B. Wrede and E. Shriberg. Spotting hotspots in meetings: Human judgments and prosodic cues. In *Proc. Eurospeech*, Geneva, Sept. 2003.

[133] C.R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, number 19(7).

[134] L. Xie, S.-F. Chang, A. Divakaran, and H. Sun. Unsupervised discovery of multilevel statistical video structures using hierarchical hidden Markov models. *ICME*, 28-30 July 2003.

[135] J. Yamato, J. Ohya, and K. Ishii. Recognizing human action in time-sequential images using hidden Markov models. In *Proceedings of the 1992 International Conference on Computer Vision*, 1991.

[136] J. Yang, L. Weier, and A. Waibel. Skin-color modeling and adaptation. *Proc. Asian Conf. on Computer Vision*, (II:687–694), 1998.

[137] L. Zelnik-Manor and M. Irani. Event-based video analysis. In *Proc. IEEE CVPR*, Dec. 2001.

[138] D. Zhang, D. Gatica-Perez, S. Bengio, and I. McCowan. Modeling individual and group actions in meetings with layered HMMs. In *IEEE Trans. on Multimedia*, June 2001.

[139] D. Zhang, D. Gatica-Perez, S. Bengio, I. McCowan, and G. Lathoud. Modeling individual and group actions in meetings: a two-layer HMM framework. In *Proc. CVPR Workshop on Event Mining in Video*, July 2004.

[140] D. Zhang, D. Gatica-Perez, S. Bengio, and D. Roy. Learning influence among interacting Markov chains. *Advances in Neural Information Processing Systems (NIPS)*, 18, 2005.

[141] H. Zhong, J. Shi, and M. Visontai. Detecting unusual activity in video. In *Proc. IEEE CVPR*, June. 2004.

[142] M. Zobl, F. Wallhoff, and G. Rigoll. Action Recognition in Meeting Scenarios using Global Motion Features. In *Proc. Fourth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS-ICVS)*, Mar. 2003.

Curriculum Vitae

DONG ZHANG

Permanent address:	No.49 Zhichun Road, Beijing, 100080	Phone:	+86 10 62617711
	Sigma Haidian District	email:	zhang@idiap.ch
	Beijing 100080, China	Citizenship:	China

Work Experience

2003 – IDIAP Research Institute, Switzerland
Research Assistant

2001 – 2003 Microsoft Research Asia, Beijing, China
Assistant Researcher

Education

2003 – Docteur ès Sciences (anticipated Dec. 2006)
Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland
Dissertation: *Probablistic Graphical Models for Human Interaction Analysis.*

1998 – 2001 Master, Chinese Academy of Sciences, Beijing, China

1994 – 1998 Bachelor, Beijing Institute of Technology, Beijing, China

Computer Experience

Operating Systems:	Solaris, UNIX / Linux, Windows
Computer Applications:	HTK, GMTK, MATLAB, DirectShow
Languages:	C, C++, JAVA.

Publications

Journals

1. D. Zhang, D. Gatica-Perez, S. Bengio, and I. McCowan, "Modeling Individual and Group Actions in Meetings with Layered HMMs," *IEEE Transactions on Multimedia* June 2006.

2. I. McCowan, D. Gatica-Perez, S. Bengio, G. Lathoud, M. Barnard and D. Zhang, "Automatic Analysis of Multi-modal Group Actions in Meetings," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* Vol. 27, No. 3, pp. 305–317, Mar. 2005.

Conferences and Workshops

1. R. Rienks, D. Zhang, D. Gatica-Perez, and W. Post "Detection and Application of Influence Rankings in Small Group Meetings," In *the Eighth International Conference on Multimodal Interfaces (ICMI)* 2006.

2. D. Zhang, D. Gatica-Perez, D. Roy, and S. Bengio "Modeling Interactions from Email Communication," In *IEEE International Conference on Multimedia and Expo (ICME)* 2006.

3. D. Zhang, D. Gatica-Perez, S. Bengio, and D. Roy "Learning Influence among Interacting Markov Chains," *In Advances in Neural Information Processing Systems (NIPS) 18*. MIT Press, 2005.

4. D. Zhang, D. Gatica-Perez, S. Bengio and I. McCowan "Semi-supervised Adapted HMM for Unusual Event Detection," *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)* 2005.

5. D. Zhang, D. Gatica-Perez and S. Bengio "Semi-supervised Meeting Event Recognition with Adapted HMMs," In *IEEE International Conference on Multimedia and Expo (ICME)* 2005.

6. D. Gatica-Perez, I. McCowan, D. Zhang and S. Bengio, "Detecting Group Interest-level in Meetings" *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* 2005.

7. D. Gatica-Perez, D. Zhang and S. Bengio, "Extracting Information from Multimedia Meeting Collections" *ACM SIGMM Information Workshop on Multimedia Information Retrieval, in conjunction with ACM Multimedia 2005*.

8. M. Al-Hames, A. Dielmann, D. Gatica-Perez, S. Reiter, S. Renals, and D. Zhang, “Multimodal Integration for Meeting Group Action Segmentation and Recognition” *Machine Learning for Multimodel Interaction (MLMI)* 2005.

9. D. Zhang, D. Gatica-Perez, S. Bengio, I. McCowan and G. Lathoud, “Multimodal Meeting Action Modeling Using Multi-Layer HMM Framework” *NIPS 2004 Workshop on Multimodal Signal Processing*.

10. D. Zhang, D. Gatica-Perez, S. Bengio, I. McCowan and G. Lathoud, “Multimodal Group Actions Clustering in Meetings” *ACM 2nd International Workshop on Video Surveillance and Sensor Networks, in conjunction with 12th ACM Multimedia* 2004.

11. D. Zhang, D. Gatica-Perez, S. Bengio, I. McCowan and G. Lathoud, “Modeling Individual and Group Actions in Meetings: a Two-Layer HMM Framework,” *the Second IEEE Workshop on Event Mining: Detection and Recognition of Events in Video, In Association with CVPR* 2004.

12. D. Zhang, S. Z. Li, D. Gatica-Perez, “Real-Time Face Detection Using Boosting Learning in Hierarchical Feature Spaces” *International Conference on Pattern Recognition (ICPR)* 2004.

13. P. Yang, S. Shan, W. Gao, S. Z. Li, D. Zhang, “Face Recognition Using AdaBoosted Gabor Features,” *International Conference on Automatic Face and Gesture Recognition*, 2004.

14. S. Z. Li, D. Zhang, C. Ma, H. Shum, and E. Chang, “Learning to Boost GMM Based Speaker Verification,” *Eurospeech, Geneva, Switzerland,* 2003.

15. X. Hua, D. Zhang, M. Li, H. Zhang, “Performance Evaluation Protocol for Video Scene Detection Algorithms,” *Workshop on Multimedia Information Retrieval, in conjunction with 10th ACM Multimedia* 2002.

16. Y. Wang, P. Zhao, D. Zhang, M. Li, H. Zhang, “MyVideos: a System For Home Video Management,” *ACM Multimedia,* 2002.

17. D. Zhang, W. Qi, H. Zhang, “A New Shot Boundary Detection Algorithm,” *IEEE Pacific Rim Conference on Multimedia (PCM),* 2001.

18. D. Li, D. Zhang, H. Lu, “Video Segmentation by Two Measures and Two Thresholds,” *IEEE IDEAL,* 2000.