

Unsupervised Pattern Discovery for Multimedia Sequences

Lexing Xie

Submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

in the Graduate School of Arts and Sciences

Columbia University

2005

© 2005

Lexing Xie

All Rights Reserved

ABSTRACT

Unsupervised Pattern Discovery for Multimedia Sequences

Lexing Xie

This thesis investigates the problem of discovering patterns from multimedia sequences. The problem is of interest as capturing and storing large amounts of multimedia data has become commonplace, yet our capability to process, interpret, and use these rich corpora has notably lagged behind.

Patterns refer to the recurrent and statistically consistent units in a data collection, their recurrence and consistency provide useful bases for organizing large corpora. Unsupervised pattern discovery is important, as it is desirable to adapt to diverse media collections without extensive annotation. Moreover, the patterns should be meaningful, since meanings are what we humans perceive from multimedia. The goal of this thesis is to devise a general framework for finding multi-modal temporal patterns from a collection of multimedia sequences, using the self-similarity in both the appearance and the temporal progression of the content. There, we have addressed three sub-problems: learning temporal pattern models, associating meanings with patterns, and finding patterns in multimodality.

We propose novel models for the discovery of multimedia temporal patterns. We construct dynamic graphical models for capturing the multi-level dependency between the audio-visual observations and the events. We propose a stochastic search scheme for finding the optimal model size and topology, as well as unsupervised feature grouping for selecting relevant descriptors for temporal streams.

We present novel approaches towards automatically explaining and evaluating the patterns in multimedia streams. Such approaches link the computational representations of the patterns with words in the video stream. The linking between the representation of audio-visual patterns, such as those acquired by a dynamic graphical model and the metadata, is achieved by statistical association.

We develop solutions for finding patterns that reside across multiple modalities. This is realized with layered dynamic mixture model, and we address the modeling problems of intra-modality temporal dependency and inter-modality asynchrony in different parts of the model structure.

With unsupervised pattern discovery, we are able to discover from baseball and soccer programs the common semantic states, *play* and *break*, with accuracies comparable to their supervised counterparts. On large broadcast news corpus we find that multimedia patterns have good correspondence with news topics that have salient audio-visual cues. These findings demonstrate the potential of our framework of mining multi-level temporal patterns from multimodal streams, and it has broad outlook in adapting to new content domains and extending to other applications such as event detection and information retrieval.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Problems addressed	4
1.2.1	Mining statistical temporal patterns	4
1.2.2	Assessing the meanings of audio-visual patterns	6
1.2.3	Discovering multi-modal patterns	6
1.3	Summary of findings	7
1.3.1	Summary of contributions	7
1.3.2	Prospective applications	8
1.4	Organization of the thesis	8
2	Prior work	10
2.1	Connection to other data mining problems	11
2.2	Multimodal processing	13
2.2.1	Machine Perception	13
2.2.2	Extracting multimedia features	16
2.3	Unsupervised learning	19
2.4	Video content analysis: from syntax to semantics	22
2.4.1	The syntactic gap and the semantic gap	23

2.4.2	Parsing multimedia syntax	24
2.4.3	Understanding multimedia semantics	24
2.4.4	Discussions	25
3	Mining statistical temporal patterns	27
3.1	Patterns in video	27
3.2	Summary of our approaches	29
3.3	Hierarchical hidden Markov model	31
3.3.1	Representing an HHMM	33
3.3.2	Overview of HHMM inference and estimation	34
3.4	Model adaptation	35
3.4.1	An overview of MCMC	35
3.4.2	MCMC for HHMM	37
3.5	Feature selection for unsupervised learning	39
3.5.1	The feature selection algorithm	40
3.5.2	Evaluating the information gain	41
3.5.3	Finding a Markov Blanket	43
3.5.4	Ranking the feature subsets	44
3.6	Experiments and Results	45
3.6.1	Parameter and structure learning	47
3.6.2	With feature selection	50
3.6.3	Testing on a different domain	52
3.6.4	Comparing to HHMM with simplifying constraints	53
3.7	Visualizing and interpreting multimedia patterns	55
3.8	Chapter summary	56

4	Assessing the meanings of audio-visual patterns	57
4.1	The need for meanings in multimedia	57
4.2	Cross-modal association in temporal streams	59
4.2.1	The temporal division of the multimodal streams	59
4.2.2	Tokenizing each modality	60
4.2.3	Co-occurrence analysis	62
4.2.4	Machine translation	65
4.3	Experiments	67
4.3.1	Visualizations of co-occurrence and machine translation	67
4.3.2	Word prediction and correlation with topics	70
4.4	Discussions	72
4.4.1	Related work in multi-modal association	72
4.4.2	Word association and beyond	73
4.5	Chapter summary	74
5	Discovering multi-modal patterns	75
5.1	Multi-modal patterns in video	75
5.2	Unsupervised asynchronous multi-modal fusion	78
5.2.1	The layered representation	78
5.2.2	Unsupervised mid-level grouping	79
5.2.3	The fusion layer	81
5.3	Experiments	82
5.3.1	Multi-modal features	83
5.3.2	Inspecting the clusters	85
5.3.3	News topics	87
5.3.4	Discussions	89

5.4	Chapter summary	90
6	Conclusions and future work	91
6.1	Research summary	91
6.1.1	Mining statistical temporal patterns	92
6.1.2	Assessing the meanings of audio-visual patterns	93
6.1.3	Discovering multi-modal patterns	93
6.2	Improvements and future directions	94
6.2.1	Applications of multimedia patterns	94
6.2.2	Working with unstructured data	96
6.2.3	Incorporating context	97
6.2.4	Multi-modal, multi-source fusion	97
	Appendix	99
A	The inference and estimation for HHMM	99
A.1	Representing an HHMM	99
A.2	Three question in HHMM inference and estimation	101
A.3	The forward-backward algorithm	103
A.4	The Viterbi algorithm	104
A.5	Parameter estimation	105
A.6	The complexity of learning and inference with HHMM	108
B	Model adaptation for HHMM	109
B.1	Proposal probabilities for model adaptation	109
B.2	Computing different moves in RJ-MCMC	110
B.3	The acceptance ratio for different moves in RJ-MCMC	111
C	Low-level visual features for sports videos	113
C.1	Dominant color ratio	113

C.2	Motion features	116
References		119

List of Figures

1.1	The accumulation of content in different domains.	3
3.1	Graphical HHMM representation at level d and $d + 1$: (A) Tree-structured representation; (B) DBN representations, with observations X_t drawn at the bottom. Uppercase letters denote the states as random variables in time t , lowercase letters denote the state-space of HHMM, i.e., values these random variables can take in any time slice. Shaded nodes are auxiliary <i>exit</i> nodes that turn on the transition at a higher level - a state at level d is not allowed to change unless the exiting states in the levels below are <i>on</i> ($E^{d+1} = 1$).	31
3.2	Visualization of the MCMC stochastic search strategy for model selection.	36
3.3	Feature selection algorithm overview	40
3.4	Comparison with HHMM with left-to-right transition constraints. Only 3 bottom-level states are drawn for the readability of this graph, models with 6-state sub-HMMs are simulated in the experiments. . .	53
3.5	HHMM model visualization with feature distributions and video storyboard.	55

4.1	Generating co-occurrence statistics from the HHMM <i>labels</i> and word <i>tokens</i> . The grayscale in the co-occurrence matrix indicate the magnitude of the co-occurrence counts, the brighter the cell, the more instances of word w and label q that were seen in the same story. . . .	62
4.2	The analogy between (a) metadata association in multimedia streams and (b) machine translation in language. $t(\cdot \cdot)$ denote the probability under the ideal underlying model; $c(\cdot \cdot)$ denote the observed co-occurrence statistic using imprecise alignment.	64
4.3	The <i>generation</i> of a sentence of N_f French words from N_e English words according to the unigram machine translation model $t(f e)$, i.e., <i>Model 1</i> by Brown et. al. [21]. The clear nodes (the correspondences) are hidden; the shaded nodes (the French words) are observed. $U[1, \dots, N]$ denotes a uniform distribution on $\{1, \dots, N\}$	66
4.4	Word-association with likelihood ratio before and after machine translation. The likelihood ratio function L is rendered in log-scale.	70
4.5	Word-association with labels generated by hierarchical HMM and K-means. The likelihood ratio function L is rendered in log-scale.	71
5.1	Asynchronous multimodal streams in news.	76
5.2	The layered dynamic mixture model, in the dimensions of time, various modalities and perceptual levels. Shaded nodes: observed, clear nodes: hidden; different colors represent nodes in different modality.	78
5.3	Models for mapping observations to mid-level labels. (a) Hierarchical HMM; (b) PLSA.	80

5.4	Example clusters, each row contains the text transcript and the keyframes from a story. Left: cluster #5, sports, precision 16/24. Right: cluster #29, weather, precision 13/16. The models are learned on CNN set A, evaluated on set B.	85
5.5	The most probable features predicted by the model (red) and observed in the clusters (navy). A magenta cell results from the joint presence from both, and a blank one indicates that neither has high confidence. The features retained during the automatic feature selection process are shown in blue. The models are learned on CNN set A, evaluated on set B.	86
5.6	Two example stories from TDT-2 corpus [121] that exhibit intra-topic diversity and inter-topic similarity.	89
6.1	Dominant color ratio as an effective feature in distinguishing three kinds of views in soccer video. Left to right: global, zoom-in, close-up. Global view has the largest grass area, zoom-in has less, and close-ups has hardly any (including cutaways and other shots irrelevant to the game).	114

List of Tables

3.1	Sports video clips used in the experiment.	46
3.2	Evaluation of learning schemes (1)-(4) against ground truth on clip <i>Korea</i>	49
4.1	Example word-label correspondences. <i>HHMM</i> label (m, q) denotes the q^{th} state in the m^{th} HHMM model; the <i>visual concepts</i> are the features automatically selected for model m ; the <i>predicted word-stems</i> are the entries of $L^c(w, q)$ that have values in the top 5%.	72

Acknowledgements

Over the last five years I have had the privilege to work with groups of terrific mentors and colleagues, who have made my time at Columbia rewarding and enjoyable. Without them this dissertation would not be possible.

First and foremost, I would like to thank my advisor, Shih-Fu Chang. He has been a source of excellent guidance and great support throughout my graduate study. His sharp insights and his emphasis on the demonstrability of ideas have immensely influenced my research. His constant encouragement to look beyond improving the state of the art, and the precious freedom he granted for pursuing research directions that satisfy my interest and curiosity has made research fun and rewarding.

I would like to express my gratitude to members of my dissertation committee and other mentors who have guided me through my research path. Many thanks to Dan Ellis for his insightful comments and discussions over the years, Ajay Divakaran for his unwavering encouragements and collaborations, Tony Jebara for his feedback that improved the manuscript and helped me look beyond the thesis, and Xiaodong Wang for the discussions that shaped the work in this thesis. Thanks to Patrick Pérez for giving me the opportunity to intern at Microsoft Research Cambridge and guiding me through a fun summer in the machine learning and perception group. Also thanks to Nevenka Dimitrova for being a wonderful mentor and a great source of support and encouragement.

I would like to thank the current and former fellow students of the Digital Video and Multimedia (DVMM) lab for making the group a stimulating and memorable place to work and to learn – Hari Sundaram, Dongqing Zhang, Winston Hsu, Lyndon Kennedy, Tian-Tsong Ng, Yong Wang, Jessie Hsu, Eric Zavesky, Akira Yanagawa, Ana Benitez, Shahram Ebadollahi, Alejandro Jaimes, Peng Xu, Raj Kumar,

William Chen and Hualu Wang. I also thank members of LabROSA for discussions and moments of fun and inspirations: Adam Berenzweig, Manuel Reyes-Gomez, Marios Athineos, Keansub Lee, Michael Mandel, Graham Poliner and Sambarta Bhattacharjee.

I am grateful to Mitsubishi Electric Research Laboratories (MERL) for their support during the first four years of my graduate study. Special thanks to Huifang Sun, Anthony Vetro and Regunathan Radhakrishnan for their input and collaborations.

I also thank the multimedia research teams in IBM T. J. Watson Research center for the fruitful interactions and collaborations: John R. Smith, Ching-Yung Lin, Milind Naphade, Apostol Natsev, Jelena Tesic, Giridharan Iyengar, Murray Campbell, Belle Tseng and Harriet Nock.

I thank all my other fellow Ph.D. students, colleagues and friends who shared my academic life in various occasions, without them my graduate school experience would not be as pleasant and colorful.

Chapter 1

Introduction

This thesis presents a computational framework for the unsupervised organization of large collections of multimedia streams.

1.1 Motivation

With the rapid advances in recording devices and disk drives, large amounts of multimedia data now seem commonplace in personal, educational, business or entertainment environments. Affordable consumer and prosumer cameras allow us to capture our daily events with ease. Advances in storage devices make archiving large amounts of produced and consumer content no longer a luxury. As a result, many of us have tens of thousands of digital pictures or hundreds of hours of home videos on our home PCs; many educational institutions have thousands of hours of lectures on tape; large corporations have archives for seminars, meetings, and presentations; security and intelligence departments monitor tens or hundreds of channels of news or closed-circuit surveillance at any time. As shown in [Figure 1.1](#), these multimedia corpora are already commonplace with camcorder/cameras at several hundred US dollars and personal video recorders (PVR) that can store 200~400 gigabytes. It

is convenient to preserve the raw data and casually consume them whenever we like, but we often found ourselves in a practical paradox that much more bits do not lead to more useful information. At this scale, linear access seems infeasible already, yet annotating all the streams are even more daunting as it will require around ten times real-time. i.e., it would have taken us more than three months to look through and annotate four years' worth of personal photos, three years and seven months for an entire semester's lecture videos, and two and a half months for news programs from any one channel, assuming twenty-four hour working days non-stop. Furthermore, linear browsing and simple tagging would not be able to meet our daily needs for accessing the data. For instance, it would not be a quick task if we were to count: how many people went for the camping trip last summer; in how many different ways have color representation being covered each year in the digital image processing class; what were the types of discussions that usually happen in a project meeting or an executive meeting; how many times have topics related to greenhouse effect been mentioned in today's news.

Identifying and recognizing important concepts for a multimedia corpus are among the ultimate goals of multimedia analysis. They are however extremely difficult tasks, given that they rely on many of the open issues in computational vision, audition, linguistics, knowledge representation and information retrieval.

The goal of this thesis is necessarily much more modest and limited in scope – we resort to data-driven approaches and use the self-similarity in a domain to discover patterns, the recurrent and statistically consistent units in a data collection.

Patterns are ubiquitous across many domains. For example, textures are spatial patterns in images; association rules are relational patterns in transaction databases; melodies and rhythms are patterns in music; wedges and cycles are temporal patterns in time series. Multimedia patterns distinguish themselves from patterns from

corpus	a personal album 2001~2005	engineering lectures cvn.columbia.edu spring 2005	ABC news year 2005
			
amount	3,000 photos/yr × 5 yrs = 15,000 photos	$2\frac{1}{2}\text{hr/week} \times 13\text{weeks} \times$ 114courses/semester = 3,075 hrs/semester	$\frac{1}{2}\text{hr/day} \times 365\text{ days/yr} =$ 183 hrs/yr
storage	8.8GB	1,482GB	120GB
access time	250 hrs @1sec/photo	3,075 hrs	183 hrs
annotation time	2,500 hrs ~ 3.5 mo.	30,750 hrs ~ 3.6 yrs	1830 hrs ~ 2.54 mo.

Figure 1.1: The accumulation of content in different domains.

earlier data mining domains in two ways: they exist across multiple input modalities in addition to being persistent in space and time¹, and they represent semantics in addition to the apparent syntactic structures. Multimedia patterns form useful representations for detecting events, as well as summarizing and browsing multimedia content. For instance, most sports videos consist of regular plays, highlights and breaks; many news programs include a weather forecast, a financial report, several well-structured stories and a few headline briefings; surveillance videos can be categorized into the background, the frequent actions and the rare events.

In order to identify these patterns, it would be desirable to prepare sufficient training data and learn computational representations for each domain. However,

¹The multi-modal perspective is seen in problems from a few other domains such as multi-sensor fault diagnosis [77].

this is not always possible, because (a) we may not have extensively annotated data; (b) the domain may be unknown (e.g., content from different TV broadcasters, surveillance systems to be deployed at a new site, video and still cameras used by different households); (c) the set of ground truth varies both across different collections and even over time for the same collection, making it necessary to handle recognition targets not previously defined. Hence the flexible alternative of unsupervised pattern discovery is preferred, as unsupervised approaches will be able to adapt to new data without retraining. In this approach, we learn a statistical description for the patterns and simultaneously identify the instances of such patterns in the content.

1.2 Problems addressed

We address three connected subproblems within multimedia pattern discovery. In the following subsections we shall overview the scopes and the intuitions behind our solutions to each of them.

1.2.1 Mining statistical temporal patterns

In this work we are interested in finding patterns from video that are of consistent statistical characteristics. Literally, *pattern* is defined as “*an example, an instance, esp. one taken as typical, representative, or eminent.*” from *Oxford English Dictionary* [123].

The multimedia pattern discovery problem can be decomposed into two subproblems with inter-related solutions: choosing suitable representations of the raw data, and learning a model from these representations. The first problem can be formulated as choosing a subset from a pool of representations (features). This problem

is due to the sheer volume and the high redundancy in the raw media signals. Unlike the electrocardiogram signals or the transaction itemsets for a supermarket, it is less obvious how to abstract audio-visual data into symbolic or numerical representations. The second problem is concerned with learning a model to capture the salient recurrence in the data streams without supervision. The statistical recurrence in audio-visual sequences differs from clusters of data points in two ways: (1) the data are often highly correlated in time, and the patterns are in the relationships between data points as well as in the data values themselves; (2) the patterns are continuous segments in the long media sequence with unknown boundary, and we cannot assume a fixed pattern length or known temporal cycles a priori. Our solution of the feature selection problem builds on top of the model-learning problem.

We use hierarchical hidden Markov model (HHMM), a special form of dynamic Bayesian network, to tackle the model-learning problem. HHMM is versatile enough to handle the dependency between adjacent audio-visual observations, the flexible temporal boundaries, and model the dependency between events, yet it is simple enough that we can carry out exact inference. In order to adapt to new domains with varying descriptive complexity, the size of the model is automatically determined using the minimum description length principle in conjunction with stochastic search. For the feature selection problem, algorithms abound under supervised learning scenarios, while the unsupervised learning scenario over temporal data sequences is rarely addressed, due to the difficulty in evaluating relevance without ground truth. We alter the notion of feature relevance to allow multiple relevant feature subsets, since different patterns in the same sequence may require different representations. For example, the plays or breaks in sports videos are reflected in the visual cues, while the highlights are more identifiable from the audio cues (e.g., the excited commentary and the cheers from the audience). We also re-define relevance to

be relative among the candidate features using information-theoretic criteria, with which we partition the original feature pool into several subsets. We then evaluate Bayesian dependencies within each subset so as to filter out the redundancies.

1.2.2 Assessing the meanings of audio-visual patterns

Multimedia patterns are only useful when the syntax is interpreted with respect to the underlying semantics. These interpretations link the model and the semantic concepts in a particular domain. Such links can be hard to identify when the semantics are diverse, complex or unknown. In news videos, for instance, we would not immediately know which ones of the hundred labels generated by the model should correspond to politics, war, hurricane coverage, weather reports, or the financial sections. It is hence desirable to have computational models automatically establish the links from the syntactic patterns to the semantics.

This process of interpreting patterns is enabled by the associated text transcripts or other metadata. The information they carry complements those in the audio and visual channels, and they are much easier and natural for humans to understand. We devise statistical models to estimate the probabilities of associating certain words given the audio-visual patterns (and vice versa) via co-occurrence analysis and statistical machine translation. In news videos this association explains a few audio-visual clusters with groups of words, and these word groups in turn indicate consistent topical themes such as politics or weather.

1.2.3 Discovering multi-modal patterns

Watching video is a multisensory experience, salient recurrent patterns in video will necessarily involve the visual, audio, and text channels. For example, anchor shots in news videos are salient visual patterns, however if jointly viewed with the audio or

closed captions these very similar shots may cover very different stories. Moreover, the different modalities in video are asynchronous as the signal rates vary from a few dozen bits (e.g., words) to a few thousand bits (e.g., audio) per second, and the semantics are often not aligned in time due to production syntax, making the fusion of different streams and the discovery of patterns across modalities is a non-trivial task.

We propose a layered dynamic mixture model for finding multi-modal patterns. The model is designed with a multi-layer structure that separates temporal correlation and cross-modal dependence, with one of the layers accounting for the temporal correlation and the variability within each modality, and the other fusing the information from all the modalities with loose temporal binding.

1.3 Summary of findings

We now summarize the findings of this thesis in solving the three sub-problems. In temporal pattern discovery we have proposed novel models for pattern discovery, model selection and feature selection, in sports videos the patterns discovered using low-level audio-visual features correspond to the basic semantic units play and break in different games. In news videos we have identified a few meaningful audio-visual clusters associated with a group of related words, furthermore we have found a few multi-modal patterns that have better correspondence to news topics than text-based techniques.

1.3.1 Summary of contributions

The original contributions of this thesis are as follows:

- A computational model for the unsupervised discovery of temporal patterns.

- A stochastic model adaptation scheme for determining the optimal model complexity.
- An unsupervised feature selection algorithm for temporal streams.
- A model for the automatic annotation of temporal video patterns with meta-data.
- A model for the fusion of asynchronous multi-modal streams.

1.3.2 Prospective applications

Multimedia temporal patterns can reveal the unique syntax in a given domain, so naturally it is useful for browsing and frequent event detection, as shown in [chapter 3](#) and [chapter 5](#), respectively. The domain-specific patterns can also provide global structure information that can help rare event detection, multimedia retrieval, and summarization applications. In addition to the few domains tested, the pattern discovery problem is applicable to a much wider scope, including unstructured video recordings such as surveillance videos, home videos, as well as other multi-modal streams such as personal lifelogs and business logs.

1.4 Organization of the thesis

The rest of this thesis is organized as follows. In [chapter 2](#) we review relevant prior work in data mining, machine learning and multimedia analysis. In [chapter 3](#) we discuss the problem of discovering syntactical patterns in temporal sequences. We use hierarchical hidden Markov model as the computational tool for pattern discovery, we present stochastic search strategies with Markov chain Monte Carlo for automatic adaptation of model complexity, and we also present the unsupervised

partition and selection from a pool of features for optimal pattern representation. In [chapter 4](#) we discuss the problem of identifying meanings for the patterns and present our solution using co-occurrence analysis and statistical translation. In [chapter 5](#) we discuss the problem of finding patterns across multiple streams, and we present the layered dynamic mixture model for the clustering of multiple asynchronous modalities in video. In [chapter 6](#) we present the conclusions and discuss a few future research directions. Appendix [A](#) introduces the inference of the hierarchical hidden Markov model, appendix [B](#) provides the details for the stochastic search algorithm for model selection, and appendix [C](#) include the feature extraction algorithms for the low-level color and motion features specially designed for sport videos.

Chapter 2

Prior work

The problem of multimedia pattern discovery closely relates to research in several areas. The abstract problem closely resembles the data mining problem of extracting frequent recurrences in various data collections. Multimedia pattern mining draws upon techniques from multi-modal perception so as to extract meaningful representations from the raw media streams. It uses unsupervised learning to uncover the target concept subject to the non-deterministic content characteristics and measurement noises.

In [section 2.1](#) we shall review some of the prior work in data mining and comment on the similarities and differences between patterns in multimedia and those in conventional domains. In [section 2.2](#) we shall briefly mention human and machine perception principles and summarize common multimedia features. [section 2.3](#) include a few statistical pattern recognition techniques, with an emphasis on unsupervised and semi-supervised models that are most relevant to our pattern discovery task. In [section 2.4](#) we shall discuss prior research efforts in multimedia content analysis.

2.1 Connection to other data mining problems

The problem of mining unknown recurrent patterns from massive datasets is being solved in many domains.

The celebrated shopping-basket problem [1] in database mining is concerned with the problem of finding association rules among the “transactions” in the space of “supermarket goods”. In this problem, interesting structures takes the form of *large itemsets*, where an *itemset* is an n -tuple of items, *large* is defined with the notion of *confidence* and *support* to characterize how often items appear together. Fast algorithms [2] for discovering large itemsets rely on the property that an itemset can be large if and only if all of its subset are large, thus reducing the search complexity. Temporal constraints in transaction time [3] and the lexical structure of items [113] (such as apple, fruit, produce) can be incorporated as additional dimensions to help expand sensible correlations. Additional speed-up on huge collections of transactions can be achieved via sampling [125] or asynchronous update of the candidate sets [20]. These association rules and constraints are deterministic in nature, and the original notion of *confidence* and *support*, however, does not cover exclusion relationships between items. In fact, association rules shall generalize as finding the correlations [19] and implications [20] in data with statistical tests, which are dependency relationships, conditional probabilities, or likelihood ratios in spirit.

Biological sequence analysis faces similar challenges: a few interesting and relatively conserved motifs (e.g., short DNA or protein segments that are similar) are hidden among the majority of background “noise” in a large number of long gene sequences. Motif sampler [73, 71] posed the motif discovery problem as “aligning” all sequences in the database at the position of the (only) motif, learning a product-of-multinomial model for the motif. The *i.i.d* (independently and identi-

cally distributed) assumption in the motif sampler seems too restrictive, so MEME (motif-based hidden Markov modeling of biological sequences) [8] and HMM [26] use a markov model to model the sequential dependency of the sequence observations. In this context, the motif and the background models are probabilistic, while the raw observations are symbolic and uni-dimensional in nature.

In Internet traffic analysis and performance modeling, patterns are directly derived from the continuous-valued measurements. They are often analyzed in the forms of periodicity, peaks, tail and trend behavior and so on. Continuous state-space statistical temporal models such as the auto-regressive moving average model (ARMA) [58] model can be learned to capture the structure of interest while at the same time taking into account the noise.

Intuitively, video mining tries to find consistent subsegments that re-occur in the collection. It may well be cast as finding association rules in an audio-visual concept space, aligning video motifs, or identifying trends in video while filtering out noise. A dynamic video structure model can be a “dense” version of the motif sampler [71] without explicit modeling of the background, and a generalization from the MEME [8] model by introducing dependencies across different features measured at the same time. Note however, there are a few domain differences: (1) The noise in the data entities. The shopping basket and motif finding problem are concerned with *clean* signals in that “coffee” would not be confused with “milk” and neither would “G” with “I” in amino acids, while measurement noise (such as the noise in CCD sensors and the impreciseness in camera motion estimation) must be taken into account in multimedia. Therefore, the uncertainty lies only in structure for data mining yet both in the structure and the measurement for multimedia. (2) The absence of known temporal boundaries and fixed time scales. In the data mining examples, the notions of “baskets” or long candidate sequences or the fixed daily or

weekly cycles are unambiguous. However, many video sequences do not have a clear boundary for “patterns”, nor does the time scale agree even within the same video genre (e.g., the length of a news story or a movie scene would vary [117, 66, 53]).

(3) The need for identifying meanings. The *interestingness*, or evaluation criteria for the target patterns are better-defined for domain experts, so that signal-level representations such as ”coffee and cream” (a shopping habit) or “VGIIGAG” (a biological control signal) or “a high peak around noon” (a browsing pattern) are readily useful. On the contrary, the meaning of “a human scream and a face closeup followed by fast camera pan” can vary across content collections, users, or tasks.

2.2 Multimodal processing

The processing units for multimodal signals are the necessary front end for any recognition systems. In this section we first give an overview of the general principles of visual, audio and language processing, followed by a review of common features used in video and multimedia analysis that make use of these principles. Throughout this work, the word “modality” is used somewhat liberally in its psychological sense, “a category of sensory perception” from *Oxford English Dictionary* [123]. For the multimedia analysis in particular we are mainly concerned with the *computer sensories* of vision, audition and text, although other modalities (such as touch, gesture, handwriting or biometrics) can be incorporated in appropriate occasions.

2.2.1 Machine Perception

Mimicking or reproducing the human processing of multi sensory signals is the holy grail of machine perception. On one hand it is natural to base our algorithm on the understanding of human perception obtained by psychological and physiological

studies, on the other under the currently incomplete picture of human perception there are many useful algorithms that can approximate simple perceptual tasks.

Vision and audition The generation and perception of light and sound in the environment share a similar trilogy in their *life-cycles*: the production of the mechanical or electromagnetic waves, the interaction of the signal with the environment, and the neurophysiological and psychological process of perception and understanding. Hence it is natural to look at vision and audition side-by-side.

The first two processes of signal generation and transmission are studied in acoustics and optics, the results of which have been very useful in graphics and sound rendering to synthesize visual and aural experiences. The fact that satisfactory synthesis is possible by implementing only a subset of the signal production and transmission constraints is because of the compensation for impreciseness during third process, human perception.

The studies in aural and visual perception [136, 129, 18] recognize similar principles such as constancy, perceptual grouping and hierarchical processing structures. One useful model of vision is presented in *Vision* [78], one of the first viable theoretical foundations for computational models of perception, where Marr mandates that the analysis hierarchy in vision lies in three increasingly abstract layers: *implementation*, *algorithm* and *computational theory*. This is a purely bottom-up view that our visual system works by building up progressively complex representations of the visual world until a full world model is constructed in our heads. Although useful, its correctness was later disputed psychologically and philosophically [108, 28], nonetheless the idea of having world models at different but not necessarily independent levels still stands on neurophysiological grounds measured by e.g. neural response time [22].

Natural language processing and information extraction. The percep-

tion of language differs from vision and audition in that text generation and consumption does not involve a physical process. Furthermore the language representation is already symbolic, rather than numerical. Although generally regarded as a more faithful representation of *meanings*, text still needs to go through a few stages of processing [65] in order to better approximate the *semantics*. These stages include: morphological (word-stemming), syntactical (part-of-speech tagging), semantic (lexicon, word sense disambiguation) and pragmatic (discourse and dialogue analysis). The goal of these processing steps are to summarize the content and reveals the underlying intention of the speaker/writer that is not obvious from the words.

Multi-modal fusion. We encounter the world as a multi-sensory experience, the machine analysis of this experience should also be multi-modal. Multi-modal phenomena have been investigated extensively in psychology and neuroscience, using both behavioral studies and neuroimaging measurements of neuronal activities. In behavioral studies the subjects are presented with the stimuli and are asked to either describe their experience or answer related questionnaires. Examples of cross-modal influence of the human percepts have been found in the literature, such as the McGurk effect for audio-visual hearing where the overlay of an audible syllable (“ba”) onto videotape of a speaker mouthing a different syllable (“ga”) would result in a perception of “da” [79]. Neuroimaging studies builds upon the knowledge of the functional division in the brain and measures the metabolic or electromagnetic signals in the regions of interest. And these studies have found additional localized brain activities during multi-modal or cross-modal tasks, such as audio-visual object recognition [44] or silent lip-reading [23]. Studies also suggest that multimodal fusion not only happens in both the early and the late stages of neural perception, but also happens both in the bottom-up and the top-down fashion where the attention

and context can influence what are being “perceived” [22]. While there is still a large gap between the current understanding of the multi-modal neural systems and the capability of our algorithms, there are useful references for computational fusion strategies that we may draw from existing observations, such as a preference for flexible computational architectures in the information fusion process over pure early or late fusion only, or a uni-directional information flow.

2.2.2 Extracting multimedia features

Features (or descriptors) are “the measurements which represent the data” [83]. Features not only influence the choice of subsequent decision mechanisms, their quality is also crucial to the performance of the learning system as a whole. Considerable amounts of work have been, and are continuing to be put into extracting good features from image sequences, audio clips and text documents.

We view features as either the *bottom-up* or the *top-down* type depending on the absence or presence of infused semantics and domain knowledge. Distinguishing features based on the knowledge they need emphasizes the general process of information fusion regardless of the perceptual modality being addressed, while de-emphasizing the distinction in each modality. Examples of the *bottom-up* type include generic color histogram, edge descriptor, spectral features and word frequency; the *top-down* type include face detection score, audio types or named-entity extraction. Features in the former category rely on generic assumptions about perception, and those in the latter use the infused knowledge in the forms of a training corpus or additional domain assumptions. The *bottom-up* features are more generalizable to different domains, while the *top-down* ones can better capture the saliency within domains of interest. In the rest of this section we shall review a few features used in the subsequent chapters.

Common bottom-up features include:

- Color and texture descriptors. These descriptors can be categorized as global or local depending on the range of spatial information used. Global color descriptors captures the global distribution or statistics of colored pixels, such as the color histogram [48], the distribution of pixels in quantized color space, or the color moments [114], the moment statistics in each color channel. Local descriptors contain information on a small region in the image, such as directional filter responses at different scales [76], or the principal components of the patch intensity values. Global descriptors are computationally efficient and compact in their representation and invariant to scale changes, while local descriptors can more accurately reflect object or region level information. Texture features describe the rate of change in image intensity, and are usually captured by filter responses. Popular filters include Gabor filters [95] and steerable filters [42]. The color correlogram [55] captures both color and spatial variation by globally expressing how the spatial correlation of pairs of colors changes with distance, and it has been shown to out-perform color histograms alone in image retrieval tasks. These features have been effective in texture or color image classification and retrieval [110, 126], object matching and detection [40, 76] tasks under controlled conditions, and are being actively used in state-of-the-art visual retrieval systems. For specialized domains, we can achieve further reduction of dimensionality with content constraints. For example, dominant color ratio ([33, 132], appendix C) computes the percentage of the most-frequent color in a frame, where the value and range of the dominant color are learned by aggregating the global color histogram over a long segment of video. This feature is useful for content domains constrained

in their scene location which have only a limited number of distinct colors, such as most types of sports.

- Motion features. Accurate estimation of true motion in the scene from one single view is a hard problem [51], solutions are often hard to generalize and computationally expensive. Various schemes have been proposed to generate fast approximations to the true motion field, such as a motion intensity descriptor that computes the mean and variance of block-based translation estimates [60], and a camera motion estimate from a least-squares fit to the motion field ([119], appendix C).
- Audio features. Generic audio features [45] captures some aspects of the sound generation or propagation process, and they have been very useful in speech recognition [98] and general classification tasks [105]. These features can include: waveform features such as the zero-crossing rate (ZCR); energy features such as volume or sub-band energy; spectral features such as spectral roll-off and mel-frequency cepstral coefficients (MFCC) [56]; model-based features such as the linear prediction coefficients (LPC) [98].
- Text features. The simplest feature from text documents shall be the unigram or n-gram frequency, i.e., the counts for words and adjacent n-tuple of words, usually after stemming [96] and stop-word removal [104]. Though simplistic, the unigram is still widely used since it suffers less from the sparsity than the higher-order statistics, and it does not propagate noise due to recognition and parsing errors.

The *top-down* features, sometimes are also referred to as *mid-level* features. The term *mid-level* is adopted to signal its difference with low-level perceptual

quantities, and high-level semantics, e.g. green color and hairy texture versus grass versus picnic in a park. The extraction of such features often requires some training, either in the form of learning a model over a set of labeled data or specific domain knowledge. Reliably extracting mid-level features has been an active topic both in research explorations [130, 5, 127] and in common benchmarks [122]. The recent progress on this end has enabled our use of multimedia concept detectors to infer high-level semantics. Examples of visual detectors include [40, 5] face, car, vehicle, people, sports, animal, flag and so on; audio detectors include [100] speech, music, noise, male/female voice etc.; detectors on text document include named-entity extraction [9]; multi-modal detectors include monologue [5]. If we were to set a threshold for reliable detector performance at around 50% in order to be usable by higher-level analysis algorithms, a number of the detectors such as face, car, people, sports, or weather have good performances if sufficiently trained across diverse datasets or in controlled domains.

While we choose not to focus on developing feature detectors in this thesis, improvements in detector quantity and quality will certainly help high-level inference tasks.

2.3 Unsupervised learning

The majority of research in pattern recognition techniques has been on learning a decision function over a labeled corpus [36], however collecting such a corpus is often expensive, and the simple abstraction of learning labels from independent instances may not be adequate for complex data structures. Therefore learning unobserved or partially observed labels from large quantities of data has been a topic of much theoretical and practical interest. The solution to this problem is generally

formulated as optimizing a model fitting criteria (likelihood, margin, distance, and variants) over both the (partial) labels of the data and the model parameters.

The models we use in the next three chapters clearly belongs to the generative model category below. While this choice is based on the considerations of its flexibility in handling complex dependencies and the computational efficiency, we shall also briefly summarize two potential alternatives for unsupervised learning, namely, discriminative learners with close ties to support vector machines, and spectral methods.

Generative mixture models [64]. Assuming that the data come from some identifiable mixture distribution, and clusters under each mixture represent the underlying labels. The Gaussian mixture model (GMM) and mixture of multinomials are the simplest form of such models for continuous and discrete-valued observations, respectively, where k-means is a degenerative case of the GMM with isotropic covariance and “hard” cluster assignment. The mixture models are learned via the Expectation-Maximization (EM) algorithm [32] that iterates between the mixture posterior estimation and update of the mixture parameters, and it converges to a local maximum in the log-likelihood landscape. The mixture model is actually the simplest graphical model by treating the unknown *label* of a data point as a hidden variable. A general graphical model is composed of more variables (hidden or observed) and additional dependency constraints among them, and missing labels or uncertainty in existing labels can be incorporated as additional hidden variables into this process [72, 90]. The complexity of the inference of an *i.i.d.* M -mixture models on N samples is $O(MN)$, the complexity of exact inference on a more complex model scales linearly with the number of examples N and exponentially with respect to the clique size in the triangulated graph [64], i.e., the size of the mutually “dependent” set of variables. This complexity can be significantly reduced

with approximation algorithms or carefully designed structures. Graphical models have found much practical use in many semi-supervised and unsupervised learning tasks such as object recognition [130], models of human motion [112] and speech recognition [99, 11].

Transductive support vector machines (SVM) [63] aim to simultaneously find both a set of labels for the unlabeled data points and the maximum margin linear separation in the labeled and unlabeled data. Intuitively, unlabeled data shall guide the separation boundary to avoid densely populated regions, and maximum margin reduces the generalization error bound. Finding the exact transductive SVM is NP-hard, but approximation algorithms seem to work well in practice. The complexity of training an SVM is dominated by solving the quadratic program, which scales between linear and cubic of the sample size requiring quadratic memory, but sequential optimization techniques [94] can reduce the memory requirement to $O(N)$ while reducing the time complexity from $O(N^3)$ up to $O(N)$. For the purpose of multimedia analysis, while SVMs are effective in separating labeled data and handling high-dimensional observations [126], they require custom designs to incorporate dependency within the data (e.g., specialized kernels [75]), the results are harder to interpret, and they can incur a higher computational cost than generative models.

Graph cut [12, 89, 107] methods define graphs where the nodes are the labeled and unlabeled examples and the edge weights represent the proximity among these examples. The optimal partition of the data is the one that minimizes the edge weights (or normalized weights) that cross the partition boundaries, and the solution can be found with eigenvalue methods. This framework is flexible in that it handles clusters of non-isotropic shapes, or clusters in a non-metric space. It also has an intuitive connection between the data distribution and the labels. The complexity involved is generally $O(N^3)$ in time and $O(N^2)$ in memory for solving the eigenvalue

problem, and extra efforts are needed for scaling up the algorithm to more than a few thousand samples. Random walk [118, 80] methods traverse the manifold structure in the dataset with a Markov random walk. Its transition probabilities are determined by the distances between examples, and this Markov transition matrix is connected to the Laplacians of the graph in spectral clustering algorithms [27, 81]. The Markov random walk method is useful in exploiting non-isotropic cluster structures, however it is difficult to scale up to high dimensions where the distances are ill-defined.

2.4 Video content analysis: from syntax to semantics

The analysis of multimedia content, video streams in particular, can be categorized into parsing the syntax and recognizing the semantics. The *syntax* and *semantics* of multimedia content are analogous to their original definitions in linguistics. There, syntax refers to the study of *rules, or patterned relations* that a (word) sequence is formed, and semantics refers to the *meanings, or relationships of meanings* in the sequence [82]. In multimedia, syntax includes the quantities, actions or operations, or the relations of multiple such entities during the generation of the multimodal signals or the capturing of multimedia sequence. Examples of syntactic elements include a shot change, a camera zoom, loud music, a long pause, cue words, or the co-presence and precedence of a few of these. Semantics are the abstract descriptions of a scene or a multimedia sequence that are natural to a human comprehension and often require context and knowledge to interpret. Examples include a story on the progress of Hurricane Dennis, a home run, the cheering audience in a rock concert, or sunset at a tropical beach.

2.4.1 The syntactic gap and the semantic gap

Neither of the general tasks of syntactic parsing and semantic understanding are solved problems. Smeulders et. al. [109] proposed the notions of “sensory gap” and “semantic gap” under the image retrieval context. There, sensory gap refers to *the gap between the object in the world and the information in a (computational) description derived from a recording of that scene*, and semantic gap refers to *the lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data have for a user in a given situation*.

In multimedia analysis, these definitions are generalized and extended. “Sensory gap” readily generalizes to multiple modalities. In auditory, visual, text and tactile senses, gaps exist in either or both of the following two stages (1) between the physical existence of objects and the recording of this existence and (2) between those recorded by the device and the computation descriptions that we can derive from it.

In the interest of our problems, Smeulders’ notion of “semantic gap” can be refined into two sub-categories. Let “syntactic gap” refer to the gap between the computational description that one can extract from the multimedia data and the interpretation with respect to its syntax. “Semantic gap”, in this narrower sense, would be the the lack of coincidence between an aggregate of the computational descriptions and the syntax, and the human interpretations of the content within the context and knowledge assumed by the analysis system.

These refined notions of the sensor, syntactic and semantic gaps are the gaps that the tasks of feature extraction techniques in [subsection 2.2.2](#), syntax parsing and semantic understanding face, respectively. And in practice the lines between sensory and syntax, syntax and semantics, as well as the lines between their analysis

are not always clean-cut.

In the rest of this section, we shall review examples of prior research on multimedia syntax parsing and semantic recognition, and then briefly discuss the position of our work under this perspective.

2.4.2 Parsing multimedia syntax

Syntactic video analysis processes the feature streams in order to identify syntactical elements, many of which can be unambiguously defined. Some of the syntactic elements can currently be detected with high accuracy (e.g., shot changes [138], text overlays [137], speaker turns [102]), while the reliable characterization of many others are still under active research (e.g. camera motion in a general scene [122]).

Take shot boundary detection, for example. A shot is defined as a continuous camera take in both space and time. The detection algorithms can be either rule-based schemes [139, 14, 138] analyzing the changes in the appearance features such as color, edge, motion and map them to the shot transitions, or learning based approaches [30, 97] that use supervised classification techniques to map feature sequences to shot.

2.4.3 Understanding multimedia semantics

Semantic analysis tries to establish mappings from the descriptors and the syntactical elements to the domain semantics. Though the full interpretation of semantics will necessarily rely on knowledge and context, in many practical cases the meanings in the media are identifiable within broad domain constraints.

Within the domain constraints, semantic multimedia analysis often tries to find a many-to-one mapping from the feature descriptors to the target semantic concept. Given the target concept, there are numerous ways to establish this mapping, such

as automatically generating multiple exemplars by perturbing an initial description from a query [24], or training belief networks [92, 87]. The approaches towards detecting semantics can be generally categorized into rule-based and statistical methods. Take sports video analysis, for example, rule-based detection include mapping edges, lines and motion patterns to corner kicks and shots on the goal [46], or mapping lines and player tracks to tennis rallies and passing shots [115]; statistical event detection include detecting baseball highlights with audio analysis [103], or parsing soccer videos into generic event categories [35, 132]. Note that there are multimedia elements that are both syntactically functional and semantically meaningful, such as video genre [128] and news story boundaries [15, 53].

2.4.4 Discussions

A fully-automatic understanding of the syntax and semantics in multimedia would require a full AI (artificial intelligence) solution. Therefore practical approaches towards semantics will necessarily be data-driven and task-dependent. This shall involve a suitable definition of the domain semantics and their relations (i.e., an ontology), together with good feature extraction techniques and model designs. There are two apparent paths to realizing data-driven recognition systems. One would be to start with a subset of a representative ontology and build a sufficient training corpus, such as the one jointly undertaken by the LSCOM concept ontology for broadcast news [49] and the TRECVID video retrieval benchmark [122]. The other would be to start with few domain assumptions, try to learn the data syntax without explicitly infusing semantics, and assume that the salient syntax would closely resemble semantics under the domain constraints. Examples include text topic detection and tracking (TDT) [121] and the work in this thesis.

Regardless of the path being taken, the goal of multimedia semantics understand-

ing, and hence the interpretation and evaluation of results would at present remain domain-specific, due to the diversity and multiple interpretations of meanings in unconstrained scenarios.

Chapter 3

Mining statistical temporal patterns

This chapter presents models for uni-modal temporal patterns as a basis for the modeling of multimedia patterns. We would like to have a model that can represent self-similar subsegments from long temporal streams, and that can easily generalize to different domains.

In the sections that follow, we first examine different types of patterns found in video in order to clarify our scope, assumptions and limitations. We then present the hierarchical hidden Markov model for unsupervised pattern discovery, along with a strategy that automatically adapts the model complexity to different domains. We will also present an unsupervised feature selection scheme for grouping relevant features into optimal subsets.

3.1 Patterns in video

Video patterns take many different forms. In a single video stream, temporal patterns are *sparse* if they do not cover the entire time axis, such as dialogue scenes in films [116]. They are considered *dense* if the entire video stream can be described as an alternation of different pattern elements with no gaps in time, such as the

state of the game in sports videos [132]. In addition, patterns can be *deterministic* [117] or *stochastic* [131] depending on their appearance, temporal progression, and the description scheme chosen. In multiple video streams, patterns can reside across streams to represent the causal or co-occurrence relationship – for instance, in multi-camera surveillance, an accident in location A will cause traffic jam in location B within time T .

In this chapter, we focus on learning stochastic descriptions of dense patterns. Being stochastic means that the models will try to describe the patterns probabilistically and allow for small deviations from the most-typical cases. Being dense means modeling constituent structures with a common parametric class, and representing their alternation would be sufficient for describing the whole data stream. In other words, we will not need an explicit background class to fill in the “gaps” in between the interesting events.

We now clarify a few assumptions before building a model for video patterns. This is a joint classification and segmentation problem since the locations of the subsequences are unknown a priori, making the sequence clustering algorithms [39, 111] inapplicable. Compared to domains such as biological sequences or web sequences, meaningful patterns or events in video have variable durations, they can start and end at any time, and this rules out models that rely on a known pattern length [8, 72] or those relying on a global system clock [58]. In order to find generic video patterns, we would also like to accommodate video events of arbitrary temporal progression without constraining the allowed transition beforehand, such as those suggested by left-to-right models [29, 86, 8].

3.2 Summary of our approaches

The arguably most widely-used model for event recognition in temporal sequences has been the hidden Markov model (HMM) [99], which models a finite *hidden* state sequence with a Markov assumption, and the observed signal as conditionally independent given the state sequence. The Markov assumption is actually not as limiting as it may seem, since an HMM with a large enough state-space can asymptotically converge to the true distribution of the sequence regardless of the actual memory length. HMMs created via supervised training have been successfully applied to solve many problems such as speech recognition [99], human action recognition [57], video genre recognition [128] or sports event analysis [132] as reviewed in [chapter 2](#).

We extend the hidden Markov model to multiple levels for the unsupervised discovery task. The intuition is to model the recurring events in video as HMMs, and the higher-level transitions between these events as another level of Markov chain. This hierarchy of HMMs forms a hierarchical hidden Markov model (HHMM). Compared to a one-level HMM with the same number of states¹, a HHMM introduces additional transition structure and uses fewer parameters. We have developed efficient methods based on the expectation-maximization (EM) [32] for inference and parameter estimation. This algorithm scales linearly with respect to the sequence length T , it is scalable to events of different complexity, and it is also flexible in that prior domain knowledge can be incorporated in terms of state connectivity, number of levels of Markov chains, and the time scale of each state.

We have also developed algorithms to address model selection and feature selection problems in order for the pattern discovery scheme to generalize to different domains. Bayesian learning techniques are used to learn the model complexity au-

¹Here we compare the case when the number of states in the HMM equal to the total number of bottom-level states in the HHMM.

tomatically, where the search over model space is done with reverse-jump Markov chain Monte Carlo, and the Bayesian Information Criteria (BIC) is used to evaluate the fitness of the model. Moreover, a combined filter-wrapper method is used for feature selection. The wrapper step partitions the feature pool into consistent groups that are relevant with respect to each other according to the mutual information criterion; the filter step eliminates redundant dimensions in each group by finding an approximate Markov blanket; finally the resulting groups are ranked with modified BIC with respect to their fitness. Our approach combines parameter estimation, model and feature selection, sequence labeling, and content segmentation in a unified process. To the best of our knowledge, this is the first work to have addressed all the above issues under the unsupervised scenario.

Evaluation on broadcast video data showed promising results. We tested the algorithm on multiple sports videos, and evaluated our unsupervised approach against the generic high-level structures of the game, namely, *plays* and *breaks* in soccer and baseball. The unsupervised approach achieves comparable or even slightly higher accuracy than our previous results using supervised classification with similar HMM model structures. In addition, the average accuracy of the proposed HHMM with generic topology is significantly better than prior work using unsupervised HMM learning with constrained structures [29, 86]. The feature selection method automatically finds a compact relevant feature set that matches the features manually selected in prior work using domain knowledge. It is encouraging to see that the hierarchical structure based on the HHMM not only provides more modeling power than prior approaches, it has also been effective in discovering patterns without supervision.

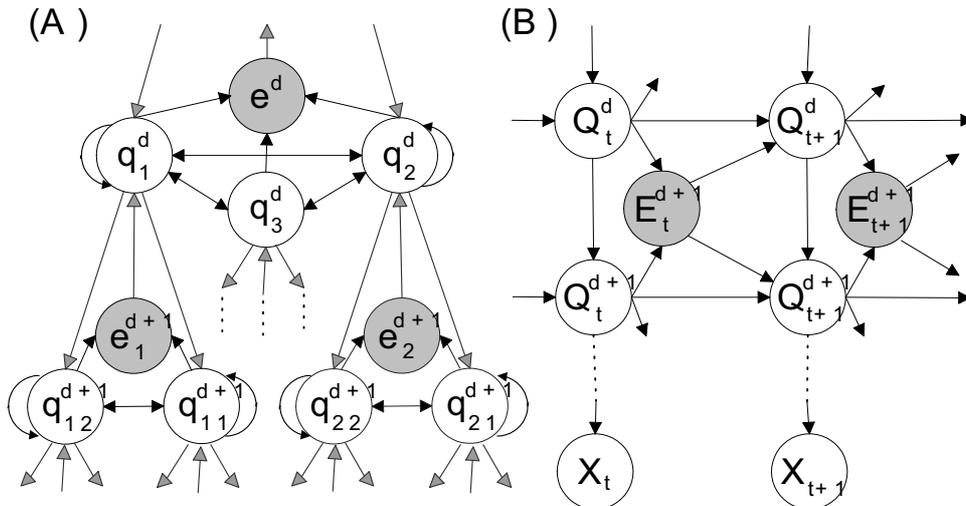


Figure 3.1: Graphical HHMM representation at level d and $d+1$: (A) Tree-structured representation; (B) DBN representations, with observations X_t drawn at the bottom. Uppercase letters denote the states as random variables in time t , lowercase letters denote the state-space of HHMM, i.e., values these random variables can take in any time slice. Shaded nodes are auxiliary *exit* nodes that turn on the transition at a higher level - a state at level d is not allowed to change unless the exiting states in the levels below are *on* ($E^{d+1} = 1$).

3.3 Hierarchical hidden Markov model

We design a multi-layer hierarchical hidden Markov model for structures in video. Intuitively, the higher-level structure elements correspond to semantic events, while the lower-level states represent variations that can occur within the same event. The lower-level states produce the observations, i.e., measurements taken from the raw video, with mixture-of-Gaussian or multinomial distributions. Note that the HHMM model is a special case of Dynamic Bayesian Networks (DBN), also note that the model can be easily extended to more than two levels. In the rest of this section we will discuss algorithms for inference and parameter estimation for a general D -level HHMM.

Hierarchical hidden Markov model was first introduced [41] as a natural gener-

alization of HMM with a hierarchical control structure. As shown in [Figure 3.2\(A\)](#), every higher-level state symbol controls a number of symbols produced by a HMM; a transition at the high level is invoked only when the lower-level model enters an *exit* state (shaded nodes in [Figure 3.2\(A\)](#)); observations are only produced at the lowest level states.

This bottom-up structure is general, and it includes several other hierarchical schemes as special cases. One such special case uses the stacking of left-right HMMs [[29](#), [86](#)] (see [Figure 3.4](#)), where across-level transitions can only happen at the first or the last state of a lower-level model. Another special cases the discrete counterpart of the jump Markov model [[34](#)] with a top-down (rather than bottom-up) control structure, where the level-transition probabilities are identical for each state that belongs to the same *parent* state.

Prior applications of HHMM can be found in three categories: (1) Supervised learning where manually segmented training sequences are available. There, each sub-HMM is learned separately on the segmented sub-sequences, and cross-level transitions are learned using the transition statistics across the subsequences. Examples include exon/intron recognition in DNA strings [[54](#)] or action recognition [[57](#)] from image sequences. (2) Unsupervised learning, where segmented data at any level are not available for training, and parameters of different levels are jointly learned. (3) A mixture of the above, where the state labels at the high level are given (with or without sub-model boundary), yet parameters still need to be estimated across several levels. This can be seen as a combination of (1) and (2). Examples include speech recognition systems that support word-level annotation [[120](#)] and text parsing and handwriting recognition [[41](#)]. No general solutions for (2) are found in the literature. Our work represents a unique approach for learning generic video patterns with no supervision.

3.3.1 Representing an HHMM

For notation convenience we introduce a single index for all multi-level state configurations in the HMM. Denote the maximum state-space size of any sub-HMM as Q , we use the *bar notation* (Equation 3.1) to write the entire configuration of a hierarchical state from the top (level 1) to the d^{th} level with a Q -ary d -digit integer, with the lowest-level states at the least significant digit.

$$q^{(d)} = \overline{(q_1 q_2 \dots q_d)} = \sum_{i=1}^d q_i \cdot Q^{d-i} \quad (3.1)$$

Here $0 \leq q_i \leq Q - 1; i = 1, \dots, d$, and we drop the superscript for q when there is no confusion. Take a two-level HHMM with two top-level states and three sub-states each, for example, the second state in the second model would have $q = 4$.

We represent the whole parameter set Θ of an HHMM as: (1) Emission parameters B that specifies the distribution of observations given the state configuration. i.e., the means μ_q and covariances σ_q when emission distributions are Gaussian. (2) Markov chain parameters λ^d in level d indexed by their parent state configuration $q^{(d-1)}$. λ^d in turn include: (2a) Within-level transition probability matrix A_q^d , where $A_q^d(i, j)$ is the probability of making a transition to sub-state j from sub-state i , and i, j are d^{th} -level state indexes having the same parent state $q^{(d-1)}$. (2b) Prior probability vector π_q^d , i.e., the probability of starting in a child state upon “entering” q . (2c) Exiting probability vector e_q^d , i.e., the probability “exiting” the parent state q from any of its children states. All elements of the HHMM parameters are then written as in Equation 3.2. For notation convenience and without loss of generality we assume that there is only one state at the very top level and thus there is no

need to define the transition probabilities there.

$$\begin{aligned}\Theta &= \left(\bigcup_{d=2}^D \{\lambda^d\}\right) \bigcup \{B\} \\ &= \left(\bigcup_{d=2}^D \bigcup_{i=0}^{Q^{d-1}-1} \{A_i^d, \pi_i^d, e_i^d\}\right) \bigcup \left(\bigcup_{i=0}^{Q^D-1} \{\mu_i, \sigma_i\}\right)\end{aligned}\quad (3.2)$$

3.3.2 Overview of HHMM inference and estimation

The graphical structure of HHMM in [Figure 3.2\(b\)](#) can be factored into a generalized chain structure without loops, thus we implement the model inference using the EM algorithm.

The estimation step evaluates the expectation of complete-data log-likelihood based on the current parameter set Θ , and the estimation step finds a set of new values $\hat{\Theta}$ that maximizes this expectation. Given the observation sequence $x_{1:T}$ we compute the posterior probabilities of the D -level hidden state sequence $q_{1:T}$ using a generalized forward-backward algorithm.

We write the expectation of the complete-data log-likelihood $\Omega(\hat{\Theta}, \Theta)$ in [Equation 3.3](#), iteratively maximizing its expected value leads to the maximization of the data likelihood $L = P(x_{1:T}|\hat{\Theta})$. The generalized chain structure of the HHMM allows a factorization of this expectation into a sum-of-unknowns form, we can then maximize each model parameter separately.

$$\Omega(\hat{\Theta}, \Theta) = E[\log(P(q_{1:T}, x_{1:T}|\hat{\Theta})) \mid x_{1:T}, \Theta] \quad (3.3)$$

$$= L^{-1} \sum_{q_{1:T}} P(q_{1:T}, x_{1:T}|\Theta) \log(P(q_{1:T}, x_{1:T}|\hat{\Theta})) \quad (3.4)$$

The E step and M- step of the algorithms are detailed in [appendix A](#). Note each iteration of this algorithm runs in linear time, or more specifically, $O(DT \cdot Q^{2D})$.

3.4 Model adaptation

Parameter learning for HHMM using EM is known to converge to a local maxima of the data likelihood since EM is a hill-climbing algorithm, and it is also known that searching for a global maxima in the likelihood landscape is intractable. Moreover, this optimization for data likelihood is only carried out over a predefined model structure, and in order to enable the comparison and search over a set of model structures, we will need not only a new optimality criteria, but also an alternative search strategy, since exhausting all model topologies is obviously intractable.

In this work, we adopt randomized search strategies to address the intractability problem on the parameter and model structure space; and the optimality criteria is generalized to incorporate *Bayesian* prior belief on the model structure. Specifically, we use Markov chain Monte Carlo(MCMC) method to maximize the Bayesian information criteria (BIC) [106]. The motivation and basic structure of this algorithm are presented in the following subsections.

We are aware that alternatives for structure learning exist, such as the deterministic parameter trimming algorithm with entropy prior [17], which ensures the monotonic increase of model priors throughout the trimming process. However, we would have to start with a sufficiently large model in order to apply this trimming algorithm, which is undesirable for computational complexity purposes and also impossible if we do not know a bound of the model complexity beforehand.

3.4.1 An overview of MCMC

MCMC is a class of algorithms that can solve high-dimensional optimization problems, and there has been much recent success in using this technique to solve the problem of Bayesian learning of statistical models [7]. In general, MCMC for

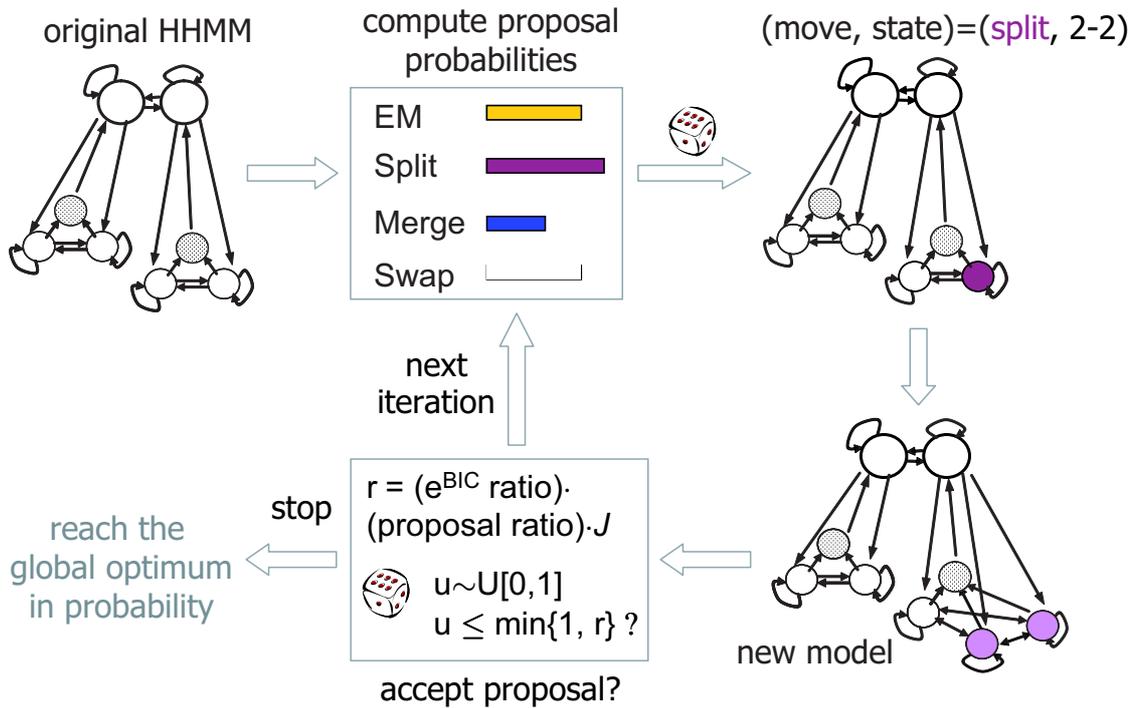


Figure 3.2: Visualization of the MCMC stochastic search strategy for model selection.

Bayesian learning iterates between two steps: (1) The proposal step gives a new model sampled from certain *proposal distributions*, which depends on the current model and statistics of the data; (2) The decision step computes an acceptance probability α based on the *fitness* of the proposed new model using model posterior and proposal strategies, and then this proposal is *accepted* or *rejected* with probability α .

MCMC will converge to the global optimum *in probability* if benign constraints [7] are satisfied for the proposal distributions, yet the speed of convergence largely depends on the *goodness* of the proposals. In addition to parameters learning, model selection can also be addressed in the same framework with reverse-jump MCMC (RJ-MCMC) [47], by constructing reversible moves between parameter spaces of different dimensions. In particular, Andrieu et. al. [6] applied RJ-MCMC to the

learning of radial basis function (RBF) neural networks by introducing birth-death and split-merge moves to the RBF kernels. This is similar to our case of learning a variable number of Gaussians as the emission probabilities by models of different sizes.

In this work, we use EM for model parameter update and MCMC for model structure learning. We choose this hybrid strategy in place of a full Monte Carlo update of the parameter set and the model for efficiency, and the convergence behavior does not seem to suffer in practice.

3.4.2 MCMC for HHMM

Model adaptation for HHMM involves moves similar to [7] for changes in the state space that involve changing the number of Gaussian kernels that associate states in the lowest level with observations. We included four general types of movement in the state-space, as can be illustrated from the tree-structured representation of the HHMM in Figure 3.2(a): (1) *EM*, regular parameter update without changing the state space size. (2) *Split(d)*, to split a state at level d . This is done by randomly partitioning the direct children (when there are more than one) of a state at level d into two sets, assigning one set to its original parent, the other set to a newly generated parent state at level d ; when split happens at the lowest level (i.e. $d = D$), we split the Gaussian kernel of the original observation probabilities by perturbing the mean. (3) *Merge(d)*, to merge two states at level d into one, by collapsing their children into one set and decreasing the number of nodes at level d by one. (4) *Swap(d)*, to swap the parents of two states at level d , whose parent nodes at level $d - 1$ was not originally the same. This specific new move is needed for HHMM, since even if two HHMMs have the same size in the state-space, the exact multi-level structure can still be different. Note we are not including birth/death moves, since

these moves can be reached with a few steps of split/merge.

Model adaptation for HHMMs is carried out as follows:

1. Initialize a HHMM model Θ_0 of given size from data, as described in appendix [A](#).
2. At iteration i , based on the current model Θ_i , compute a probability profile $P_{\Theta_i} = [p_{em}, p_{sp}(1 : D), p_{me}(1 : D), p_{sw}(1 : D)]$ according to equations [\(6.27\)](#)–[\(6.30\)](#), and then propose a move among the types $\{EM, Split(d), Merge(d), Swap(d) | d = 1, \dots, D\}$
3. Update the model structure and the parameter set by appropriate operations on selected states and their children states as described in appendix [B](#), and then perform a few iterations of EM if the state-space has been changed;
4. Evaluate the acceptance ratio r_i for different types of moves according to equations [\(6.37\)](#)–[\(6.36\)](#) in the appendix. This ratio takes into account the model posterior computed with BIC (equation [3.5](#)), and alignment terms that compensate for the fact that the model spaces before and after each move may have unequal sizes. Let the acceptance probability of this move be $\alpha_i = \min\{1, r_i\}$, we then sample $u \sim U(0, 1)$, and we accept this move if $u \leq \alpha_i$, stay to the previous configuration otherwise.
5. Stop if the BIC criteria does not change in a few consecutive iterations, otherwise goto step [2](#).

BIC [\[106\]](#) is a measure of *a posteriori* model fitness, it is the major factor that determines whether or not a proposed move is accepted. Intuitively, BIC trades off data likelihood $P(X|\Theta)$, the model complexity $|\Theta|$ and the amount of data available

$\log(T)$, T being the sequence length, with weighting factor λ . Larger models are penalized by the number of free parameters in the model $|\Theta|$. We empirically choose the weighting factor λ as $1/16$ in our experiments, in order for the change in data likelihood and that in model prior to be numerically comparable over one iteration. We have also observed that the resulting model size is not sensitive to the value of λ . Note that we can also consider λ as a representation of the inherent complexity of the model class, hence information-geometric schemes [85, 4] for choosing λ are also possible.

$$BIC = \log(P(x|\Theta)) - \lambda \cdot \frac{1}{2}|\Theta| \log(T) \quad (3.5)$$

3.5 Feature selection for unsupervised learning

Feature extraction methods for audio-visual streams abound, as a result we are usually left with a large pool of diverse features without knowing which ones are actually relevant to the important events and structures in the data sequences. A few features can be selected manually if adequate domain knowledge exists. Yet very often such knowledge is not available in new domains, or the connection between high-level structures and low-level features is not obvious. In general, the task of feature selection is divided into two aspects - eliminating *irrelevant* features and *redundant* ones. Irrelevant features usually disturb the classifier and degrade classification accuracy, while redundant features add to the computational cost without bringing in new information. Furthermore, for unsupervised structure discovery, different subsets of features may relate to different events, and thus the events should be described with separate models rather than being modeled jointly.

The scope of our problem is to select relevant and compact feature subset that fits the observations well in unsupervised learning over data with dependency under

the HHMM model assumption. There has been prior research in feature selection for supervised learning where features can be either turned “on” or “off” [70], or weighted linearly [69, 13]. For unsupervised learning, Xing and Jordan [133] has extended Koller and Shahami’s algorithm [70] and presented a feature selection scheme over *i.i.d* samples. In this work we address the feature selection problem under the binary (“on-off”) scenario for unsupervised learning over temporal streams, which to the best of our knowledge has not been explored.

3.5.1 The feature selection algorithm

Denote the feature pool as $F = \{f_1, \dots, f_n\}$, the data sequence as $X_F = X_F^{1:T}$, then the feature vector at time t is X_F^t . The feature selection algorithm proceeds goes through two stages, as shown in Figure 3.3.

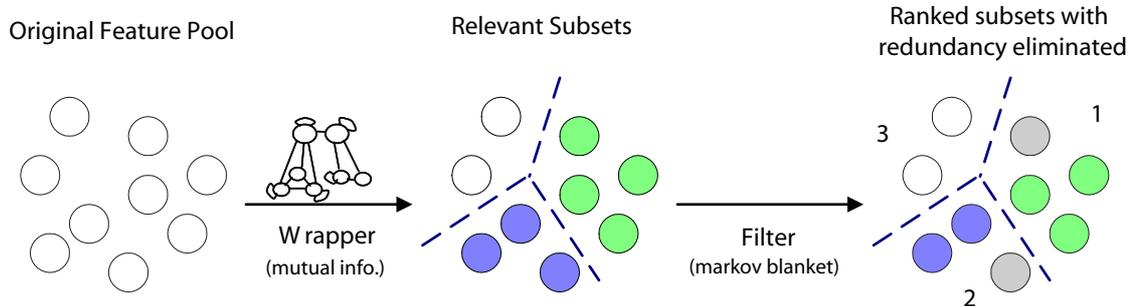


Figure 3.3: Feature selection algorithm overview

The first stage partitions the original feature pool according to their *relative relevance*, and thus grouping mutually relevant features in to the same subset. The feature *relevance* is measured by mutual information (a.k.a. information gain), and the measurement is taken over the maximum-likelihood state sequence rather than the original feature stream, thus avoiding explicitly addressing the correlation in the temporal data stream. This is called a *wrapper* stage as the parameter learning for the model (to obtain the state sequence) is carried out as an inner loop in the

process of partitioning of the feature pool. After the pair-wise mutual information is obtained, we can use any clustering to partition F into N_F subsets. Detailed steps for computing the mutual information is covered in [subsection 3.5.2](#).

The second stage eliminates the redundancy within each *relevant* subset generated in the previous stage. This is done by identifying features that do not contribute to the maximum-likelihood labeling of the sequence, and this is put formally as finding the Markov blanket for the redundant features [70], as detailed in [subsection 3.5.3](#).

After the feature-model combinations are generated automatically, a human operator can look at the structures marked by these models, and then come to a decision on whether a feature-model combination shall be kept based on the meaningfulness of the resulting structures. Alternatively we can use the modified BIC criteria taking into account not only the data likelihood, the model complexity, but also the feature representation, covered in [subsection 3.5.4](#).

3.5.2 Evaluating the information gain

Information gain, or mutual information [31], is one suitable measure to quantify the the degree of *agreement* between different (subsets of) features.

A model Θ_f learned over a feature set f generates a labeling of the original sequence x_f , i.e., the maximum-likelihood hidden state sequence $q_f^{1:T}$. When there are at most Q possible labels, we denote the label sequence as integers $q_f^t \in \{1, \dots, Q\}$. We compute the probability of each label using its empirical portion, i.e. counting the samples that bear label i over time $t = 1, \dots, T$ ([Equation 3.6](#)). Compute similarly the conditional probability of the labels q_f given the partition q_e induced by another feature subset e ([Equation 3.7](#)) by counting over pairs of labels over time t . Then the information gain between feature subsets f and e is computed as the

mutual information between q_f and q_e (Equation 3.8).

$$P_{q_f}(i) = \frac{\#\{t \mid q_f^t = i, t = 1, \dots, T\}}{T}; \quad (3.6)$$

$$P_{q_e|q_f}(i \mid j) = \frac{\#\{t \mid (q_e^t, q_f^t) = (i, j), t = 1, \dots, T\}}{\#\{t \mid q_f^t = j, t = 1, \dots, T\}}; \quad (3.7)$$

$$I(q_f; q_e) = H(P_{q_e}) - \sum_j P_{q_f}(j) \cdot H(P_{q_e|q_f=j}) \quad (3.8)$$

where $i, j = 1, \dots, N$

Here $H(\cdot)$ is the entropy function $H(p) = E(-\log p)$ where $E(\cdot)$ is the expectation function. Intuitively, a larger information gain suggests that the partition q_f closer to q_e , i.e. feature subsets f and e are more *relevant* to each other. After computing the information gain $I(q_f; q_e)$ for each feature pair e and f , we use a clustering algorithm to compute a clustering for the feature pool using I as the similarity matrix, and partition them into N_F subsets. Popular clustering algorithms can include agglomerative clustering using a dendrogram [59], K-means [36], or spectral clustering [89].

Note that if the mutual information were to be directly evaluated on the original feature streams we would have to estimate the joint probability of the entire sequence $P(x_f^{1:T})$, this would quickly become intractable and no reliable estimate can be obtained with a reasonable number of sequences. In this work we measure mutual information on the labeling of the stream instead, thus take into account the feature stream temporal correlation in the HHMM state labeling process, while making reliable estimates. Recent development in machine learning have proposed to directly compute distances between models to define kernels in the model space [61], such approach is also potentially applicable here.

3.5.3 Finding a Markov Blanket

After partitioning the original feature pool with the information gain criteria, we are left with a subset of features with consistency yet possible redundancy. The approach for identifying redundant features naturally relates to the conditional dependencies among the features. For this purpose, we need the notion of a Markov blanket [70].

Definition Let f be a feature subset, M_f be a set of features that does not contain f , we say M_f is the Markov blanket of f , if f is conditionally independent of all variables in $\{F \cup C\} \setminus \{M_f \cup f\}$ given M_f .

Computationally, a feature f is redundant if the partition q of the data set is independent to f given its *Markov Blanket* M_f . In prior work [70, 133], the Markov blanket is identified with the equivalent condition that the posterior probability distribution of the class given the feature set $\{M_f \cup f\}$ should be the same as that conditioned on the Markov blanket M_f only. i.e.,

$$\Delta_f = D(P(q|M_f \cup f) || P(q|M_f)) = 0 \quad (3.9)$$

where $D(P||Q) = \sum_x P(x) \log(P(x)/Q(x))$ is the Kullback-Leibler distance [31] between two probability mass functions $P(\cdot)$ and $Q(\cdot)$.

For unsupervised learning over a temporal stream, however, this criteria cannot be readily employed. This is because (1) The posterior distribution of the labels depends not only on the current observed features but also on adjacent samples. (2) We would have to condition the state label posterior over all dependent feature samples, and such conditioning quickly makes the estimation of the posterior intractable as the number of conditioned samples grows. (3) We will not have enough

data to estimate these high-dimensional distributions by counting over feature-class tuples since the dimensionality is high. We therefore use an alternative necessary condition that the optimum state-sequence $q^{1:T}$ should not change conditioned on observing $M_f \cup f$ or M_f only.

Koller and Sahami [70] have also proved that sequentially removing feature one at a time with its Markov blanket identified will not cause divergence of the resulting set, since if we eliminate feature f and keep its Markov blanket M_f , f remains unnecessary in later stages when more features are eliminated. In addition, as few if any features will have a Markov Blanket of limited size in practice, we sequentially remove features that induces the least change in the state sequence given the change is small enough ($< 2\%$). Note this is a filtering step in our HHMM learning setting, since we do not need to retrain the HHMMs for each candidate feature f and its Markov blanket M_f . Given the HHMM trained over the set $f \cup M_f$, the state sequence q_{M_f} decoded with the observation sequences in M_f only, is compared with the state sequence $q_{f \cup M_f}$ decoded using the whole observation sequence in $f \cup M_f$. If the difference between q_{M_f} and $q_{f \cup M_f}$ is *small enough*, then f is removed since M_f is found to be a Markov blanket of f .

3.5.4 Ranking the feature subsets

Iterating over the procedures in [subsection 3.5.2](#) and [subsection 3.5.3](#) results in disjoint subsets of features $\{F_i, i = 1, \dots, N_F\}$ that are non-redundant and relevant. It will still be desirable to be able to compare the different HHMMs learned over these different subsets, since this would filter out the subsets that may be obviously unfit for the model assumptions in HHMM, and provide a reference for a human operator to look at the pool of models.

Existing criteria for comparing clustering algorithms [38] include scatter separa-

bility and maximum likelihood (ML). Note the former is not suitable to temporal data since multi-dimensional Euclidean distance does not take into account temporal dependency, and it is non-trivial to define another proper distance measure for temporal data; while the latter is also known [38] to be biased against higher-dimensional feature sets. Here we use a normalized BIC criteria (Equation 3.10) to quantify the trade-off between normalized data likelihood \tilde{L} with model complexity $|\Theta|$ and feature dimension. Note the model complexity term is modulated by the total number of samples $\log(T)$; and \tilde{L} for HHMM is computed in the same forward-backward iterations, except all the emission probabilities $P(x|q)$ are replaced with $P'_{x,q} = P(x|q)^{1/n}$, i.e., normalized with respect to data dimension n , under the *naive-Bayes* assumption that features are independent given the hidden states.

$$\widetilde{BIC} = \tilde{L} - \frac{\lambda}{2} |\Theta| \log(T) \quad (3.10)$$

3.6 Experiments and Results

In this section, we report tests of the proposed methods in automatically finding salient events, learning model structures, and identifying informative feature set in soccer and baseball videos.

Sports videos represent an interesting domain for testing the proposed techniques in automatic structure discovery. Two main factors contribute to this match between the video domain and the statistical technique: the distinct set of semantics in one sport domain exhibit strong correlations with audio-visual features and the well-established rules of games and production syntax in sports video programs poses strong temporal transition constraints. For example, in soccer videos, *plays* and *breaks* are recurrent events covering the entire time axis of the video data. In

baseball videos, the different events and their transitions, such as pitching, batting, and running, indicate the semantic state of the game.

Clip Name	Sport	Length	Resolution	Frame rate (f/s)	Source
Korea	Soccer	25'00"	320 × 240	29.97	MPEG-7
Spain	Soccer	15'00"	352 × 288	25	MPEG-7
NY-AZ	Baseball	32'15"	320 × 240	29.97	TV program

Table 3.1: Sports video clips used in the experiment.

All our test videos are in MPEG-1 format, their profiles are listed in [Table 3.1](#). For soccer videos, we have compared with our previous work using supervised methods (scheme 1 below and [132]). The evaluation basis for the structure discovery algorithms are two semantic events *play* and *break*, defined according to the rules of soccer game. These two events are dense since they cover the whole time scale of the video. Distinguishing *break* from *play* will be useful for efficient browsing and summarization, since *break* takes up about 40% of the screen time. Viewers may browse through the game play by play, skipping all the breaks in between, or randomly access the break segments to find player responses or game announcements. For baseball videos, the model was learned without any labeled ground truth or manually identified features *a priori*. A human observer (the author) reports observations on the automatically selected feature sets and the resulting structures afterwards. This is analogous to the actual application of structure discovery to an unknown domain, where evaluation and interpretation of the result is done after automatic discovery algorithms are applied.

It is difficult to define general evaluation criteria for automatic structure discovery results that are applicable across different domains, this is especially true when only domain-specific semantic events are of interest. This difficulty lies in the gap between computational optimization and semantic meanings: the results of

unsupervised learning are optimized with measures of statistical fitness, yet the link from statistical fitness to semantics needs a match between general domain characteristics and the computational assumptions imposed in the model. Despite the difficulty, our results have shown success for constrained domains such as sports. This is encouraging since models built over statistically optimized feature sets have good correspondence with semantic events in the selected domain.

3.6.1 Parameter and structure learning

We first test the automatic model learning algorithms with a fixed feature set manually selected based on heuristics. The selected features, *dominant color ratio* and *motion intensity*, have been found effective in detecting soccer events in our prior works [134, 131]. Such features are uniformly sampled from the video stream every 0.1 second. We compare the performance of the following learning schemes against the ground truth.

1. HMM: (a) with supervised training as developed in our prior work in [132]. One HMM per semantic event (i.e., play and break) is trained on manually labeled segments. For test video data with unknown event boundaries, the data likelihood of each 3-second segment is evaluated with each of the trained HMMs. The final event boundaries are refined with a dynamic programming step taking into account the model likelihoods and the transition likelihoods between every 3-second segments in the training set. (b) unsupervised. Learn a HMM with Q states, with the value of Q taken as the total number of bottom level states automatically learned with model adaptation in scheme 4.
2. Supervised HHMM: Individual HMMs at the bottom level of the hierarchy are learned separately, essentially using the models trained in scheme 1; the level-

existing and top level transition statistics are also obtained from segmented data; then, segmentation is obtained by decoding the Viterbi path from the hierarchical model on the entire video stream.

3. Unsupervised HHMM without model adaptation: An HHMM is initialized with a fixed size of state-space as learned in scheme 4; the EM algorithm is used to learn the model parameters; and segmentation is obtained from the Viterbi path of the final model.
4. Unsupervised HHMM with model adaptation: An HHMM is initialized with arbitrary size of state-space and random parameters; the EM and RJ-MCMC algorithms are used to learn the size and parameters of the model; state sequence is obtained from the converged model with optimal size. Here we will report results separately for (a) model adaptation in the lowest level of HHMM only, and (b) full model adaptation across different levels as described in section 3.4.
5. K-Means clustering.

For supervised schemes 1 and 2, K-means clustering and Gaussian mixture fitting is used to randomly initialize the HMMs. For unsupervised schemes 3 and 4 (as well as all HHMM learning schemes with feature selection in subsection 3.6.2), the initial emission probabilities of the initial bottom-level HMMs are obtained with K-means and Gaussian fitting. We then estimate the one-level transitions probabilities by counting over the MAP Gaussian labels, the multi-level Markov chain parameters are factored from this *flat* transition matrix using a dynamic programming technique that groups the states into different levels by maximizing the number of bottom-level transitions, while minimizing top-level transitions among the Gaussians.

For schemes 1-3, the model size is set to six bottom-level states per event, identical to the optimal model size automatically learned by scheme 4a. We run each algorithm for 15 times with random start and compute the per-sample accuracy against manual labels. The median and semi-interquartile (SIQ) range ² of event detection accuracy across five rounds are listed in Table 3.2.

Learning Scheme	Supervised?	Model type	Adaptation?		Accuracy	
			Bottom-level	High-levels	Median	SIQ ²
(1a)	Y	HMM	N	N	75.5%	1.8%
(1b)	N	HMM	N	N	75.2%	0.8%
(2)	Y	HHMM	N	N	75.0%	2.0%
(3)	N	HHMM	N	N	75.0%	1.2%
(4a)	N	HHMM	N	Y	75.7%	1.1%
(4b)	N	HHMM	Y	Y	75.2%	1.3%
(5)	N	K-means	N	N	64.0%	10%

Table 3.2: Evaluation of learning schemes (1)-(4) against ground truth on clip *Korea*

Results show that the performance of the unsupervised models are comparable to those of the supervised learning, and sometimes it achieved even slightly better accuracy than the supervised learning counterpart. This is quite surprising since the unsupervised learning of HHMMs is not tuned to the particular ground-truth. The results maintain a consistent accuracy, as indicated by the low semi-interquartile range. Also note the comparison basis using supervised learning is actually conservative since (1) unlike prior supervised results [132], the HMMs are learned and evaluated on the same video clip and results reported for schemes 1 and 2 are actually training accuracies; (2) the models without structure adaptation are assigned the *a posteriori* optimal model size that are actually discovered by the unsupervised approach.

²Semi-interquartile as a measure of the spread of the data, is defined as half of the distance between the 75th and 25th percentile, it is more robust to outliers than the standard deviation.

For the HHMM with full model adaptation (scheme 4b), the algorithm converges to two to four high-level states, and the evaluation is done by assigning each resulting cluster to the majority ground-truth label in the cluster. We have observed that the resulting accuracy is still in the same range without knowing how many interesting structures there are to start with. The reason for this performance match lies in the fact that the *additional* high level structures are actually a sub-cluster of *play* or *break*, they are generally of three to five states each, and two sub-clusters correspond to one larger event of play or break (refer to a three-cluster example in [subsection 3.6.2](#)).

3.6.2 With feature selection

On top of the model parameter and structure learning algorithm, we test the performance of the automatic feature selection method ([section 3.5](#)). We use the two test clips, Korea and Spain as profiled in [table 3.1](#). A nine-dimensional feature vector sampled at every 0.1 seconds are taken as the initial feature pool, details for extracting these features are found in [appendix C](#):

Dominant Color Ratio (DCR), Motion Intensity (MI), the least-square estimates of camera translation (Mx, My), and five audio features - Volume, Spectral roll-off (SR), Low-band energy (LE), High-band energy (HE), and Zero-crossing rate (ZCR).

We run the feature selection method with the model learning algorithm on each video stream for five times, with one or two-dimensional feature set as the as initial reference set in each iteration. After eliminating degenerate cases that only consist of one feature in the resulting set, we evaluate the feature-model pair that has the largest *Normalized BIC* value as described in [section 3.5.4](#). In our experiments,

other feature-model pairs are mostly low-level audio features. They may represent alternative meanings in the video, while we do not possess relevant ground truth to evaluate them, we have observed that the other labels has a rate of change much higher than the highest-ranking feature subsets, and hence would not represent to play-break.

For clip *Spain*, the selected feature set is {DCR, Volume} The model converges to two high-level states in the HHMM, each with five lower-level children states. Evaluation against the *play/break* labels shows a 74.8% accuracy. For clip *Korea*, the final selected feature set is {DCR, Mx}, with three high-level states and {7, 3, 4} children states respectively. If we assign each of the three clusters to the semantic event with a majority rule (which would be {*play*, *break*, *break*} respectively), per-sample accuracy would be 74.5%. The automatic selection of DCR and Mx as the most relevant features actually confirm the two features DCR and MI, manually chosen in our prior work [132]. Mx is a feature that approximates the horizontal camera panning motion, the most dominant factor contributing to the overall motion intensity (MI) in soccer videos. Horizontal panning is also the most popular camera movements used to track the ball in a soccer field, this motion cue is intuitively useful for following the overall game status [134].

The accuracies are comparable to their counterpart (scheme 4) in section 3.6.1 without varying the feature set (75%). Yet the small discrepancy may due to (1) Variability in RJ-MCMC (section 3.4), for which convergence diagnostic is still an active area of research [7]; (2) Possible inherent bias may exist in the normalized BIC criteria (equation 3.10) where we have used the same weighting factor λ for models of different state-space size or different parameters. While it did not take into account the inherent *complexity* of the parametric class [85], the discrepancy is empirically small and did not affect our results significantly.

3.6.3 Testing on a different domain

We have also conducted a preliminary study on the baseball video clip described in table 3.1. The same 9-dimensional features pool as in section 3.6.2 are extracted from the stream also at 0.1 second per sample. The learning of models is carried out without having labeled ground truth or manually identified features *a priori*. Observations are made based on the selected feature sets and the resulting structures of the test results. This is a standard process of applying structure discovery to an unknown domain, where automatic algorithms serve as a pre-filtering step, and evaluation and interpretation of the result can only be done afterwards.

HHMM learning with full model adaptation and feature selection is conducted, resulting in three consistent compact feature groups: (a) HE, SR, ZCR; (b) DCR, MX; (c) Volume, LE. It is interesting to see audio features falls into two separate groups, and the visual features are also in a individual group.

The BIC score for the second group, dominant color ratio and horizontal camera pan, is significantly higher than that of the other two. The HHMM model in (b) has two higher-level states, each has six and seven children states at the bottom level, respectively. Moreover, the resulting segments from the model learned with this feature set correspond to interesting semantic events, with one cluster of segments mostly corresponding to pitching shots and other field shots when the game is in play, while the other cluster contains most of the cutaways shots, score boards and game breaks, respectively. Evaluation against *play* and *break* in baseball showed an accuracy of 82.3%. It is not surprising that this result agrees with the intuition that the status of a game can mainly be efficiently inferred from visual information.

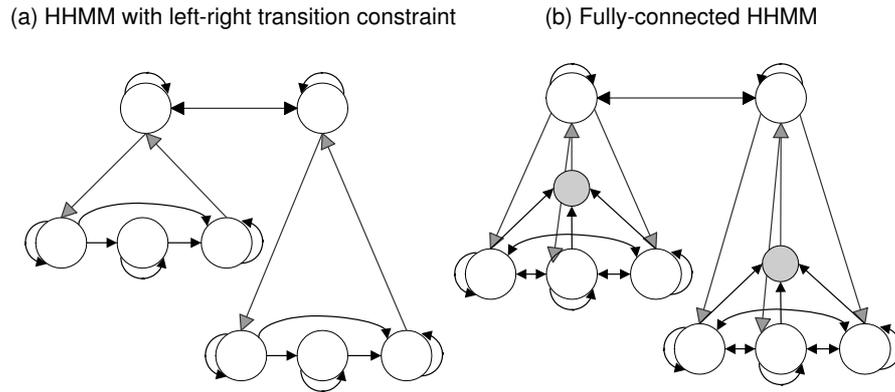


Figure 3.4: Comparison with HHMM with left-to-right transition constraints. Only 3 bottom-level states are drawn for the readability of this graph, models with 6-state sub-HMMs are simulated in the experiments.

3.6.4 Comparing to HHMM with simplifying constraints

In order to investigate the *expressiveness* of the multi-level model structure, we compare unsupervised structure discovery performances of the HHMM with a similar model with constraints in the transitions each node can make [29, 86].

The two model topologies being simulated are visualized in [Figure 3.4](#):

- (a) The simplified HHMM where each bottom-level sub-HMM is a left-to-right model with skips, and cross level entering/exiting can only happen at the first/last node, respectively. Note the right-most states serving as the single exit point from the bottom level eliminates the need for a special *exiting* state.
- (b) The fully connected general 2-level HHMM model used in scheme 3, section 3.6.1, a special case of the HHMM in [Figure 3.2](#)). Note the dummy *exiting* cannot be omitted in this case.

Topology (a) is of interest because the left-to-right and single entry/exit point constraints allows the use of learning algorithms for regular HMMs by *collapsing* this model and setting hard constraints (zeros) on the transition probability. The

collapsing can be done because unlike the general HHMM, there is no ambiguity in whether or not a cross-level has happened in the original model given the last state and the current state in the collapsed model. Equivalently, the *flattened* HMM transition matrix can be uniquely factored back to recover the multi-level transition structure. Note the trade-off here for model generality is that parameter estimation of the collapsed HMMs is of complexity $O(T|Q|^{2D})$, while HHMMs will need $O(DT|Q|^{2D})$, as analyzed in section A.6. With the total number of levels D typically a fixed small constant, this difference is not significant and does not prevent the application of HHMM to long sequences.

Topology (a) also contains models in two prior work as special cases: [29] uses a left-to-right model without skip, and single entry/exit states; [86] uses a left-to-right model without skip, single entry/exit states with one single high-level state, i.e. the probability of going to each sub-HMM is independent of which sub-HMM the model just came from, thus eliminating one more parameter than [29]. Both of the prior cases are learned with standard HMM learning algorithms.

Both models in Figure 3.4 are tested and compared on the soccer video clip *Korea*. It performs parameter estimation with a fixed model structure of six states at the bottom level and two states at the top level, over the pre-defined features set of DCR and MI (subsection 3.6.1). Results obtained over 5 runs of both algorithms showed that the average accuracy of the constrained model (Figure 3.4(a)) is 2.3% lower than that of the fully connected model (Figure 3.4(b)).

This result demonstrates that adopting a fully connected model with multi-level control structures indeed brings in extra modeling power for the chosen domain of soccer videos.

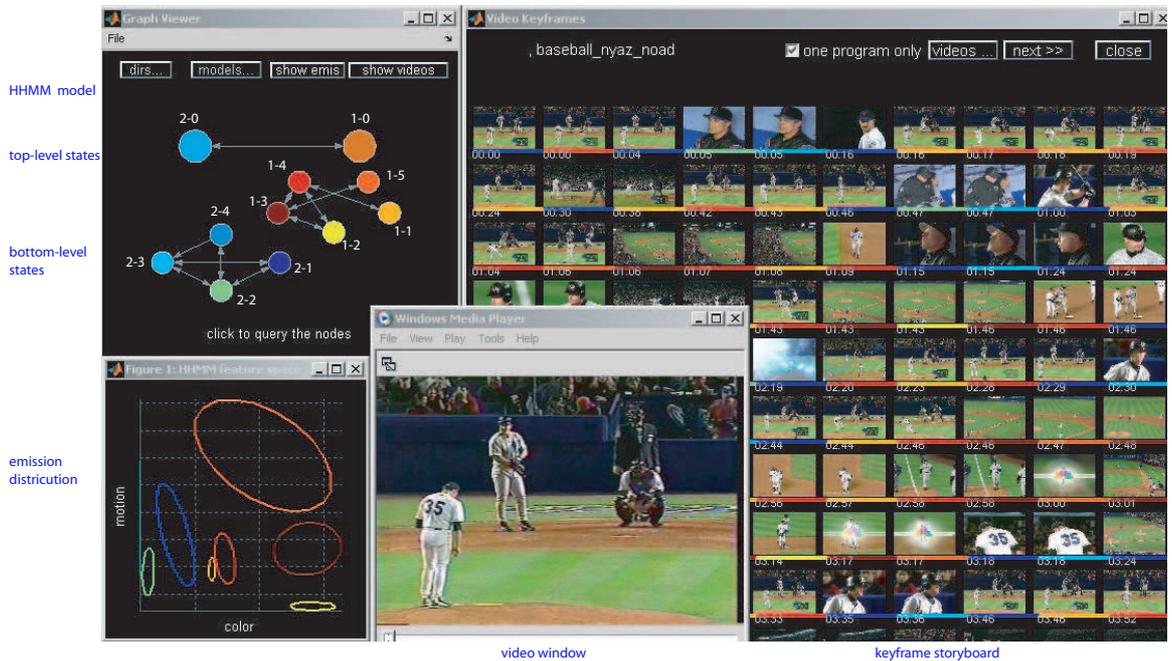


Figure 3.5: HHMM model visualization with feature distributions and video storyboard.

3.7 Visualizing and interpreting multimedia patterns

The HHMM models, once learned, can be used not only to label the videos and as a navigation aid for exploring and revealing the structure in the videos.

Figure 3.5 shows an example visualization interface for video navigation using the model. The HHMM state-space is visualized as a two-level tree structure on the top left panel; the feature emission probabilities associated with the bottom-level states are shown on the bottom-left panel; frames from the video are shown on the right. The emission probabilities and video keyframes are color-coded according to their state labels. The nodes in the HHMM model, the feature distributions, and the corresponding video segments are hyperlinked in the data structure so that we can query a state or a transition in the model to browse the key frames associated with the state label, we can also query a keyframe to browse and play back the

actual video segment.

From the baseball video in [Figure 3.5](#) for example, we noticed that state 1-1 has a compact distribution in the feature space (medium value in grass color percentage, low value in global motion) and is only connected to state 1-4 through temporal transition. When mapped into the video keyframes and segments, we can see that state 1-1 corresponds to preparation of the pitchings with high accuracy, and the subsequent state 1-4 labels correspond to the actual pitching that are of higher motion and similar color layout.

In this example, HHMM (or similar generative models) not only serves as a clustering, browsing and indexing engine, but also provide an intuitive explanation of the data. Such unsupervised approach to finding patterns embedded in video sequences is very useful in defining domain-specific events, which can then be used to guide the development of supervised solutions for event detection.

3.8 Chapter summary

In this chapter, we presented the problem of unsupervised pattern discovery in temporal sequences. We used hierarchical hidden Markov model for unsupervised learning of patterns in videos. We have devised a strategy to automatically adapt the model complexity to different domains, as well as an unsupervised feature selection scheme for grouping relevant features into optimal subsets. Our approach has shown comparable performance with its supervised counterparts on various sport videos, we have also shown an visualization interface that visualizes the learned models and explains the patterns discovered.

Chapter 4

Assessing the meanings of audio-visual patterns

In this chapter we look at the problem of automatically finding the meanings of the discovered audio-visual patterns. This problem of interpreting and accessing the quality of the results arises in many unsupervised discovery tasks, yet the perspective of having meaningful patterns is specific to multimedia. This is because the consumption of media content by humans is leading not to statistics or abstract relationships but to meanings and semantics.

In the sections that follow we first look at the need for finding meanings in multimedia, we then present algorithms for the statistical association of tokens from different streams, we examine the experiment results on news video corpus, and we finally discuss possible extensions and the relations with other work that associates image/video with textual terms.

4.1 The need for meanings in multimedia

Unsupervised learning algorithms use the internal structure of the data for grouping, consequently the outcome lacks the explicit interpretations that the training labels provide in a supervised learning scenario. Thus interpreting the meanings of the

resulting clusters and assessing their quality become unavoidable sub-problems for unsupervised clustering. In nicely constrained domains, this assessment can usually be addressed with expert knowledge or domain assumptions. For the sports videos discussed in the previous chapter, for example, we found the best association from the HHMM labels to the handful of meanings in the sport by searching through all possible correspondences between patterns and domain-specific event classes. For other data analysis tasks such as mining the shopping basket [1] or the web-logs [58] the resulting patterns are examined by human experts to confirm that they are “meaningful” with respect to the particular domain.

For the task of multimedia data mining in general, however, evaluation with expert knowledge is not always possible, because: (1) The number of interesting patterns may be unknown and large, making the search over all permutations impractical. (2) The set of interesting patterns in the domain may change over time, and the emerging new meanings and the vanishing old meanings make any assessment to quickly become obsolete, resulting in a constant demand for refreshing the expert knowledge. One such domain is news videos. There the represented meanings are so diverse and dynamic that coming up with a stable and precise news event ontology is a challenging task in itself.

To our advantage, audio-visual streams in many domains comes with metadata, such as the closed caption in TV broadcasts or the descriptions surrounding an image on a web page. These metadata, especially the surrounding textual descriptions, provide potential information to annotate and explain the audio-visual content. On the other hand, the audio-visual content enriches the users’ experiences, and serves to illustrates the text descriptions. While textual data themselves do not equate to semantics, a text summary in a few words is more succinct in representation (than the raw audio and images) and easy for a person to grasp.

In this chapter, we explore the statistical association of temporal audio-visual patterns with text using co-occurrence analysis and machine translation techniques. We use news videos as the test domain and find promising associations from the video patterns to distinct topics such as *el-nino* or *politics*, we have also demonstrated the advantage of using a dynamic structure model over a plain clustering alternative.

4.2 Cross-modal association in temporal streams

In this section we describe models for establishing statistical multimodal association. This association will enable the selection of words for annotating a given audio-visual segment or vice versa. Our algorithm follows three steps: we start from a sensible division of the multimodal stream in time; we then pre-process both the audio-visual stream and text stream and discretize them into “tokens” in either stream; finally we associate the tokens in the same temporal segment with co-occurrence analysis and further refine the co-occurrence statistic with machine translation models.

4.2.1 The temporal division of the multimodal streams

News videos are in the form of time-stamped streams that are not synchronous across the modalities. In the visual channel, the signal is typically sampled at 30 frames per second. There, a natural temporal syntactical unit is a shot, defined as a continuous camera take in space and time, which typically lasts from just a few seconds to tens of seconds. In the audio channel, the signal is sampled at several thousand times per second, where the spoken content typically has a few syllables per second, and a continuous background ambience lasts a few seconds to tens of seconds. In the text stream, the closed caption comes at roughly the same rate as

the spoken content, i.e. we can expect to see one to four words each second, and each phrase or each sentence will have a few to a few dozen words.

In order to establish correspondence across multiple modalities, a common temporal division would help us computing the correspondence with enough information and less noise by restricting them into a reasonable temporal range. Note under the above cross-modal asynchrony there are no common natural boundaries in the multimodal signals. And if we choose the consistent unit in one modality, such as a shot, a word, or a sentence, as the reference for the other modalities, it would imply insufficient statistics or redundancy for others.

To this end, we can make use of the higher level semantic units defined for many domains of multimedia, such as scenes in film or stories in news. A news story is defined as “a segment of a news broadcast with a coherent news focus which contains at least two independent, declarative clauses” [122]. In news broadcasts, story boundaries do not necessarily coincide with shot boundaries or sentences endings [53], however they represent a common meaningful temporal division in news, and the transition of semantic topics happens at these boundaries. In this work, we try to establish correspondence between the audio-visual channel and the text within each story. Our evaluations show that co-occurrence statistics on stories yields word precisions about ten times that of the shots while producing comparable recalls.

4.2.2 Tokenizing each modality

We need to *tokenize* each of the audio-visual stream and the text inputs before starting to estimate their correspondence. This process is needed because (1) Some of the observations can be *continuous* both in value and in time (e.g. color, motion, or audio volume values), while our perception of these stream are discrete events (e.g. an anchor, a scream, or an explosion). The building blocks in each stream

shall resemble our perception for the correspondence to be more meaningful. (2) For temporally correlated noisy streams the *tokenization* process also serves to de-correlate and denoise the raw observations, such as the use of HHMM modeling and shallow parsing operations described below.

The tokenization of multi-dimensional continuous or discrete observations can be easily done with clustering or vector quantization algorithms such as K-means or Gaussian mixture modeling [59]. In temporally correlated streams such as video, it is also desirable to take the temporal dimension into account, and this can be achieved with dynamic models such as the HMM or the HHMM as described in the previous chapter. This tokenization process can be guided by two factors: (1) To minimize the distortion or maximize the data likelihood, as done in typical quantization or clustering algorithms; (2) To identify algorithms that are more suitable for the domain, and that reveals meaningful clusters, which is done with domain knowledge and via evaluations with other modalities, such as association with semantic tags. In this chapter we experiment with HHMM and K-means. The algorithm of using HHMM to label the sequence was presented in [chapter 3](#). Details for the specific features for news videos and the conversion from features to tokens are covered in [section 4.3](#).

The text streams already come in the discrete form of time-stamped words. The discourse style of a news program is usually concise and direct, hence it suffices to stay at the level of individual words rather than going to higher-level concepts via linguistic or semantic analysis, to represent the topic of discussion. We choose to focus on a lexicon of frequent and meaningful words, freeing ourselves from the noise introduced by stop words and statistically insignificant ones. The lexicon is obtained from the corpus after a few shallow parsing operations: (1) Stem the words from a speech-recognizer output or the closed caption with an off-the-shelf stemming

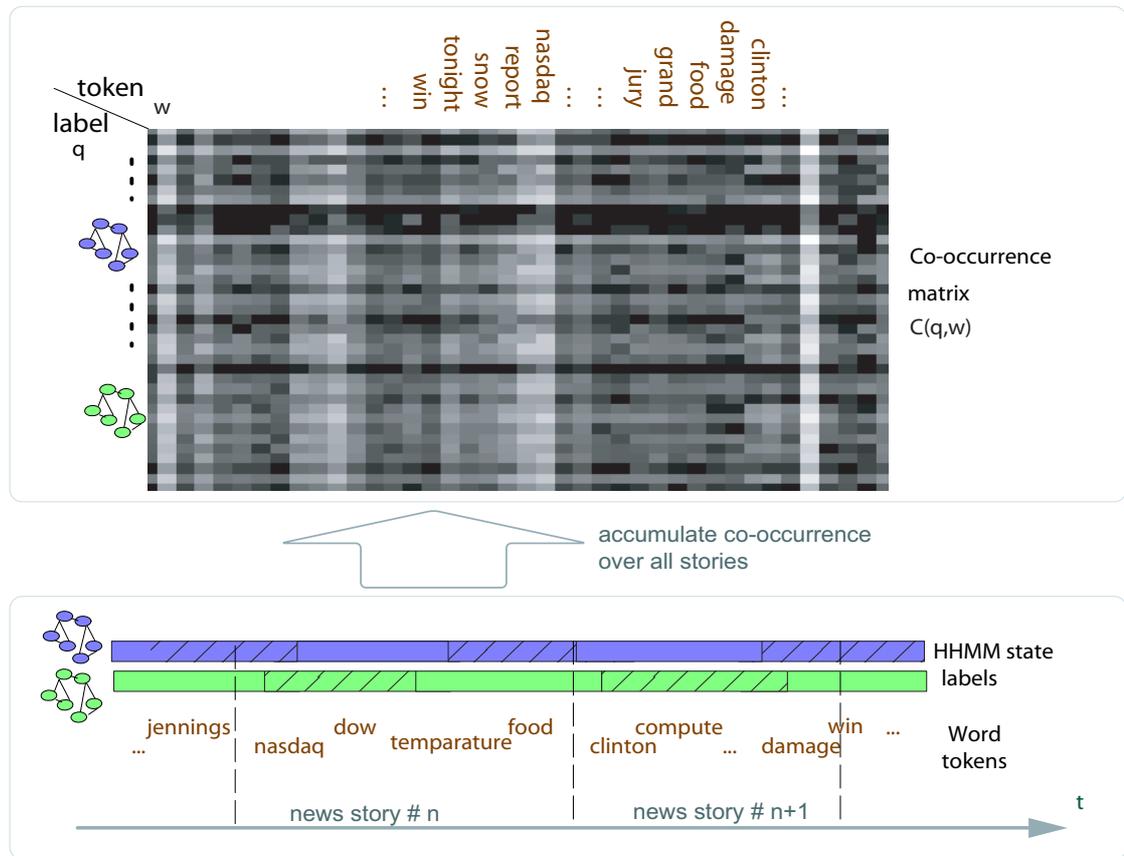


Figure 4.1: Generating co-occurrence statistics from the HHMM *labels* and word *tokens*. The grayscale in the co-occurrence matrix indicate the magnitude of the co-occurrence counts, the brighter the cell, the more instances of word w and label q that were seen in the same story.

algorithm such as the Porter Stemmer [96]; (2) Prune stop words taken from popular stoplists [124], and domain-specific stop words such as “news” or “today”; (3) Prune the rare word stems that appear no more than a few times.

4.2.3 Co-occurrence analysis

One simple and widely-used method for establishing statistical association is via co-occurrence analysis. This method has been popular in many other problem domains such as mining item correlations in shopping baskets [1]. As illustrated in Fig. 4.1,

we obtain the co-occurrence statistic $C(q, w)$ for a HHMM state *label* q and a word *token* w by counting the number of times that the state label q and the word w both appear in the same temporal segment, and this statistic is accumulated across all video clips.

Denote the index for all video clips in the corpus as $k = 1, \dots, K$, we partition each video into non-overlapping temporal intervals (news stories in this chapter, from automatic segmentation [53]) $\mathcal{S}_k = \{s_j^{(k)}, j = 1, \dots, L_k\}$, where j indexes the stories and L_k is the number of stories in video k . Denote the stream of audio-visual labels, i.e. the maximum-likelihood state sequence of HHMM or the cluster labels, as $q_u^{(k)}$, where u is the time index for the tokens in the audio-visual channel, and q takes values from the label set \mathcal{Q} . Denote the word tokens in video k as $w_v^{(k)}$, where v is the time stamp for the words, and w takes its value from the lexicon \mathcal{W} . The co-occurrence statistic $C(q, w)$ is simply calculated as the counts of simultaneously observing symbols q and w in the same temporal interval

$$C(q, w) = \sum_{k=1}^K \sum_{j, u, v} I\{q_u^{(k)} = q, w_v^{(k)} = w, u \in s_j^{(k)}, v \in s_j^{(k)}\} \quad (4.1)$$

$$\forall q \in \mathcal{Q}, w \in \mathcal{W}$$

Here $I(\cdot)$ is the indicator function, and the notation $u \in s_j^{(k)}$ and $v \in s_j^{(k)}$ means that the temporal index u and v are contained in the story interval $s_j^{(k)}$.

Once the co-occurrence statistics are in place, we normalize the co-occurrence counts to obtain empirical estimates for the conditional probabilities of seeing the words given labels and vice versa, denoted by lower-case symbol $c(q|w)$ and $c(w|q)$, as shown in [Equation 4.2](#) below. These quantities can serve as a basis for predicting words that annotate an audio-visual segment with label q , or for retrieving best

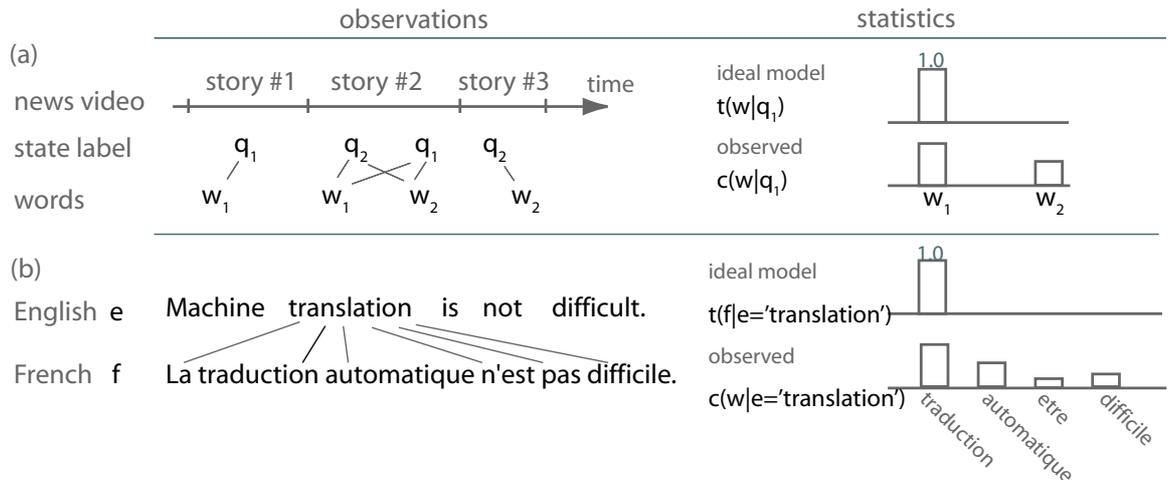


Figure 4.2: The analogy between (a) metadata association in multimedia streams and (b) machine translation in language. $t(\cdot|\cdot)$ denote the probability under the ideal underlying model; $c(\cdot|\cdot)$ denote the observed co-occurrence statistic using imprecise alignment.

video shots to illustrate a certain word w .

$$c(w|q) = \frac{C(q, w)}{\sum_w C(q, w)}, \quad c(q|w) = \frac{C(q, w)}{\sum_q C(q, w)} \quad (4.2)$$

Note that the co-occurrence counts necessarily ignores the temporal orders that the tokens appear in either stream. This is assumption greatly simplifies the model, and it may not be as restricting as it seems, given the nature of the multimedia streams: there is no apparent order (or grammar) that the audio-visual narrative elements observe in order to illustrate a story, as a story can start with anchor shots but is also very likely to start in a middle of an anchor or no anchor at all [53]. While more complex co-occurrence models for languages exist [21], we choose the simple co-occurrence since the *grammars* in multi-modal streams, if exist, are weaker and more diverse.

4.2.4 Machine translation

In co-occurrence analysis, every label receives one increment from every word in the same story. As illustrated in [Figure 4.2\(a\)](#), the co-occurrence matrix from this calculation is a “smoothed” version of the ideal associations, because words and labels of different meanings can appear in the same news story, and thus receiving extra counts from the temporal co-location. It will be desirable to “un-smooth” the co-occurrence counts and recover a clean version of the association.

This problem has been addressed in the context of machine translation (MT) [\[21\]](#). As illustrated in [Figure 4.2\(b\)](#), statistical machine translation tries to learn from data the word-level probabilities of French words f given an English word e . The corpus is sentence-level aligned bi-text, no word-level alignment information is available. Brown and colleagues proposed a unigram model (referred to as *Model 1* in the original text) for the sentence generation, as illustrated in [Figure 4.3](#).

The unigram translation model consist of conditional probabilities $t(f|e)$ of the French words $f \in \mathcal{F}$ given the English words $e \in \mathcal{E}$. Given an English sentence (i.e. a collection of words $\{e_1, e_2, \dots, e_{N_e}\}$), a French sentence can be *generated* by independently drawing $f_i \sim t(f|e_i)$, $i = 1, \dots, N_e$. Given the two sentences in each language, a *complete* likelihood would involve the prior unigram probabilities $p(e)$, “translation” probabilities $t(f|e)$ and the *alignment* information as for which English word generates the observed French word f . Since the alignment is hidden in general, we can obtain a partial data likelihood by marginalizing all possible *hidden alignments* between the words in each language. It is easy to see that under the simple unigram assumption the marginalization is separable for each position in the sentence, and the sufficient statistics for this model only contain the co-occurrence counts $C(f, e)$.

a French sentence of N_f words

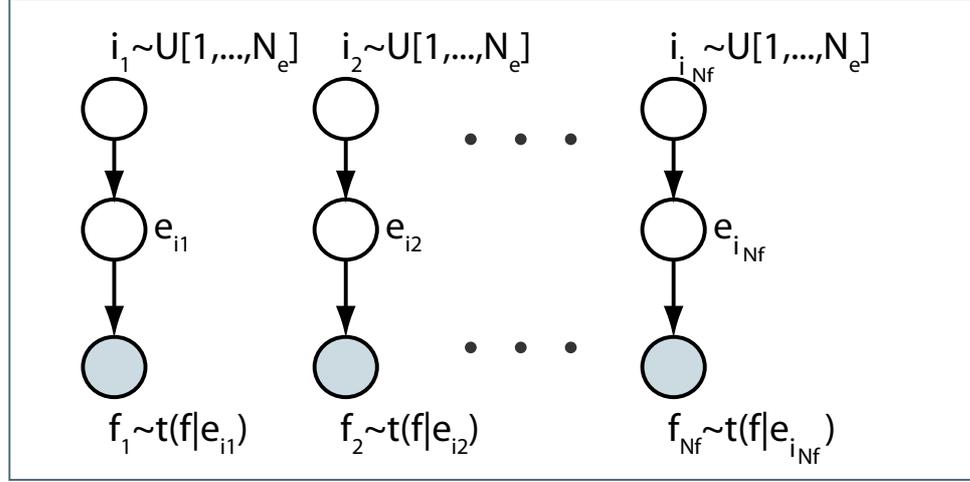


Figure 4.3: The *generation* of a sentence of N_f French words from N_e English words according to the unigram machine translation model $t(f|e)$, i.e., *Model 1* by Brown et. al. [21]. The clear nodes (the correspondences) are hidden; the shaded nodes (the French words) are observed. $U[1, \dots, N]$ denotes a uniform distribution on $\{1, \dots, N\}$

We follow the inference of the unigram model, and use it to estimate the conditional probability for both the words given the labels, and vice versa. The E-step for the posteriors is easily derived from the current model:

$$\bar{t}(q|w) = \frac{t(w|q)}{\sum_q t(w|q)}, \quad \bar{t}(w|q) = \frac{t(q|w)}{\sum_w t(q|w)} \quad (4.3)$$

The M-step computes a new estimate of the conditionals based on both the posteriors and the observed co-occurrence counts:

$$t(w|q) \leftarrow \frac{C(q, w)\bar{t}(q|w)}{\sum_w C(q, w)\bar{t}(q|w)}, \quad t(q|w) \leftarrow \frac{C(q, w)\bar{t}(w|q)}{\sum_q C(q, w)\bar{t}(w|q)} \quad (4.4)$$

Each EM iteration above gives more weight to the larger co-occurrence counts. Intuitively, this becomes a “sharpening” process that suppresses the smaller counts

that come from the co-occurring but unrelated words, as will be seen in the next section.

4.3 Experiments

In this section, we discuss the results of predicting the correspondence using the co-occurrence statistic and the probabilities refined by MT on TRECVID news video data-sets [122].

4.3.1 Visualizations of co-occurrence and machine translation

We first report a few observations on the TRECVID 2003 corpus, which consist of 44 half-hour programs from ABC World News and CNN Headline News. Each video comes with the audio-visual stream, the ASR words, and the ground-truth for story boundaries that came from the text transcript [121]. We divide the data into four sets each having 11 programs from the same channel. We rotate the roles of these sets as the training set, from which the HHMM models and the correspondences are *learned* (without additional supervision), and the test set where the models are used to *predict* words in the new videos.

On the audio-visual channel, we use automatic visual concept detectors scores on each shot as the features for learning the HHMM state labels. The concepts used are $\{weather, people, sports, non-studio, nature-vegetation, outdoors, news-subject-face, female speech, airplane, vehicle, building, road\}$, selected from the 16 Trecvid2003-evaluated concepts that have a reported average precision greater than 50%. The concept detection outputs [5] are the fusion results from multiple SVM classifiers on image features. These concept fusion scores are obtained via various strategies (min., max, linear combination, etc.), and we normalize the scores to between 0 and

1 and uniformly quantize them into three levels. These mid-level detector results are preferred in place of low-level audio-visual concepts, because the news videos have two notable differences compared to the sports programs in [chapter 3](#): (1) The settings in news programs are much more diverse than sports, and the camera view does not directly relate to the color, texture, and motion cues. (2) The syntax in news videos is mostly controlled by the change of cameras (and hence the location and the objects it may contain), not by the temporal trend in low-level cues such as zoom and camera pan.

On each of the 11-video sets (the training set), we learn $M = 10$ different HHMM models using different subsets of the 12 concepts. These subsets are generated with hierarchical agglomerative clustering on the mutual information metric ([section 3.5](#)); the number of models is set to traverse into considerable depth into the clusters; the HHMM models on this dataset typically has $5 \sim 10$ distinct bottom-level states, as determined by the model selection algorithm in [section 3.4](#), the ten HHMMs have 59 states in total. For comparison, we also use the same features subsets and (the automatically selected) number of clusters to learn 10 clusterings via K-means.

The correspondence of the state labels in all models to a 155-word-stem lexicon in the ASR transcript (in the training set) is then estimated according to Equations [\(4.1-4.4\)](#) to produce conditional confidence values $c(w|q)$, $c(q|w)$ and $t(w|q)$, $t(q|w)$, respectively. These probabilities can be interpreted in two complementary contexts. One is *auto-annotation*, i.e., predicting words upon seeing an HHMM label, $c(w|q)$ is the *precision* value of this token-prediction process on the testing set by the counting processing in Equation [\(4.1\)](#); the other is *retrieval*, i.e., producing possible labels upon seeing a word, and $c(q|w)$ is *recall* of this label-retrieval process. It is easy to see, however, that the precision is biased towards frequent words or infrequent labels, while the recall tends to be large with infrequent words or frequent labels.

To compensate for the prior bias in the co-occurrence matrix, we examine an alternative measure, the *likelihood ratio* L , computed as the ratio of the conditional to the prior probabilities. Equation 4.5-4.6 computes in ratio for the statistic before and after machine translation, where the prior probabilities are estimated as $p(q) = \sum_w c(q, w)$ and $p(w) = \sum_q c(q, w)$. Intuitively $L(q, w)$ takes value from 0 to $+\infty$, where a value of 1 means that the two events $q = q$ and $w = w$ are independent, a large L implies strong association, and a small L implies strong exclusion.

$$L_w^c(q, w) = \frac{c(w|q)}{p(w)}, \quad L_q^c(q, w) = \frac{c(q|w)}{p(q)}; \quad (4.5)$$

$$L_w^t(q, w) = \frac{t(w|q)}{p(w)}, \quad L_q^t(q, w) = \frac{t(q|w)}{p(q)}; \quad (4.6)$$

Note that for the co-occurrence statistic the *likelihood ratio* on either variable is the joint probability normalized by the product of the two marginals, i.e.,

$$L_w^c(q, w) = L_q^c(q, w) = \frac{c(q, w)}{\sum_q c(q, w) \cdot \sum_w c(q, w)}.$$

Figure 4.4 visualizes the likelihood ratio before and after machine translation. We can see that the EM algorithm “un-smoothes” the raw co-occurrence counts as expected, resulting in more bright (strong word-label associations) or dark (exclusions) spots in the association matrix.

Figure 4.5 visualizes the likelihood ratio of co-occurrence counts for the audio-visual labels generated by HHMM and K-means. We can see that operating on the same feature sets and the same model sizes, there are much more strong associations (bright peaks) and exclusions (dark valleys) in the labels obtained with HHMM than that of the K-means, and this shows that temporal modeling is indeed more capable of discovering patterns that correlate to the semantic tags for the news domain.

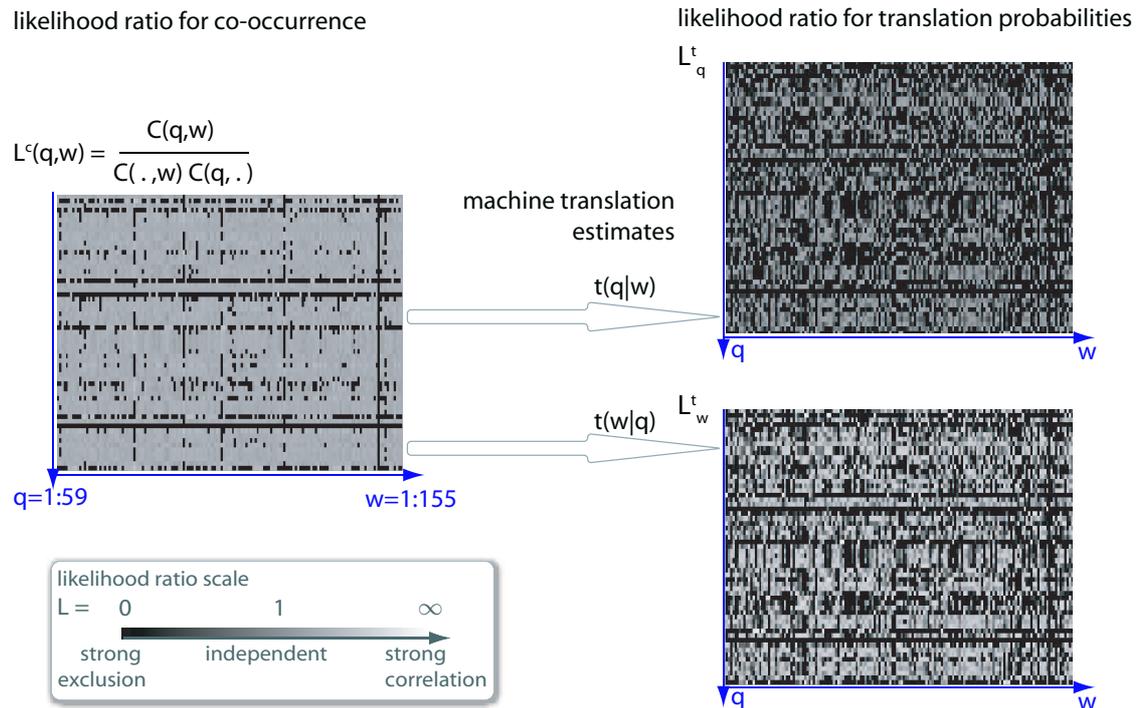


Figure 4.4: Word-association with likelihood ratio before and after machine translation. The likelihood ratio function L is rendered in log-scale.

4.3.2 Word prediction and correlation with topics

We now examine the models and associations in the previous subsection in detail and see how they can be interpreted in terms of the meanings in the news. We sort all the $(label, token)$ pairs based on L^c , L_w^t and L_q^t , respectively, we then examine the *salient* pairs that lie in the top 5% of each L values.

One interesting label as shown in [Table 4.1](#) is $(m, q) = (6, 3)$ (the third state in the sixth HHMM model), indicating high confidence in both of its raw concepts $\{people, non-studio-setting\}$. With the list of predicted words as $\{storm, rain, forecast, flood, coast, el, nino, administr, water, cost, weather, protect, starr, north, plane, northern, attorney, california, defens, feder, gulf\}$, it clearly indicates the topic *weather*. In fact, it covers the news stories on el-nino and storms that prevailed the United States in the spring of 1998 with 80% and 78% recall on the training and

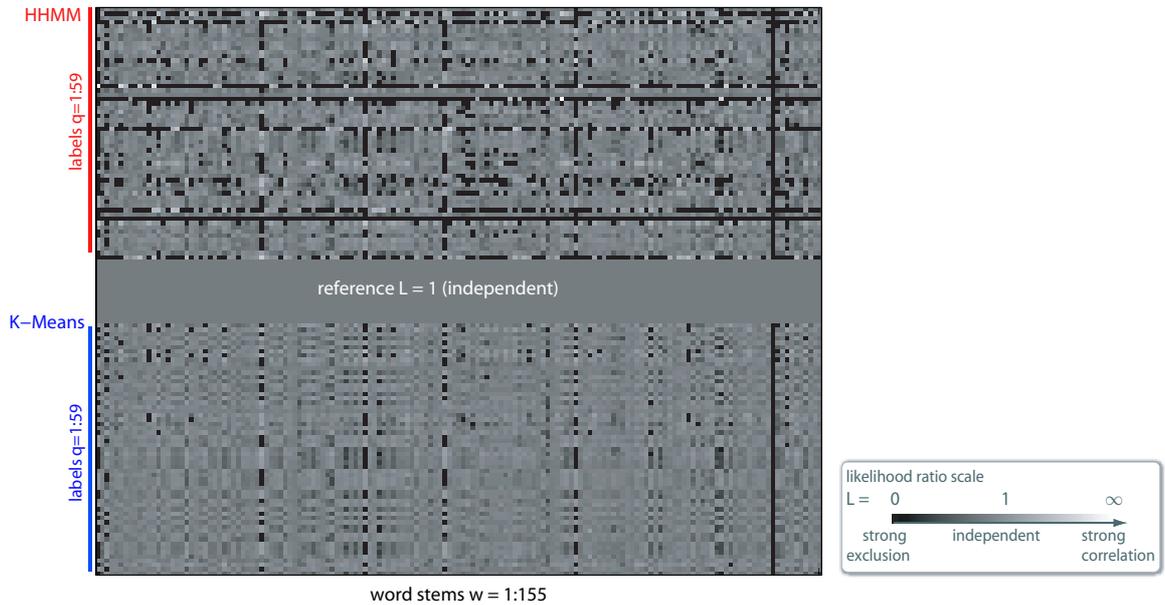


Figure 4.5: Word-association with labels generated by hierarchical HMM and K-means. The likelihood ratio function L is rendered in log-scale.

testing set, respectively. Note this weather cluster is found without including the “weather” concept as input to the HHMM or using the specific weather segments via supervised procedure.

The second half of [Table 4.1](#) shows details of another interesting label $(m, q) = (9, 1)$, the first label (among a total of seven) in a model learnt over the visual concepts $\{\textit{outdoors}, \textit{news-subject-face}, \textit{building}\}$. The HHMM emission probabilities for this state shows low probability for the concept *outdoors* and high probability for news-subject-face and building. The word-level precision and recall are around 20% and 30 ~ 60%, respectively; the list of words intuitively indicate the topic of politics and legal affairs, the audio-visual content of which often contains scenes of news-subjects and buildings; and the word list is effectively expanded by MT. Further examining the actual stories that contain this label (42 out of a total 216), we find that these 42 stories cover stories on Iraqi weapon inspection with 25.5%

Automatic			Manual
HHMM Label	Visual Concepts	Predicted Word-stems	News threads
(6,3)	people, non-studio-setting	storm, rain, forecast, flood, coast, el, nino, administer, water, cost, weather, protect, starr, north, plane, ...	El-nino98
(9,1)	outdoors, news-subject-face, building	murder, lewinski, congress, allege, jury, judge, clinton, preside, politics, saddam, lawyer, accuse, independent, monica, white, ...	Clinton-Jones Iraq-Weapon

Table 4.1: Example word-label correspondences. *HHMM label* (m, q) denotes the q^{th} state in the m^{th} HHMM model; the *visual concepts* are the features automatically selected for model m ; the *predicted word-stems* are the entries of $L^c(w, q)$ that have values in the top 5%.

recall and 15.7% precision, and simultaneously contain the stories on Clinton-Jones lawsuits with 44.3% recall and 14.3% precision.

4.4 Discussions

Having presented our algorithms and observations for metadata association for explaining and assessing temporal audio-visual patterns, we now briefly discuss related work and possible extensions along similar directions.

4.4.1 Related work in multi-modal association

Cross-modal association has been a recent topic of interest spanning the vision, learning, and information retrieval community. Duygulu et. al. [37] posed the problem of object recognition as using words to annotation regions in the image, given a training set that contains images with the object and keywords associated with the entire image. The learning process also rely on co-occurrence statistics

and machine translation un-smoothing using the EM algorithm. Jeon et. al. [62] approached the same image-region annotation problem using the relevance model in information retrieval by treating the words and image regions as independently generated *terms* in a *document*, the conditional probabilities of regions and words were obtained with empirical counts and background smoothing. Our work differs from the above in modeling the temporal structures within the audio-visual streams, instead of treating the video segments as independent entities.

4.4.2 Word association and beyond

It's worth noting that words are not equal to meanings. While the words associated with a few labels are easy to decipher, most labels are associated with diverse words from which distinct topics are not easy to find. The inherent co-occurrence in related words can be used to further reveal the meanings of words, and natural language processing techniques such as latent semantic analysis can be employed to compute these correlations (e.g., “white” and “house” often appear together) and to resolve the ambiguity of words (e.g., “rise” can refer to the stock index, the temperature, a higher elevation, or even an angry political reaction). In news videos, the concepts in the audio-visual stream may not be those present in the speech transcript. It maybe that different streams are at apparently different perceptual levels, such as “people” and “building interior” in the video frame actually correspond to an press conference held in Washington D.C. Although this is unlike the sentence-wise aligned bi-text between two languages [21], or the individual words annotating the visual content [37], we can explore the inherent correlations in both the audio-visual stream and the text stream, and use these inherent uni-modal patterns to influence the composition of cross-modal associations. Some of these issues lead us to the topics to be addressed in [chapter 5](#).

4.5 Chapter summary

In this chapter we examined the problem of finding meanings for multimedia patterns. We presented algorithms for the statistical association words and recurrent patterns in audio-visual streams. Experiments on news video corpus supports the existence of meaningful audio-visual clusters with consistent text terms.

Chapter 5

Discovering multi-modal patterns

Having addressed the discovery of temporal patterns in [chapter 3](#) and tagging meanings to these patterns in [chapter 4](#), we wonder whether the information in the audio visual channel and the text streams can be more tightly integrated. Can text be used to influence the patterns being discovered, instead of just as tags to the audio-visual clusters? Can we also exploit the internal patterns in text, instead of looking at individual words?

5.1 Multi-modal patterns in video

The problem of finding meaningful patterns using all the modalities leads to multimodal fusion, and this problem is better understood by taking a closer look at the video data. As shown in [Figure 5.1](#), the audio, video, and text streams in produced media streams are naturally asynchronous at both the signal level and at the semantic level: At the signal level, the information rates vary from a few dozens to a few thousands of bits per second because of the different processing rates in hearing, vision and reading systems. At the semantic level, the audio-visual content (e.g., an anchor person or a reporter on the street) may not immediately reflect the

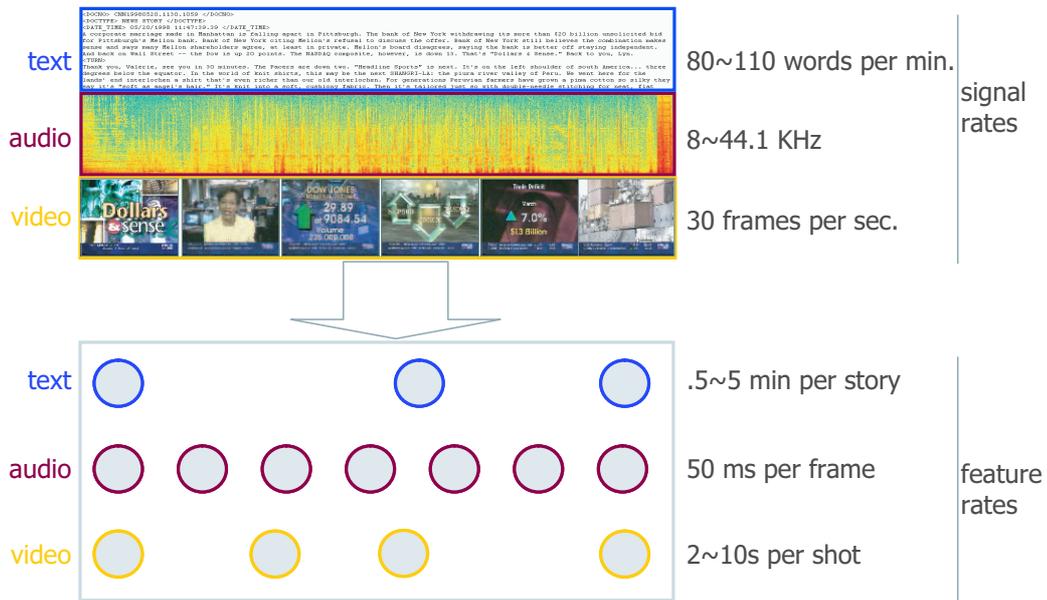


Figure 5.1: Asynchronous multimodal streams in news.

topics in the text, yet those spoken content may closely associate with the visuals a few shots later (e.g. a paper showing a newly passed bill, or the dashboard in the stock market). In addition, the semantics in audio, video, and text are at diverse levels among which direct associations may not exist. For instance, loud speech, background cheering, crowds at night, and words like *tonight*, *people*, *three hours* may imply a “sports fan celebration” event, yet they may also imply a “protest”.

Prior work addressed unsupervised pattern discovery on one input stream, such as finding the latent dimensions in text [50] or mining temporal patterns from audio-visual observations (chapter 3). Multi-modal fusion also appeared in various contexts, such as audio-visual speech recognition [88] where the audio and video are in exact sync, and cross-media annotation [62] where different modalities bear the same set of meanings. None of these models readily handles multi-modal fusion across the asynchronous and semantically diverse audio, visual and text streams.

We propose a layered dynamic mixture model for unsupervised asynchronous

multi-modal fusion. The model first groups each stream into a sequence of mid-level labels so as to account for the temporal dependency and noise. It then addresses the cross-stream asynchrony by allowing loose correspondence across different modalities and infers a high-level label from the mid-level labels in all streams.

Layered, or hierarchical models have come to the context of many research problems, our model is certainly not the first such structure, and hopefully not the last one either. This layered structure is similar to the commonly-used input-output model [10] or coupled temporal model [16] in that states are influenced by multiple streams, yet this structure is different from the prior models in that it infers additional hidden states that represent multiple streams instead of maintaining separate states for each modality. It is similar to the layered HMM for multi-modal user interaction [91], except that the layered HMM only handles fixed-rate inputs and enforces that the audio/visual information is only allowed to influence the user input for the same short time period. Another similar model is the dynamical system trees [52], where subset of the nodes in the model explicitly encode group structures. Our use of text information resembles those in multimedia annotation ([37, 62], chapter 4), except that we explicitly model the correlations in text, and the text information directly influences the composition of the multi-modal clusters instead of just serving as an explanation to the visual content.

We evaluate this model on Trecvid 2003 broadcast news videos. Results show that the multi-modal clusters have better correspondence to news topics than the clusters using audio-visual features only; on a subset of topics that bear salient audio-visual cues, they have even better correspondence than text. Manual inspection of the multi-modal clusters reveals a few consistent clusters that capture the salient syntactic units typical of news broadcasts, such as the financial, sports or commercial sections. Furthermore, for these clusters, the mixture model is able to predict audio-

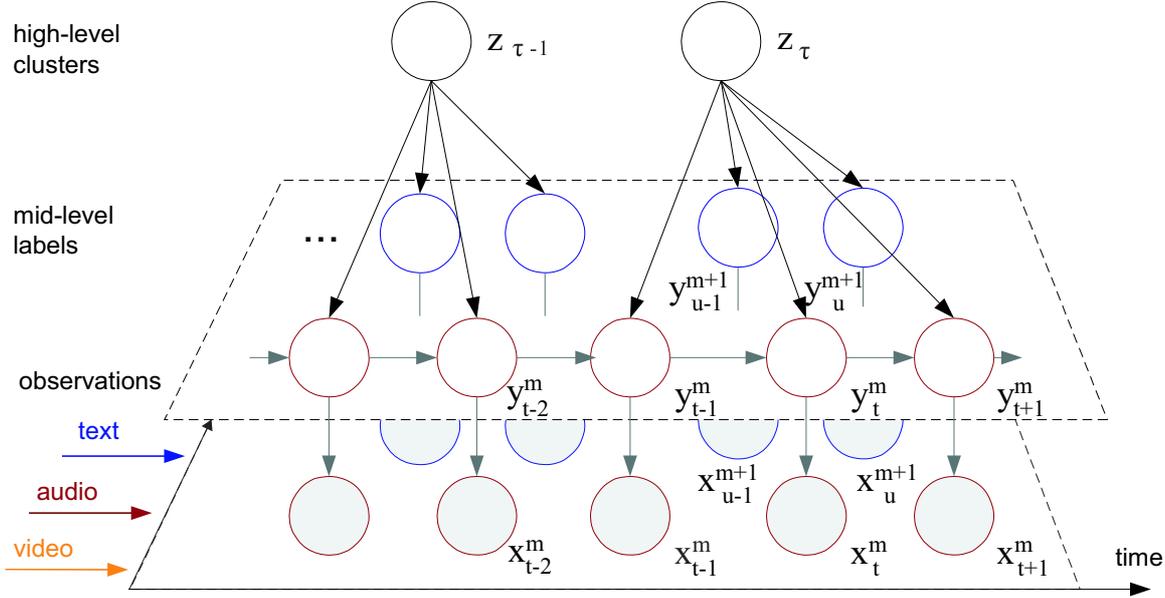


Figure 5.2: The layered dynamic mixture model, in the dimensions of time, various modalities and perceptual levels. Shaded nodes: observed, clear nodes: hidden; different colors represent nodes in different modality.

visual features and words that are indeed salient in the actual story clusters.

5.2 Unsupervised asynchronous multi-modal fusion

Multi-modal fusion for unsupervised learning differs from those for supervised learning [53] in that no labeled ground-truth is available to guide the fusion model. Therefore we use the data likelihood in generative models as an alternative criterion to optimize the multi-level dynamic mixture model.

5.2.1 The layered representation

The structure of the layered mixture model is shown in Figure 5.2. The layered dynamic mixture representation consists of the low-level feature streams, the mid-level labels, the high-level fused clusters, and the two layers of probabilistic models in between. We use the layers to divide the multi-modal fusion problem into two parts:

(1) modeling noise and temporal dependencies in the individual modalities and (2) fusing the modalities of different temporal resolution. This separation enables the use of different model structures in the lower layer so as to take advantage of the domain knowledge in each individual modality. A layered model is more flexible in its structure of representation and yields better clustering results than one-level clustering as seen in [subsection 5.3.3](#). Aside from enhancing the robustness and reducing parameter tuning as argued in [\[91\]](#), introducing layers in unsupervised learning intuitively conforms with the layered perceptual model, which allows more-efficient staged computational optimization in the different layers.

The layered mixture model has a set of different temporal indexes. As shown in [Figure 5.2](#), it allows different temporal index t^m for each input modality m . The lower layer models group each input stream x_t^m into a mid-level label stream y_t^m using generative models tailored to each modality, and y_t^m has the same temporal resolution as x_t^m .

We further partition the time axis into a set of non-overlapping loose temporal *bags* τ , and each bag contains a number of continuously indexed samples in each stream (assumed non-empty without loss of generality), denote as $\{t^m \mid t^m \in \tau\}$. We assign one top layer node z to each bag, and with a somewhat loose notation, we also use τ to index the top-layer nodes as z_τ .

5.2.2 Unsupervised mid-level grouping

The audio-visual streams exhibit temporal dependency, while independence between the keywords in adjacent stories can reasonably be assumed as they are often on different topics. For the former we use Hierarchical HMM for unsupervised temporal grouping as described in [chapter 3](#); for the latter we use probabilistic latent semantic analysis (PLSA) [\[50\]](#) to uncover the latent semantic aspects.

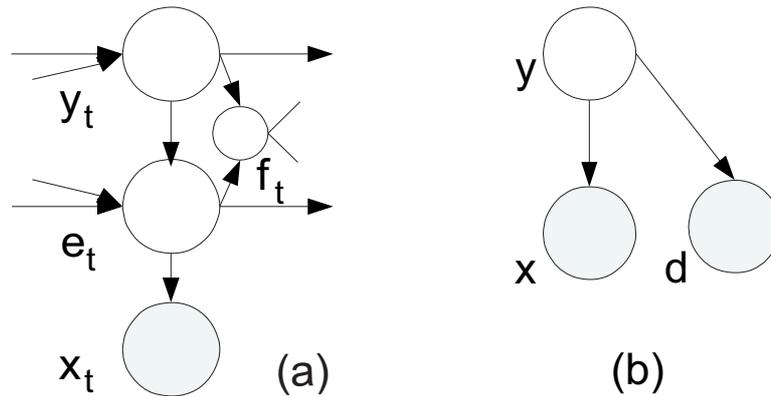


Figure 5.3: Models for mapping observations to mid-level labels. (a) Hierarchical HMM; (b) PLSA.

The HHMMs, described in [chapter 3](#) and graphically reproduced in [Figure 5.3\(a\)](#), are learned over the corpus with automatic model and feature selection using the mutual information and Bayesian information criteria (BIC). The resulting HHMM typically has two to four top-level states. With this model, we can incorporate existing partial knowledge of news video syntax and make the learned states in the HHMM more meaningful. This is achieved by taking into account the known news story boundaries: the HHMM inference algorithm only allows highest-level state transitions at these boundaries, hence restricting the segment coming from the same story to stay in the same branch of the Markov chain hierarchy. The learned HHMM then labels the videos stream with the most likely state sequences.

The PLSA model [\[50\]](#) is shown in [Figure 5.3\(b\)](#). Observing the words x in story d , the model learns the conditional dependencies between the hidden semantic aspects y and both observed variables. The double mixture structure of PLSA provides more flexibility for capturing word distributions than a simple mixture model, and this is achieved by replacing a single conditional parameter $p(x|y)$ with a linear combination of conditionals involving the documents d , i.e., $p(x|y)p(y|d)$. The inference of PLSA is carried out with the EM algorithm. We have observed

that the PLSA distributions more accurately capture sets of semantically related words, rather than being deluged by the frequent words.

5.2.3 The fusion layer

In the top-level fusion layer, the boundaries for the temporal bags τ lie on syntactically meaningful boundaries in the video, such as scene transition points or news stories boundaries. For efficiency, the clustering in each layer is carried out in separate stage, i.e., the values of mid-level labels y are taken as the maximum-likelihood value from the lower-layer model and considered as “observed” when inferring the high-level node z .

Denote the set of mid-level labels within each bag (given a priori) as $y_\tau = \cup_m \{y_t^m, t^m \in \tau\}$. The model prescribes that the mid-level nodes y_τ in each modality are influenced by z_τ , and given z_τ , the different modalities are conditionally independent. i.e.,

$$p(y_\tau, z_\tau) = p(z_\tau) \prod_m \prod_{t^m \in \tau} p(y_t^m | z_\tau) \quad (5.1)$$

Under this assumption, the temporal orders within each bag no longer influence the value of z_τ ; when each of y_t^m takes discrete values, y_τ can be represented by a set of multi-dimensional co-occurrence counts

$$c(m, \tau, y) = \#\{y_t^m = y, t^m \in \tau\}.$$

Intuitively, this is to treat the mid-level labels y^m , obtained by *de-noising* x^m , as if they were generated by multinomial draws conditioned on the high-level *meaning* z .

According to this definition, we rewrite the complete-data log-likelihood of y

and z in bag τ from [Equation 5.1](#), and estimate the model parameters with the EM algorithm:

$$\begin{aligned} l(\tau, z) &\stackrel{\Delta}{=} \log p(y_\tau, z_\tau = z) \\ &= \log p(z) + \sum_{(m,y)} c(m, \tau, y) \log p(y^m | z). \end{aligned} \quad (5.2)$$

The E-step reads:

$$p(z_\tau = z | y_\tau) = \frac{\exp l(\tau, z)}{\sum_{z \in \mathcal{Z}} \exp l(\tau, z)} \quad (5.3)$$

The M-step follows:

$$p^*(z) = \frac{1}{T} \sum_{\tau=1}^T p(z_\tau = z | y_\tau) \quad (5.4)$$

$$p^*(y^m | z) = \frac{\sum_{\tau=1}^T c(m, \tau, y) p(z_\tau = z | y_\tau)}{\sum_y \sum_{\tau=1}^T c(m, \tau, y) p(z_\tau = z | y_\tau)} \quad (5.5)$$

We can extend this basic fusion model to include a joint inference from observations x_t^m to the highest level z_τ , to model dependency within each temporal window, or to allow flexible temporal bags to be learned while performing model inference.

5.3 Experiments

We test the proposed fusion model on TRECVID news corpus [[122](#)]. This data set contains 151 half-hour broadcasts of *CNN Headline News* and *ABC World News Tonight* from January to June, 1998. The videos are encoded in MPEG-1 with CIF resolution; also available are the ASR transcripts produced by LIMSI [[43](#)]. We partition the dataset into four quadrants each containing half of the videos from one

channel.

5.3.1 Multi-modal features

We extract from each video the following sets of low-level audio-visual descriptors, visual concepts, and text terms as the base layer in the hierarchical mixture model:

1. The color histogram of an entire frame is obtained by quantizing the HSV color space into fifteen bins: white, black and gray by taking the extreme areas in brightness and saturation; equal-width overlapping bins on the hue values resembling the six major colors red, yellow, green, cyan, blue and magenta in high and low saturations, respectively. Since the histogram is normalized with respect to the size of each frame (and hence is linearly dependent), we exclude an arbitrary bin (yellow with low saturation) to make a linearly independent 14-dimensional feature vector, in order for the Gaussian observation probabilities not to degenerate in the later learning stage. The color histogram is averaged over a time window of one second.
2. Motion intensity consists of the average of motion vector magnitude and the least-square estimate of horizontal pan from the MPEG motion vectors, extracted every second (appendix C).
3. The audio features contain a four-dimensional vector every half a second: the mean *pitch* value; the presence/absence of *silence* and *significant pause*, the latter obtained by thresholding locally normalized pause length and pitch jump values; six-class *audio category* labels from a GMM-based classifier (silence, female, male, music, music+speech, or other noise) [100].

4. A visual concept vector contains the confidence scores modeling how likely the keyframe of a shot is to contain a set of visual concepts. The concepts used in this work are pruned from a lexicon of over a hundred in order to make them as specific as possible while having reasonable detection accuracy. They include: five events - *people, sport, weather, cartoon, physical violence*; five scenes - *studio setting, non-studio setting, nature-vegetation, nature non-vegetation, man-made scene*; twelve objects - *animal, face, male news person, male news subject, female news person, female news subject, person, people, crowd, vehicle, text overlay, graphics*. These concept scores are the fusion results of SVM classifiers via various strategies (min., max, linear combination, etc.) [5], we normalize the scores to between 0 and 1 and uniformly quantize them into three levels.

5. Keyword features can be obtained from either the closed captions or the automatic speech recognition (ASR) transcripts. Stemming, part-of-speech tagging and rare word pruning are performed, retaining a 502-token lexicon of frequent nouns, verbs, adjectives and adverbs. The tf-idf score of the words within a news story are used as the feature vector.

After feature extraction, one HHMM is learned on each of the color, motion and audio features. The visual concepts, due to their diverse nature, yield three HHMMs grouped by the automatic feature selection algorithm. The words in all stories are clustered into 32 latent dimensions with PLSA. The fusion model then infers a most-likely high-level hidden state from all the mid-level states in each story, taking one of 32 possible values (chosen as in approximately the same order of magnitude with the number of news topics in [subsection 5.3.3](#)). The multi-level clustering algorithm runs in linear-time, and it typically takes less than three hours on a 2GHz PC for

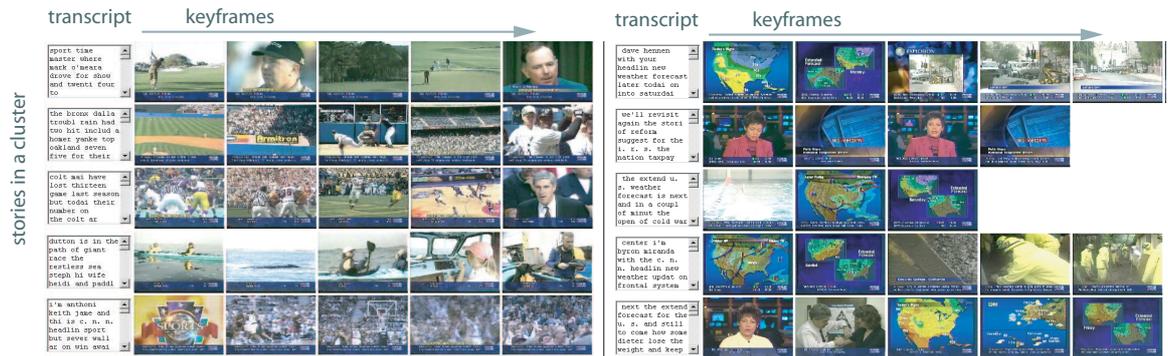


Figure 5.4: Example clusters, each row contains the text transcript and the keyframes from a story. Left: cluster #5, sports, precision 16/24. Right: cluster #29, weather, precision 13/16. The models are learned on CNN set A, evaluated on set B.

19 hours of video.

5.3.2 Inspecting the clusters

We first inspect a story-board layout of each cluster. A weather forecast cluster (shown in [Figure 5.4](#)) results from the consistent color layout and similar keywords, while the sports cluster is characterized by the motion dynamics and visual concepts regarding people. Other interesting clusters include segments of CNN financial news (precision 13/18 stories) with similar graphics, anchor person and common transitions between them. There is also a commercial cluster (7 out of 8) characterized by audio-visual cues, since there are no consistent text terms across different commercials. These observations suggest that meaningful multi-modal clusters can be captured by modeling recurrent syntax within and across multiple input streams.

We also inspect the distribution of the multi-modal features. We compare the most likely feature distribution predicted by the model and those observed in the actual story clusters. An agreement between these two would suggest that this may be a salient feature that the model manages to capture. In [Figure 5.5](#) we plot

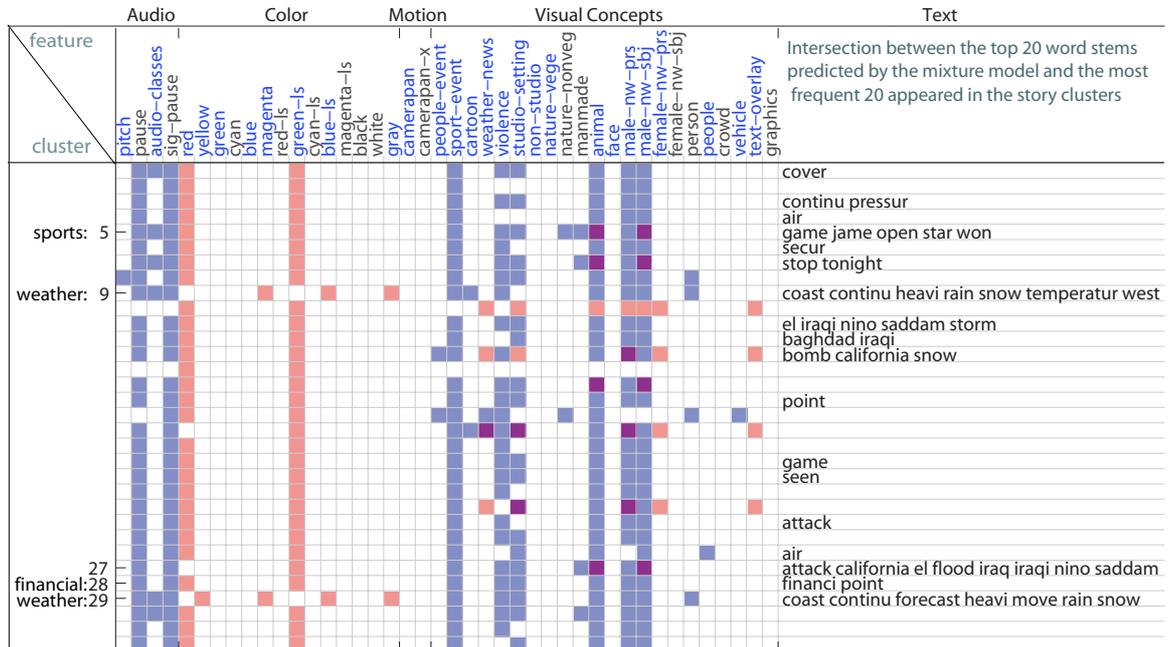


Figure 5.5: The most probable features predicted by the model (red) and observed in the clusters (navy). A magenta cell results from the joint presence from both, and a blank one indicates that neither has high confidence. The features retained during the automatic feature selection process are shown in blue. The models are learned on CNN set A, evaluated on set B.

the predicted and observed cluster-feature pairs with high statistical confidence into two color channels by applying simple cut-off rules for having high prediction confidence. We require: $p > 0.8$ for both the top-level mixture model $p(y^m|z)$ and the emission probabilities $p(x|y)$ of the discrete-valued features, a peak greater than twice the second-highest mode for the continuous features described by a mixture of Gaussians, or the intersection of the top 20 words in the PLSA probabilities $p(x|y)$ with those in the text transcript. This visualization intends to provide intuition by linking the features at the bottom level and the cluster labels z at the top level, and by-passing the mid-level cluster labels y that mostly encode the syntactic pattern but not semantic meanings.

In the sports cluster (shown in Figure 5.4, left; and Figure 5.5, row #5), we can see

that (1) both the model and the actual stories in the cluster agree on sports-related words: game, star, won, etc. (2) the model and the observations agree on the high confidence of the visual concepts *male-news-subject*, which is reasonable given that the cluster corresponds to the topic “1998 Winter Olympics”. In the two weather clusters (Figure 5.5, row #9 and #29, and Figure 5.4, right) the model predicts (1) yellow and blue in the color layout, which is also salient in Figure 5.4, (2) a set of weather-related words such as rain, snow, coast, forecast. We can also interpret from the common keywords that some clusters are a mixture of two or more general categories, e.g., politics and weather in cluster #27. These observations explain the composition of the meaningful top-level clusters by providing evidence from the multi-modal feature streams.

5.3.3 News topics

Topics are the semantic threads in news, defined as “an event or activity, along with all directly related events and activities” [121]. We compare the multi-modal clusters with the 30 news topics seen in our corpus (labeled by TDT [121], covering $\sim 15\%$ of all news stories in the TRECVID set). We use the TDT evaluation metric *detection cost*, a weighted sum of the precision and recall for each topic s :

$$C_{det}(s) = \min_z \{P_M(z, s) \cdot P(s) + P_{FA}(z, s)(1 - P(s))\} \quad (5.6)$$

Here P_M and P_{FA} are the miss and false alarm probabilities of a cluster z with topic s , i.e., $P_M = |s \cap \bar{z}|/|s|$ and $P_{FA} = |\bar{s} \cap z|/|\bar{s}|$; and $P(s)$ is the prior probability of topic s . A “best” cluster is chosen as the one minimizing this weighted sum of P_M and P_{FA} , and its detection cost is then assigned as the detection cost $C_{det}(s)$ for topic s . This raw detection cost is then normalized by $\min\{p(s), 1 - p(s)\}$ to avoid

trivial decisions, i.e.,

$$\bar{C}_{det}(s) = \frac{C_{det}(s)}{\min\{p(s), 1 - p(s)\}}.$$

Note the smaller the value of $\bar{C}_{det}(s)$, the better a topic can be represented by one of the clusters.

We compare the layered fusion structure with a one-level clustering algorithm, the latter is obtained by substituting the observations x into the places of y in the multimodal inference (Equation 5.2-5.5), with other parameters fixed (Gaussians for the color and motion features, multinomials for the rest, 32 clusters). The average detection cost for the former is lower than the latter by 0.240, i.e., multimodal clusters achieved an average relative improvement of 24% scaled by the respective topic priors. We also compare the multi-modal clusters with single modal audio-visual clusters by taking the minimization in Equation 5.6 is across all single-modality HHMMs (that are equivalent to those in chapter 3-4). The multimodal clusters outperform the uni-modal ones by decreasing 0.227 in the average detection cost (i.e., a relative improvement of 22.7%).

In seven out of the thirty topics, multi-modal clusters have lower detection costs than using text alone (i.e., the PLSA clusters), these topics include *1998 Winter Olympics*, *NBA finals* and *tornado in Florida* among others. The topics of improved performance tend to have rich non-textual cues, for instance the correspondence of cluster #5 in Figure 5.5 to *1998 Winter Olympics* can be explained by the prevalence of *music+speech* in the audio and *male-news-subject* in the visual concepts, which is reasonable since most of these sports stories begin with a lead-in anchor graphics accompanied by music and the commentator.

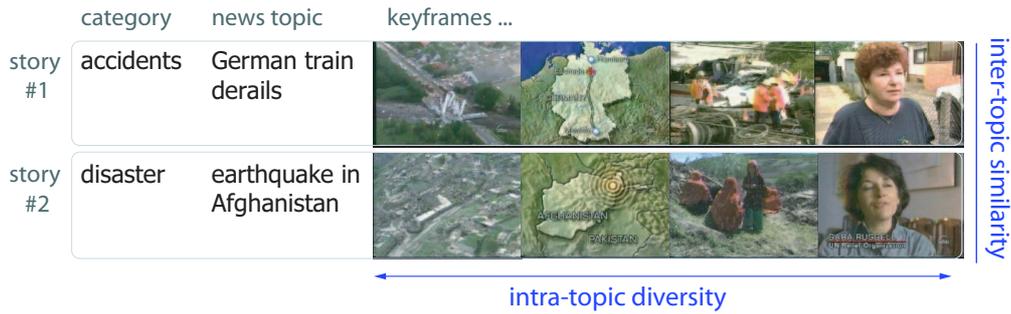


Figure 5.6: Two example stories from TDT-2 corpus [121] that exhibit intra-topic diversity and inter-topic similarity.

5.3.4 Discussions

Note that the topics being evaluated are defined and labeled solely with the news transcripts [121], and this may implicitly bias the evaluation towards the evaluations with text-only. For example, cluster #28 in Figure 5.5 corresponds to the topic *Asian economy crisis*, yet a better correspondence can be obtained with a PLSA cluster with the most-frequent keywords *dow*, *nasdaq*, *dollar*, etc. Comparing the topic correspondence and the cluster inspection in subsection 5.3.2 also suggest that the multimodal clusters may be at a different level than the news topics. For example, Figure 5.6 shows a few keyframes from two different topics: *German train derails*, defined in the broader *accident* category, and *earthquake in Afghanistan*, in the *disaster* category. It is obvious for human viewers, that the visual content within each topic are quite diverse (maps, aerial shots, rescue scenes), yet the visual appearance across these topics can be very similar. This suggest that a new set of definitions for topic taking into account not only *what* is in the news story but also *how* the story is covered may be more suitable for multimedia.

5.4 Chapter summary

In this chapter we discussed the problem of finding multimedia patterns across multimodal streams. We presented layered dynamic mixture model for fusion asynchronous information from the visual, audio, and text streams. On news video corpus, compared to text topics in the news, layered multimodal patterns has shown better correspondence than audio-visual patterns alone, or one-level clusters, it has also improved upon text clustering on a subset of topics with salient audio-visual cues.

Chapter 6

Conclusions and future work

In this chapter, we shall first summarize the work presented in this thesis, along the lines of the three sub-problems that have been tackled – learning temporal video patterns, associating patterns to meanings, and discovering patterns in multimodality. We shall then present a few potential areas of research for the application and extension of this work.

6.1 Research summary

This thesis formulated and addressed the problem of unsupervised pattern discovery in multimedia sequences. This problem is seen as finding recurrent and syntactically consistent multimedia segments from a collection of sequences without knowing their syntactic characteristics a priori. Unsupervised pattern discovery complements the supervised pattern classification problem being solved for multimedia and many other domains, since unsupervised discovery can help building an initial model and providing the initial annotations, filtering the feature pool, assessing the space of concepts and semantics that needs to be learned, or dynamically monitoring the data statistics.

There are two main ideas behind this work: (1) meaningful patterns can be discovered by statistical temporal models from production syntax, via low-level and mid-level features; (2) the mapping between syntactic patterns and semantic meanings can be established via the joint statistical analysis of media and metadata. We have worked on three specific aspects within.

6.1.1 Mining statistical temporal patterns

We have proposed novel models and algorithms for the discovery of multimedia temporal patterns. We proposed hierarchical hidden Markov model for capturing generic temporal patterns. This model generalizes upon the celebrated hidden Markov model, from which it inherits the ability to model the temporal correlations in streams and the flexibility to allow segments of arbitrary duration; furthermore it extends the HMM in being able to model multi-level across-event temporal transitions. The entire HHMM model is learned efficiently using the EM algorithm without the need for prior training of the individual components.

We have proposed strategies for automatically adapting the complexity of the unsupervised pattern model to that of a new domain. This adaptation is achieved with stochastic search strategies in the model space, where the search is conducted via reverse-jump Markov chain Monte Carlo, and Bayesian information criterion is used to evaluate the resulting model. This strategy finds a favorable trade-off between the data likelihood and the model complexity, while being much more computationally efficient than exhaustively searching the state-space, being quite stable to the initialization of model size, and having probabilistic convergence guarantees thanks to the nature of the Monte Carlo simulation.

We have also proposed strategies for selecting the optimal set of descriptors for pattern discovery. We perform unsupervised feature selection on temporal streams

by partitioning the original feature set into several consistent subsets, using mutual information as the consistency criterion. We then eliminate redundant features from each subset, using Markov blanket filtering.

Evaluations of this temporal pattern mining scheme have showed promising performances in broadcast soccer and baseball programs. The unsupervised pattern mining algorithms achieved 75% ~ 83% accuracy in distinguishing *plays* and *breaks* in the games, these performances are comparable to their supervised counterparts. The automatically identified feature sets are intuitive and consistent with those manually identified for this domain. We can see that the resulting models match our intuitions in their feature distributions and temporal transitions via a visualization system.

6.1.2 Assessing the meanings of audio-visual patterns

We presented novel approaches towards automatically explaining and evaluating the patterns in multimedia streams. This approach links the computational representations of the recurrent patterns with tokens in the metadata stream. The metadata, such as the words in the closed caption or the speech transcripts, are noisy yet informative representations of the semantics in the multimedia stream. The linking between the representation of audio-visual patterns, such as those acquired by a HHMM and the metadata is achieved by statistical association.

Evaluations on large news video corpora reveals interesting patterns with consistent word associations, representing prevalent news themes.

6.1.3 Discovering multi-modal patterns

We further developed the solutions to multimedia pattern discovery by explicitly addressing patterns across multiple modalities. This problem is of interest because

neither the audio-video nor the text alone would be a complete representation of the multimedia content, and the challenge lies in the natural asynchrony across different modalities. We have proposed a layered dynamic mixture model, this model separates the modeling of intra-modality temporal dependency and that of inter-modality asynchrony.

On the TRECVID [122] broadcast news corpus the layered dynamic mixture model identified a few syntactically salient and semantically meaningful patterns corresponding to news topics and news program sections such as TV commercials. On a few perceptually salient news topics the automatically discovered multimodal topics showed superior detection performances than the text baseline.

6.2 Improvements and future directions

Multimedia pattern discovery is a challenging task. While this thesis began an effort in presenting solutions to its important aspects, there are necessarily gaps to be found in the framework that can be addressed in the future. While we have tested mainly on sub-domains within produced content, there are many other domains that these techniques are applicable to. In the few subsections that follow, we shall try to spell out a few applications and extension that we are interested in pursuing in the near future.

6.2.1 Applications of multimedia patterns

Multimedia patterns represent the meaningful syntax in the domain, they are thus useful for content browsing and are potentially useful in several related applications.

These include:

- Multimodal event detection. While unsupervised discovery will most likely

capture frequent events, the resulting model would help provide a “deviation” measure to incoming new content (similar to [101]), and will therefore be useful for novel event detection as well.

- **Multimedia retrieval.** On one hand the patterns can be learned offline on the database to help re-rank the retrieved results similar to the re-ranking in text retrieval [74]. The approaches can integrate plain retrieval score and cluster-conditioned retrieval score, or help reduce the search space for each query. On the other hand a learned pattern model can be used to help label the incoming queries in a query-by-example scenario, thus invoking different scoring strategies or multi-modal combination strategies. This can extend the query-class dependent multimedia retrieval models via manual [25, 135] or automatic [67] query-class determination.
- **Multimodal summarization.** If we view multimedia summarization as an entity-utility optimization problem [116], a multimedia pattern model can be incorporated into the utility function of a candidate summary. Scores from the model (in the form of data likelihood or classification confidence) can then influence the utility of a candidate set of entities (e.g., frames, shots, scenes, stories, audio segments, text) in two ways: (1) They can preserve the continuity and minimum length/duration of entities and maintain a pattern instance being comprehensible. (2) They can improve the coverage among all multimedia entities and to ensure that as many different pattern classes as possible are represented.

6.2.2 Working with unstructured data

This thesis has tested pattern discovery in produced video contents. The ideas of capturing syntactic patterns and mapping them to semantics via metadata are also applicable to other produced content collections, once a sufficient pool of content descriptors is supplied.

What then, if we look outside of professionally produced videos? There is an abundance of videos being captured and stored without professional editing (which thereby imposes production syntax), they are found as meeting and seminar recordings, consumer home videos, surveillance feeds, as well as professional raw footage. It is said that CNN gains 300 hours of such raw video everyday. Intuitively there are indeed spatial-temporal relationships within a content domain, such as the topic transitions common to many technical talks, the interaction patterns of attendees due to a specific conference room setup, the people movement patterns in a specific surveillance location, or the people and activity distinctions in the raw footage about parties or outdoor trips. It is noteworthy that a shot is typically longer in this footage than in the produced content, if there is a shot break at all. As a result the patterns therein more likely lie in answers to the questions “who, when and how” than in the low-level features that often approximate the editing and production operations. These features can be obtained from state-of-the-art detector and tracker outputs [127, 93], or generic concept detectors [122]. Aside from including adequate features, we also plan on incorporating context and partial knowledge in the forms of model structure and learning constraints, as outlined in the next subsection.

6.2.3 Incorporating context

In unsupervised pattern discovery we aim to learn recurrent patterns without the often expensive labeled examples. Needless to say being supervision-free does not exclude using partial domain knowledge to guide the discovery process. Building on top of the unsupervised learning framework presented in this thesis, we plan to start from a few simple contextual knowledge such as: (1) uniqueness – e.g., in a news program, there is usually one and only one weather and one sports section per program; (2) exclusion – e.g., two female primary anchors usually do not appear in the same broadcast. These constraints can be incorporated by introducing additional dependencies in the graphical model, the expected outcome would be patterns that correspond better to meanings or topics in the domain.

While this is just a starting point towards incorporating knowledge and context in multimedia, the computational formulation and instrumentation of rich knowledge is certainly a long-term challenge.

6.2.4 Multi-modal, multi-source fusion

In this thesis we have focused on the integration of asynchronous sources without requiring signal-level synchronization for the low-level data samples. It is worth noting that both fusion strategies can nonetheless be integrated, and this strategy shall be able to vary across different content or scene types (e.g., a talking head in news or a voice-over in a film). This variation conforms with the fact that the level of perceptual fusion would differ depending on what are being perceived and fused (e.g., texture, motion, pitch, versus object, phoneme or word) [68].

Multimedia streams do not come in isolation. News videos, for example, co-exist in time with many other online or print news materials. Home videos, on the other

hand, would bear multiple correlations with the calendar on the home PC, travel logs as recorded by the GPS, and weather reports for the surrounding areas. Some of these sources can be tightly integrated as another data modality, while others may be better incorporated as contextual constraints.

Multimedia pattern mining faces grand challenges on two fronts: the data analysis challenge of building powerful and versatile models, and the biological mystery of perception and multi-sensory integration. While this effort is ambitious and far from complete, I look forward to making multimedia data more useful with the coming advances in both the computational versatility and perceptual insights.

Appendix

A The inference and estimation for HHMM

This appendix provides details of the inference and estimation algorithm of a HHMM. The rest of this section is arranged as follows: for convenience [subsection A.1](#) repeats the representation of a HHMM from [subsection 3.3.1](#), [subsection A.2](#) outlines the three questions in HHMM inference and estimation, [subsection A.3](#) presents the generalized forward-backward algorithm for estimating the posterior probability of the multi-level hidden states, [subsection A.4](#) outlines the Viterbi algorithm for decoding the maximum-likelihood multi-level hidden state sequence, [subsection A.5](#) contains the parameter estimation steps, and [subsection A.6](#) discusses the computational complexity of the inference with HHMM.

A.1 Representing an HHMM

For notation convenience we introduce a single index for all multi-level state configurations in the HMM. Denote the maximum state-space size of any sub-HMM as Q , we use the *bar notation* ([Equation 3.1](#)) to write the entire configuration of a hierarchical state from the top (level 1) to the d^{th} level with a Q -ary d -digit integer,

with the lowest-level states at the least significant digit.

$$q^{(d)} = \overline{(q^1 q^2 \dots q^d)} = \sum_{i=1}^d q^i \cdot Q^{d-i} \quad (6.1)$$

Here we use superscript without brackets q^i to denote the i^{th} digit in a d -digit number $q^{(d)}$ (i is never the exponent), with $0 \leq q^i \leq Q-1; i = 1, \dots, d$. When there is no confusion we drop the bracketed superscript for $q^{(d)}$ and use the short hand q to represent the d -level configuration. The subscript of q are indexes of time, e.g. $q_{1:t}$ would be the hidden state sequence up to time t . For example, a two-level HHMM with two top-level states and three sub-states each, seeing the second children state in the second top-level model for time $t = 2$ would be $q_2 = 4$.

We represent the whole parameter set Θ of an HHMM as: (1) Emission parameters B that specifies the distribution of observations given the state configuration. i.e., the means μ_q and covariances σ_q when emission distributions are Gaussian. (2) Markov chain parameters λ^d in level d indexed by their parent state configuration $q^{(d-1)}$. λ^d in turn include: (2a) Within-level transition probability matrix A_q^d , where $A_q^d(i, j)$ is the probability of making a transition to sub-state j from sub-state i , and i, j are d^{th} -level state indexes having the same parent state $q^{(d-1)}$. Equivalently this is also written as $A^d(q^{(d)}, q^{(d)})$ for two full configurations $q^{(d)}, q^{(d)}$ that only differ at their d^{th} digit and lower. (2b) Prior probability vector π_q^d , i.e., the probability of starting in a children state upon “entering” q . (2c) Exiting probability vector e_q^d , i.e., the probability “exiting” the parent state q from any of its children states. All elements of the HHMM parameters are then written as in [Equation 3.2](#), for notation convenience and without loss of generality we assume that there is only one state at the very top level and thus there is no need to define the transition probabilities

there.

$$\begin{aligned}
\Theta &= \left(\bigcup_{d=2}^D \{\lambda^d\} \right) \bigcup \{B\} \\
&= \left(\bigcup_{d=2}^D \bigcup_{i=0}^{Q^{d-1}-1} \{A_i^d, \pi_i^d, e_i^d\} \right) \bigcup \left(\bigcup_{i=0}^{Q^D-1} \{\mu_i, \sigma_i\} \right)
\end{aligned} \tag{6.2}$$

A.2 Three question in HHMM inference and estimation

Having defined the representation of the HHMM, we now investigate its inference and estimation algorithms. Analogous to HMM [99], we would like to use HHMM for answering three questions:

1. Compute likelihood and posteriors. Given a model Θ and observation sequence $x_{1:T}$, how do we compute the probabilities of $x_{1:T}$, or the probabilities of the corresponding hidden states $q_{1:T}$?
2. Find the optimal multi-layer state sequence. Given a model Θ and observation sequence $x_{1:T}$, can we find a multi-layer hidden state sequence $\hat{q}_{1:T}$ that best “explains” the observations?
3. Parameter estimation. How do we adjust the parameters in Θ so that the observations $x_{1:T}$ are better explained?

The first question is the estimation problem, here we would like to use the generalized chain structure to efficiently compute the data likelihood and the posterior probability of the hidden states given the observations:

$$L \triangleq P(x_{1:T}; \Theta), \quad P(q_{1:T}|x_{1:T}; \Theta) = L^{-1}P(x_{1:t}, q_{1:T}; \Theta) \tag{6.3}$$

The solution uses a multi-layer forward-backward algorithm that iteratively marginalizes the hidden states over time.

The second question is the decoding problem, where we would find the sequence that jointly maximizes the observation and state-transition likelihood using a multi-layer Viterbi algorithm.

The last question concerns with maximizing the likelihood of seeing the data by iteratively finding a new parameter set that “better” explains the observations.

A.2.1 The HHMM transition parameters

For the convenience in presenting the inference and estimation, we define hyper-initial probabilities $\tilde{\pi}(q)$ and hyper-transition probabilities $\tilde{a}(k', k, d)$.

$$\pi_q = \prod_{d=1}^D \pi_{q^d}^d, \quad q = 0, \dots, Q^D - 1 \quad (6.4)$$

$$\tilde{a}(q', q, d) = \prod_{i=d}^D e_{q^i}^i \pi_{q^i}^i \cdot A^d(q'^d, q^d), \quad (6.5)$$

$$q', q = 0, \dots, Q^D - 1; \quad d = 1, \dots, D$$

Note that [Equation 6.5](#) represents the probability of going from state q' to state q with a transition happening at level d , and no transition should happen in levels above d , i.e., the highest $d - 1$ digits q_1 through q_{d-1} in q and q' should be the same, while digits below and including level d could still be the same. The hyper-transitions are parameterized as $\tilde{a}(q', q, d)$ instead of $\tilde{a}(k', k)$, in order for the complete-data likelihood to be factorable (and hence has efficient EM) in [subsection A.5](#).

A.3 The forward-backward algorithm

In order to compute the observation sequence likelihood $L \triangleq P(x_{1:T} ; \Theta)$, we need to define auxiliary forward variables $\alpha_t(q)$, i.e., the probability of the observation sequence up to time t and the HHMM being in state q under the current model Θ :

$$\alpha_t(q) \triangleq P(x_{1:t}, q_t = q \mid \Theta) \quad (6.6)$$

We then use α to iteratively compute the data likelihood L , as shown in equations (6.7)-(6.9). We use the short-hand notation of $b_q(x)$ as the emission probability of seeing observation x conditioned on state q , where its specific form will depend on the nature of the emission distribution, usually Gaussian or multinomial. Note that each iteration step marginalizes the transitions *into* state q_t , and thus making a polynomial-time algorithm for marginalizing exponential number possibilities of all hidden state sequence $q_{1:T}$.

Initialization:

$$\alpha_1(q) = \pi_q b_q(x_1); \quad q = 0, \dots, Q^D - 1 \quad (6.7)$$

Iteration:

$$\begin{aligned} \alpha_{t+1}(q) &= b_q(x_{t+1}) \sum_{q_t} \sum_d \alpha_t(q_t) \tilde{a}(q_t, q, d); \\ t &= 1, \dots, T - 1; \quad q = 0, \dots, Q^D - 1 \end{aligned} \quad (6.8)$$

Termination:

$$L = P(x_{1:T} ; \Theta) = \sum_q \alpha_T(q) \quad (6.9)$$

Similarly, we can define and compute the backward variable $\beta_t(q)$ representing the probability of seeing observations after time t given the hidden state at t . They will later become useful for parameter estimation in [subsection A.5](#):

$$\beta_t(q) \triangleq P(x_{t+1:T} \mid q_t = q; \Theta) \quad (6.10)$$

Similar to the forward iterations, $\beta_t(q)$ are computed as follows.

Initialization:

$$\beta_T(q) = 1; \quad q = 0, \dots, Q^D - 1 \quad (6.11)$$

Iteration:

$$\begin{aligned} \beta_t(q) &= \sum_{q'} \sum_d \beta_{t+1}(q') b_{q'}(x_{t+1}) \tilde{a}(q, q', d); \\ t &= T - 1, \dots, 1; \quad q = 0, \dots, Q^D - 1 \end{aligned} \quad (6.12)$$

A.4 The Viterbi algorithm

The Viterbi algorithm for decoding the optimum state sequence $\hat{q}_{1:T}$ is very similar to the forward algorithm except the following changes: (1) Replace the summation with maximum in [Equation 6.8](#). Thus we use the auxiliary variable $\delta_t(q)$ to represent the likelihood of the “best” single path up leading to state q at time t , instead of marginalizing over all paths in $\alpha_t(q)$. (2) Keep backtrack pointers $\psi_t(q) = (\hat{q}', \hat{e}')$ where \hat{q}' is the “best-likelihood” states at $t - 1$ that leads to state q at time t , and \hat{e}' is the corresponding *exit level* taking values from D to 2 ; (3) Do back tracing after the forward path, i.e., pick the “best-likelihood” from time T , and trace back to $t = 1$ according to ψ , recover the optimum path $(\hat{q}_{1:T}, \hat{e}_{1:T})$.

The multi-level Viterbi iterations are as follows.

Initialization:

$$\begin{aligned} \psi_1(q) &= (0, 1); \quad \delta_1(q) = \pi_q b_q(x_1); \\ q &= 0, \dots, Q^D - 1 \end{aligned} \quad (6.13)$$

Iteration:

$$\psi_{t+1}(q) = \arg \max_{q', d} \delta_t(q') \tilde{a}(q', q, d) \quad (6.14)$$

$$\delta_{t+1}(q) = b_q(x_{t+1}) \cdot \max_{q', d} \delta_t(q') \tilde{a}(q', q, d); \quad (6.15)$$

$$t = 1, \dots, T - 1; q = 0, \dots, Q^D - 1$$

Back tracing:

$$\hat{q}_T = \arg \max_q (\delta_T(q)); \quad \hat{e}_T = 1; \quad (6.16)$$

$$(\hat{q}_t, \hat{e}_t) = \psi_{t+1}(\hat{q}_{t+1}); \quad t = T - 1, \dots, 1; \quad (6.17)$$

A.5 Parameter estimation

Denote Θ the old parameter set, $\hat{\Theta}$ the new (updated) parameter set, then maximizing the data likelihood L is equivalent to iteratively maximizing the expected value of the complete-data log-likelihood Ω :

$$\Omega(\hat{\Theta}, \Theta) = E[\log(P(q_{1:T}, x_{1:T} | \hat{\Theta})) \mid x_{1:T}, \Theta] \quad (6.18)$$

$$= \sum_{q_{1:T}} P(q_{1:T} | x_{1:T}, \Theta) \log(P(q_{1:T}, x_{1:T} | \hat{\Theta})) \quad (6.19)$$

$$= L^{-1} \sum_{q_{1:T}} P(q_{1:T}, x_{1:T} | \Theta) \log(P(x_{1:T}, x_{1:T} | \hat{\Theta})) \quad (6.20)$$

Specifically, the "E" step evaluates this expectations, and the "M" step finds the value of $\hat{\Theta}$ that maximizes this expectation. Hence if a hidden state space is properly chosen, then Equation 6.20 will be delineated into a summation-of-unknowns form. Then we can take the partial derivative of Equation 6.20, and solve each of the unknowns in closed form.

We compute in the E-step the auxiliary state-occupancy variables $\gamma_t(q)$ and the transition variables $\xi_t(q', q, d)$ using the forward-backward variables $\alpha_t(q)$ and $\beta_t(q)$:

$$\begin{aligned}\gamma_t(q) &\stackrel{\wedge}{=} P(q_t = q \mid x_{1:T}; \Theta) \\ &= L^{-1} \cdot \alpha_t(k)\beta_t(k)\end{aligned}\tag{6.21}$$

$$\begin{aligned}\xi_t(q', q, d) &\stackrel{\wedge}{=} P(q_t = q', q_{t+1} = q, e_t = d \mid x_{1:T}; \Theta) \\ &= L^{-1} \alpha_t(q') \tilde{a}(q', q, d) b_q(x_{t+1}) \beta_{t+1}(q) \\ &t = 1, \dots, T - 1\end{aligned}\tag{6.22}$$

Obviously these auxiliary variables shall normalize as:

$$\sum_k \gamma_t(k) = 1, \quad \sum_k \sum_{k'} \sum_d \xi_t(k', k, d) = 1, \quad \gamma_t(k') = \sum_k \sum_d \xi_t(k', k, d).$$

We then obtain a new estimate for each of the parameters in Equation 6.2 by marginalizing and normalizing the corresponding auxiliary variables, this will also maximize the current expected complete-data likelihood (Equation 6.18). Here we use $q = \overline{(rir')}$ and $q' = \overline{(rir'')}$ to represent two state configurations that only differ at the d^{th} digit or lower, and share the highest $(d - 1)$ digits r , i.e., q, q', r must satisfy the constraint $r = q^{1:d-1} = q'^{1:d-1}$. Each estimation equations below estimates the a transition parameter at level d , and $r' = q_t^{d+1:D}$, $r'' = q_{t+1}^{d+1:D}$ are used to represent state configurations at levels lower than d .

The prior probabilities:

$$\hat{\pi}_r^d(j) = \frac{\sum_{t=1}^{T-1} \sum_{r'} \sum_{r''} \sum_i \xi_t(\overline{(rir')}, \overline{(rjr'')}, d)}{\sum_{t=1}^{T-1} \sum_{r'} \sum_{r''} \sum_i \sum_j \xi_t(\overline{(rir')}, \overline{(rjr'')}, d)} \quad (6.23)$$

Within-level transition probability:

$$\hat{A}_r^d(i, j) = \frac{\sum_{t=1}^{T-1} \sum_{r'} \sum_{r''} \xi_t(\overline{(rir')}, \overline{(rjr'')}, d)}{\sum_{t=1}^{T-1} \sum_{r'} \sum_{r''} \sum_j \xi_t(\overline{(rir')}, \overline{(rjr'')}, d)} \quad (6.24)$$

Level-exiting probability:

$$\hat{e}_r^d(i) = \frac{\sum_{t=1}^{T-1} \sum_r \sum_{r'} \sum_{q'} \sum_{d' \leq d} \xi_t(\overline{(rir')}, q', d')}{\sum_{t=1}^{T-1} \sum_r \sum_{r'} \gamma_t(\overline{(rir')})} \quad (6.25)$$

Also note that for implementation purposes the temporal dimension of γ and ξ are always marginalized out, in the implementation we compute $\tilde{\gamma}(q) = \sum_t \gamma_t(q)$ and $\tilde{\xi}(q, q', d) = \sum_t \xi_t(q, q', d)$ as the sufficient statistics.

Denote each observation x_t as a n -dimensional row vector, the means and covariances of state q are easily obtained with the sufficient statistic $\tilde{\gamma}(q)$;

$$\hat{\mu}_q = \frac{\sum_{t=1}^T x_t \cdot \gamma_t(q)}{\sum_{t=1}^T \gamma_t(q)}, \quad \hat{\Sigma}_q = \frac{\sum_{t=1}^T x_t^T x_t \cdot \gamma_t(q)}{\sum_{t=1}^T \gamma_t(q)} \quad (6.26)$$

For the inference and estimation of a single HHMM model over U different observations sequences, we first compute the sufficient statistics $\gamma_t^u(q)$, $\xi_t^u(q, q', d)$ for each sequence u using the recipes above, we then perform the E-step of the algorithm (Equation 6.23-6.26) by summing over all sequences $u = 1, \dots, U$.

A.6 The complexity of learning and inference with HHMM

Denote T as the length of the observation sequence $x_{1:T}$. The multi-level hidden state inference of HHMM presented by Fine et. al. is $O(T^3)$ by looping over all possible lengths of subsequences generated by each Markov model at each level. Murphy and Paskin [84] later showed an $O(T)$ algorithm with an equivalent DBN representation by unrolling the multi-level states in time (Figure 3.2(b)). The inference scheme used in [84] is the generic junction tree algorithm for DBNs, and the empirical complexity is $O(DT \cdot Q^{1.5D})$ (or more accurately, $O(DT \cdot Q^{\lceil 1.5D \rceil} 2^{\lceil 0.5D \rceil})$), where D is the number of levels in the hierarchy, and Q is the maximum number of distinct discrete values of any variable q_t^d , $d = 1, \dots, D$.

In the algorithms presented above we use a generalized forward-backward algorithm for hidden state inference, and a generalized EM algorithm for parameter estimation based on the forward-backward iterations. This algorithm is chosen for its simple structure that enables a specialized fast implementation on the forward-backward iterations. The complexity of this algorithm is $O(DT \cdot Q^{2D})$, superior to the original $O(T^3)$ algorithm [41], and similar to [84] for small D and modest Q .

B Model adaptation for HHMM

This section presents the details for the stochastic search of the optimal model size in [section 3.4](#), these include computing the proposal probabilities, carrying out the move in the state space, and computing the acceptance ratio.

B.1 Proposal probabilities for model adaptation

Let p_{sp} , p_{me} , p_{sw} and p_{em} denote the probability of splitting, merging, swapping children states or staying put in the current state-space and perform EM update. These quantities are then computed based on the current number of states κ , the prior parameter ρ , and a simulation parameter c^* controlling how likely the model size would change:

$$p_{sp}(\kappa, d) = c^* \cdot \min\{1, \rho/(\kappa + 1)\}; \quad (6.27)$$

$$p_{me}(\kappa, d) = c^* \cdot \min\{1, (\kappa - 1)/\rho\}; \quad (6.28)$$

$$p_{sw}(\kappa, d) = c^*; \quad (6.29)$$

$$d = 1, \dots, D;$$

$$p_{em}(\kappa) = 1 - \sum_{d=1}^D [p_{sp}(\kappa, d) + p_{me}(\kappa, d) + p_{sw}(\kappa, d)]. \quad (6.30)$$

Note ρ is the hyper-parameter for the truncated Poisson prior of the number of states [\[7\]](#), i.e., the expected mean of the number of states if the maximum state size is allowed to be $+\infty$. ρ and κ modulate the proposal probability on the basis of c^* . Intuitively, the probability of splitting a node decreases if the number of nodes κ is already much larger than ρ .

B.2 Computing different moves in RJ-MCMC

If EM is selected according to the probability profile computed in [Equation 6.27-6.30](#), we perform one regular hill-climbing iteration as described in [section A](#). Otherwise we select with uniform probability one (or two) states at any level for swap/split/merge, and update the parameters accordingly:

- Swap the association of two states:

Choose two states from the same level, each of which belongs to a different higher-level state; swap their higher-level association.

- Split a state:

Choose a state at random. The split strategy differs when this state is at different position in the hierarchy:

- When this is a state at the lowest level ($d = D$), perturb the mean of its associated observation sequence distribution as follows (assume Gaussian)

$$\begin{aligned}\mu_1 &= \mu_0 + u_s \eta \\ \mu_2 &= \mu_0 - u_s \eta\end{aligned}\tag{6.31}$$

where $u_s \sim U[0, 1]$, and η is a simulation parameter that ensures reversibility between split moves and merge moves.

- When this is a state at $d = 1, \dots, D - 1$ with more than one children states, split its children into two disjoint sets at random, generate a new *sibling* state at level d associated with the same parent as the selected state. Update the corresponding multi-level Markov chain parameters accordingly.

- Merge two states:

Select two *sibling* states at level d , merge the observation probabilities or the corresponding *child-HHMM* of these two states, depending on which level they are located in the original HHMM:

- When $d = D$, merge the Gaussian observation probabilities by making the new mean as the average of the two.

$$\mu_0 = \frac{\mu_1 + \mu_2}{2}, \quad \text{if } |\mu_1 - \mu_2| \leq 2\eta \quad (6.32)$$

here η is the same simulation parameter as in .

- When $d = 1, \dots, D - 1$, merge the two states by making all the children of these two states the children of the merged state, and modify the multi-level transition probabilities accordingly.

B.3 The acceptance ratio for different moves in RJ-MCMC

When moves are proposed to a parameter space with different dimensions, such as a split or a merge, we will need to take two additional terms into account when evaluating the acceptance ratio [47]: (1) a proposal ratio term to ensure detailed balance. Intuitively this is the ratio of the reverse move to the proposed move, and it serves to balance the influence of the proposal probabilities and ensures that every point in the state space would be visited in probability, i.e., “rewarding” not-so-popular moves when they are proposed. Here $p(u_s)$ is taken to be the uniform distribution. (2) a Jacobian term to align the two spaces. As shown below:

$$r_\kappa \stackrel{\wedge}{=} (\text{posterior ratio}) \cdot (\text{proposal ratio}) \cdot (\text{Jacobian}) \quad (6.33)$$

$$r_{split} = \frac{P(\kappa + 1, \Theta_{\kappa+1}|x)}{P(\kappa, \Theta_\kappa|x)} \cdot \frac{p_{me}(\kappa + 1)/(\kappa + 1)}{p(u_s)p_{sp}(\kappa)/\kappa} \cdot J \quad (6.34)$$

$$r_{merge} = \frac{P(\kappa, \Theta_\kappa | x)}{P(\kappa + 1, \Theta_{\kappa+1} | x)} \cdot \frac{p(u_s)p_{sp}(\kappa - 1)/(\kappa - 1)}{p_{me}(\kappa)/\kappa} \cdot J^{-1} \quad (6.35)$$

$$J = \left| \frac{\partial(\mu_1, \mu_2)}{\partial(\mu_0, u_s)} \right| = \begin{vmatrix} 1 & \eta \\ 1 & -\eta \end{vmatrix} = 2\eta \quad (6.36)$$

Here $P(\kappa, \Theta_\kappa | x)$ is empirically computed as the BIC posterior score of the model *Theta* having κ nodes, $p_{sp}(\kappa)$ and $p_{ms}(\kappa)$ refer to the proposal probabilities as in [Equation 6.27](#) and [Equation 6.28](#) at the same level d (omitted since *split* or *merge* moves do not involve any change across levels).

The acceptance ratio for *Swap* can be simplified as the posterior ratio since κ did not change during the move:

$$r \stackrel{\wedge}{=} (\text{posterior ratio}) = \frac{\exp(\widehat{BIC})}{\exp(BIC)} \quad (6.37)$$

C Low-level visual features for sports videos

This section provides details for extracting the features used in [chapter 3](#). One specially designed color feature, dominant color ratio is discussed in [subsection C.1](#); three generic motion features, motion intensity, horizontal and vertical translation estimates are presented in [subsection C.2](#).

C.1 Dominant color ratio

Dominant color ratio, as the name suggests, is designed to capture the percentage of the dominant color in the frame. This simple feature is intuitively effective for videos captured with several cameras in a constrained location, such as sports programs, due to three reasons: (1) The dominant color exist across many types of sport. Many sports games are carried out in a court/field of fixed dimensions, the location of the players and the fields of view of the cameras are constrained. The court or field would then be one of the most frequent object across the entire program. Example sport include soccer, baseball, basketball, volleyball, tennis, badminton, American football, etc. (2) The use of the dominant color introduces invariance to lighting, location, and even the type of sport. For example, tennis courts can be grass, clay, hard, or synthetic; soccer fields in different location are likely to have different types of grass, or different types of lighting (day/night) during the game; no to mention that soccer field and basketball court are of different colors. Automatically learning the dominant color alleviates the burden of specifying the range of the color manually for each clip. (3) The amount of dominant ratio reflects the production style of sports videos and this in turn gives a useful clue for the state of the game. The amount of dominant color is a telltale sign of whether or not the camera's attention is on the field, and this often indicates whether or not there

are interesting happenings in the game (see [Figure 6.1](#)). Computing dominant color ratio involves two steps, i.e. learning the dominant color for each clip, and then use the learned definition for each clip to find the percentage of pixels of this color.



Figure 6.1: Dominant color ratio as an effective feature in distinguishing three kinds of views in soccer video. Left to right: global, zoom-in, close-up. Global view has the largest grass area, zoom-in has less, and close-ups has hardly any (including cutaways and other shots irrelevant to the game).

C.1.1 Adaptively learning the dominant color

Take soccer videos, for example, the grass color of the soccer field is the dominant color. The appearance of the grass color, however, ranges from dark green to yellowish green or olive, depending on the field condition and capturing device. Despite these factors, we have observed that within one game, the hue value in the HSV (Hue-Saturation-Value) color space is relatively stable despite lighting variations; hence, learning the hue value would yield a good definition of dominant color.

The dominant color is adaptively learned for each video clip, using the following cumulative color histogram statistic: 50 frames are drawn from the initial five minutes (an I/P frame pool of 3000) of the video, the 128-bin hue histogram is accumulated over all sample frames, and the peak of this cumulative hue histogram correspond to the dominant color. This experiment is repeated eight times, each with a different set of frame samples; two standard deviations below and above the mean of the peak hue value over the eight trials is taken as the range for grass

color in the current video clip; this definition will include 95.4% of the grass pixels, assuming the distribution of peak hue value is Gaussian. This definition of dominant color is specific enough to characterize variations across different games, yet relatively consistent to account for the small variations within one game. We have also performed this test for two soccer videos that comes from the same game, 30 minutes apart, and results indeed show that the difference of the mean hue values over time is smaller than the standard deviation within one clip.

C.1.2 The dominant color ratio

Once we can distinguish grass pixels vs. non-grass pixels in each frame, the feature dominant-color-ratio η is just computed as $\eta = |P_g|/|P|$, where $|P|$ is the number of all pixels, and $|P_g|$ is the number of grass pixels.

C.1.3 The effectiveness of dominant color ratio

Observations showed intuitions that relates the dominant color ratio η to the scale of view and in turn to the status of the game. Experiments in [134] showed accuracies of 80% to 92% in labeling the three kinds of views using adaptive thresholds, and accuracies 67.3% to 86.5% in segmenting play/breaks from the view labels using heuristic rules. While the results are satisfactory, it is worth noticing that (1) the scale of view is a semantic concept, and most of the errors in labeling views is due to model breakdown; for example, *zoom-in* is sometimes shot with a grass background, and the area of the grass is sometimes even larger than that of the global shots; (2) it is not sufficient to model temporal relationships between views and game state with heuristic rules, as most of the errors is caused by violation of the assumption that a play-break transition only happens upon the transition of views. On the other hand, shots and shot boundaries have similar drawbacks such as (1) shot boundaries

are neither aligned with the transitions of play/break nor with switches in the scale of view; (2) shot detectors tend to give lots of false alarms in this domain due to unpredictable camera motion and intense object motion. Hence, in our learning algorithms using HMM, each game state corresponds to a feature value distribution of a mixture of Gaussians, while the temporal transitions are modeled as a Markov chain.

Note the dominant color feature can be generalized to many other types of sports as a good indicator of game status such as baseball, American football, tennis, basketball, etc.

C.2 Motion features

In this section we present the estimation of three generic motion features from the MPEG-compressed domain. These features are fast to compute and often gives an informative estimate on constrained domains such as sports [60].

C.2.1 Motion intensity

Motion intensity m is computed as the average magnitude of the *effective* motion vectors in a frame.

$$m = \frac{1}{|\Phi|} \cdot \sum_{\Phi} \sqrt{v_x^2 + v_y^2}, \quad (6.38)$$

$$\Phi = \{ \textit{inter-coded macro-blocks} \} \quad (6.39)$$

and $v = (v_x, v_y)$ is the motion vector for each macro-block.

This measure of motion intensity gives an estimate of the gross motion in the whole frame, including object and camera motion. Moreover, motion intensity carries complementary information to the color feature, and it often indicates the se-

antics within a particular shot. For instance, a wide shot with high motion intensity often results from player motion and camera pan during a play; while a static wide shot usually occurs when the game has come to a pause. Here we directly estimate motion intensity from the compressed bitstream instead of the pixel domain, since MPEG motion vectors can approximate sparse motion field quite well, and accurate motion estimation in the pixel domain is more expensive yet hard to obtain good results.

C.2.2 Camera motion estimates

In addition to the overall motion intensity, we also compute a quick estimate of the camera translation (m_x, m_y) . This estimate is obtained from the compressed-domain motion vector field $\{v(i) = (v_x(i), v_y(i)), i \in \Phi\}$ (see Equation 6.38-6.39) based on the work of Tan et. al. [119]. This estimate assumes (1) The camera motion only contain zoom (with factor a s) and translation $u = (u_x, u_y)$, (2) Object motion is negligible compared to background motion, (3) The camera is far from the scene and the differences in depth are negligible.

Since the camera zoom and translation estimates tries to characterize the global change of motion field, it will inevitably use two motion fields adjacent in time to obtain this estimate. Denote with v and v' the compressed-domain motion field for two inter-coded frames (P-frame) at time t and $t + 1$, the estimate seeks values for (s, u) that minimized the translation objective function:

$$\min J(s, u) = \sum_{i \in \Phi} \|v'(i) - s(v(i) + u)\|^2 \quad (6.40)$$

If the assumptions (1)-(3) were perfect than the minimum of J would be zero, i.e., the two motion fields would match after compensated with the zoom and translation

factors. In reality, the minimum is attained at the following values of s and u :

$$s^* = \frac{\sum_i (v'(i) - \bar{v}')^T (v(i) - \bar{v})}{\sum_i \|v(i) - \bar{v}\|^2} \quad (6.41)$$

$$u^* = \frac{\bar{v}'}{s^*} - \bar{v} \quad (6.42)$$

Here \bar{v} and \bar{v}' are the mean of the motion vectors in the current frame and the previous frame, respectively. In our implementation we also eliminate outlier motion vectors that are more than three standard deviations from \bar{v} , since the block-based motion compensation tend to incur noisy large vectors in areas with almost no texture and very little motion.

Equation 6.41 and 6.42 gives closed-form estimates for the camera parameters. They can be quickly computed from the motion field of the video stream in much shorter than real-time, and in constrained domains such as sports, the estimates are fairly accurate.

References

- [1] R. Agrawal, T. Imielinski, and A. N. Swami, “Mining association rules between sets of items in large databases,” in *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, P. Buneman and S. Jajodia, Eds., Washington, D.C., 26–28 1993, pp. 207–216. [Online]. Available: citeseer.nj.nec.com/agrawal93mining.html
- [2] R. Agrawal and R. Srikant, “Fast algorithms for mining association rules in large databases,” in *20th International Conference on Very Large Data Bases, September 12–15, 1994, Santiago, Chile*, J. B. Bocca, M. Jarke, and C. Zaniolo, Eds. Los Altos, CA 94022, USA: Morgan Kaufmann Publishers, 1994, pp. 487–499. [Online]. Available: <http://www.vldb.org/dblp/db/conf/vldb/vldb94-487.html>
- [3] —, “Mining sequential patterns,” in *Eleventh International Conference on Data Engineering*, P. S. Yu and A. S. P. Chen, Eds. Taipei, Taiwan: IEEE Computer Society Press, 1995, pp. 3–14. [Online]. Available: citeseer.nj.nec.com/agrawal95mining.html
- [4] S. Amari, “Information geometry of the EM and em algorithms for neural networks,” *Neural Networks*, vol. 8, no. 9, pp. 1379–1408, 1995.
- [5] A. Amir, G. Iyengar, C.-Y. Lin, C. Dorai, M. Naphade, A. Natsev, C. Neti,

- H. Nock, I. Sachdev, J. Smith, Y. Wu, B. Tseng, and D. Zhang, “The IBM semantic concept detection framework,” in *TREC Video Retrieval Evaluation Workshop*, 2003.
- [6] C. Andrieu, N. de Freitas, and A. Doucet, “Robust full bayesian learning for radial basis networks,” *Neural Computation*, vol. 13, pp. 2359–2407, 2001. [Online]. Available: <http://www-sigproc.eng.cam.ac.uk/~ad2/journals.html>
- [7] C. Andrieu, N. de Freitas, A. Doucet, and M. I. Jordan, “An introduction to MCMC for machine learning,” *Machine Learning*, vol. 50, pp. 5–43, Jan. - Feb. 2003. [Online]. Available: <http://www.cs.ubc.ca/~nando/publications.html>
- [8] T. L. Bailey and C. Elkan, “Unsupervised learning of multiple motifs in biopolymers using expectation maximization,” *Mach. Learn.*, vol. 21, no. 1-2, pp. 51–80, 1995.
- [9] S. Baluja, V. Mittal, and R. Sukthankar, “Applying machine learning for high performance named-entity extraction,” *Computational Intelligence*, vol. 16, November 2000.
- [10] Y. Bengio and P. Frasconi, “An input output HMM architecture,” in *Advances in Neural Information Processing Systems*, G. Tesauro, D. Touretzky, and T. Leen, Eds., vol. 7. The MIT Press, 1995, pp. 427–434. [Online]. Available: citeseer.ist.psu.edu/bengio95input.html
- [11] J. A. Bilmes, “Graphical models and automatic speech recognition,” *Mathematical Foundations of Speech and Language Processing*, 2003.

- [12] A. Blum and S. Chawla, “Learning from labeled and unlabeled data using graph mincuts,” in *Proc. 18th International Conf. on Machine Learning*. Morgan Kaufmann, San Francisco, CA, 2001, pp. 19–26. [Online]. Available: citeseer.nj.nec.com/blum01learning.html
- [13] A. L. Blum and P. Langley, “Selection of relevant features and examples in machine learning,” *Artif. Intell.*, vol. 97, no. 1-2, pp. 245–271, 1997.
- [14] J. S. Boreczky and L. A. Rowe, “Comparison of video shot boundary detection techniques,” *Journal of Electronic Imaging*, vol. 5, pp. 122–128, apr 1996.
- [15] S. Boykin and A. Merlino, “Machine learning of event segmentation for news on demand,” *Commun. ACM*, vol. 43, no. 2, pp. 35–41, 2000.
- [16] M. Brand, N. Oliver, and A. Pentland, “Coupled hidden markov models for complex action recognition,” in *CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*. Washington, DC, USA: IEEE Computer Society, 1997, p. 994.
- [17] M. Brand, “Structure learning in conditional probability models via an entropic prior and parameter extinction,” *Neural Computation*, vol. 11, no. 5, pp. 1155–1182, 1999. [Online]. Available: citeseer.nj.nec.com/brand98structure.html
- [18] A. S. Bregman, *Auditory Scene Analysis: The Perceptual Organization of Sound*. MIT Press, 1994.
- [19] S. Brin, R. Motwani, and C. Silverstein, “Beyond market baskets: Generalizing association rules to correlations,” in *SIGMOD 1997, Proceedings ACM SIGMOD International Conference on Management of Data*,

- J. Peckham, Ed. Tucson, Arizona: ACM Press, May 1997, pp. 265–276. [Online]. Available: citeseer.nj.nec.com/brin97beyond.html
- [20] S. Brin, R. Motwani, J. D. Ullman, and S. Tsur, “Dynamic itemset counting and implication rules for market basket data,” in *SIGMOD 1997, Proceedings ACM SIGMOD International Conference on Management of Data*, J. Peckham, Ed. Tucson, Arizona: ACM Press, May 1997, pp. 255–264. [Online]. Available: citeseer.nj.nec.com/brin97dynamic.html
- [21] P. F. Brown, S. A. D. Pietra, V. J. D. Pietra, and R. L. Mercer, “The mathematics of statistical machine translation: Parameter estimation,” *Computational Linguistics*, vol. 19, no. 2, pp. 263–311, jun 1993.
- [22] G. A. Calvert and T. Thesen, “Multisensory integration: methodological approaches and emerging principles in the human brain,” *Journal of Physiology, Paris*, vol. 98, no. 1–3, pp. 191–205, 2004. [Online]. Available: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=pubmed&dopt=Abstract&list_uids=15477032&query_hl=3
- [23] G. A. Calvert, E. T. Bullmore, M. J. Brammer, R. Campbell, S. C. Williams, P. K. McGuire, P. W. Woodruff, S. D. Iversen, and A. S. David, “Activation of Auditory Cortex During Silent Lipreading,” *Science*, vol. 276, no. 5312, pp. 593–596, 1997. [Online]. Available: <http://www.sciencemag.org/cgi/content/abstract/276/5312/593>
- [24] S.-F. Chang, W. Chen, and H. Sundaram, “Semantic visual templates: linking visual features to semantics,” in *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, Chicago, IL, USA, 1998, pp. 531–535.

- [25] T.-S. Chua, S.-Y. Neo, K.-Y. Li, G. Wang, R. Shi, M. Zhao, and H. Xu, “Trecvid 2004 search and feature extraction task by nus pris,” in *TREC Video Retrieval Evaluation Proceedings*, 2004.
- [26] D. Chudova and P. Smyth, “Pattern discovery in sequences under a markov assumption,” in *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM Press, 2002, pp. 153–162.
- [27] F. R. K. Chung, *Spectral Graph Theory*. American Mathematical Society, 1997, vol. 92.
- [28] P. Churchland, V. S. Ramachandran, and T. J. Sejnowski, “A critique of pure vision,” in *Large-scale neuronal theories of the brain*, C. Koch and J. L. Davis, Eds. Bradford Books MIT Press, 1994.
- [29] B. Clarkson and A. Pentland, “Unsupervised clustering of ambulatory audio and video,” in *International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, 1999. [Online]. Available: citeseer.nj.nec.com/45626.html
- [30] M. Cooper, “Video segmentation combining similarity analysis and classification,” in *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*. New York, NY, USA: ACM Press, 2004, pp. 252–255.
- [31] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

- [32] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of the Royal Statistical Society B*, vol. 39, pp. 1–38, dec 8th 1977.
- [33] Y. Deng, B. Manjunath, C. Kenney, M. Moore, and H. Shin, “An efficient color representation for image retrieval,” *IEEE Transactions on Image Processing*, vol. 10, no. 1, pp. 140–147, 2001.
- [34] A. Doucet and C. Andrieu, “Iterative algorithms for optimal state estimation of jump Markov linear systems,” *IEEE Transactions of Signal Processing*, vol. 49, pp. 1216–1227, 2001.
- [35] L.-Y. Duan, M. Xu, T.-S. Chua, Q. Tian, and C.-S. Xu, “A mid-level representation framework for semantic sports video analysis,” in *MULTIMEDIA '03: Proceedings of the eleventh ACM international conference on Multimedia*. New York, NY, USA: ACM Press, 2003, pp. 33–44.
- [36] R. Duda, P. Hart, and D. Stork, *Pattern Classification*. Wiley, 2001.
- [37] P. Duygulu, K. Barnard, N. de Freitas, and D. A. Forsyth, “Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary,” in *ECCV*, 2002. [Online]. Available: citeseer.nj.nec.com/duygulu02object.html
- [38] J. G. Dy and C. E. Brodley, “Feature subset selection and order identification for unsupervised learning,” in *Proc. 17th International Conf. on Machine Learning*. Morgan Kaufmann, San Francisco, CA, 2000, pp. 247–254.
- [39] R. El-Yaniv, S. Fine, and N. Tishby, “Agnostic classification of markovian sequences,” in *NIPS '97: Proceedings of the 1997 conference on Advances in*

neural information processing systems 10. Cambridge, MA, USA: MIT Press, 1998, pp. 465–471.

- [40] R. Fergus, P. Perona, and A. Zisserman, “Object class recognition by unsupervised scale-invariant learning,” in *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, June 2003. [Online]. Available: citeseer.nj.nec.com/580536.html
- [41] S. Fine, Y. Singer, and N. Tishby, “The hierarchical hidden Markov model: Analysis and applications,” *Machine Learning*, vol. 32, no. 1, pp. 41–62, 1998.
- [42] W. T. Freeman and E. H. Adelson, “The design and use of steerable filters,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 9, pp. 891–906, 1991.
- [43] J.-L. Gauvain, L. Lamel, and G. Adda, “The LIMSI broadcast news transcription system,” *Speech Commun.*, vol. 37, no. 1-2, pp. 89–108, 2002.
- [44] M. H. Giard and F. Peronnet, “Auditory-Visual Integration during Multimodal Object Recognition in Humans: A Behavioral and Electrophysiological Study,” *J. Cogn. Neurosci.*, vol. 11, no. 5, pp. 473–490, 1999. [Online]. Available: <http://jocn.mitpress.org/cgi/content/abstract/11/5/473>
- [45] B. Gold and N. Morgan, *Speech and Audio Signal Processing: Processing and perception of speech and music*. Wiley, 2000.
- [46] Y. Gong, L. T. Sin, C. H. Chuan, H. Zhang, and M. Sakauchi, “Automatic parsing of tv soccer programs,” in *ICMCS '95: Proceedings of the International Conference on Multimedia Computing and Systems (ICMCS'95)*. Washington, DC, USA: IEEE Computer Society, 1995, p. 167.

- [47] P. J. Green, “Reversible jump Markov chain Monte Carlo computation and Bayesian model determination,” *Biometrika*, vol. 82, pp. 711–732, 1995.
- [48] J. Hafner, H. S. Sawhney, W. Equitz, M. Flickner, and W. Niblack, “Efficient color histogram indexing for quadratic form distance functions,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 7, pp. 729–736, 1995.
- [49] A. Hauptmann, “Towards a large scale concept ontology for broadcast video,” in *International Conference on Image and Video Retrieval (CIVR’04)*, Dublin City University, Ireland, July 2004, pp. 674–675.
- [50] T. Hofmann, “Probabilistic latent semantic indexing,” in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM Press, 1999, pp. 50–57.
- [51] B. K. P. Horn and B. G. Schunck, “Determining optical flow,” *Artificial Intelligence*, vol. 59, no. 1-2, pp. 81–87, Feb 1981.
- [52] A. Howard and T. Jebara, “Dynamical systems trees,” in *AUAI ’04: Proceedings of the 20th conference on Uncertainty in artificial intelligence*. Arlington, Virginia, United States: AUAI Press, 2004, pp. 260–267.
- [53] W. H.-M. Hsu, S.-F. Chang, C.-W. Huang, L. Kennedy, C.-Y. Lin, and G. Iyengar, “Discovery and fusion of salient multi-modal features towards news story segmentation,” in *SPIE Electronic Imaging*, January 2004.
- [54] M. Hu, C. Ingram, M. Sirski, C. Pal, S. Swamy, and C. Patten, “A hierarchical HMM implementation for vertebrate gene splice site prediction,” Dept. of Computer Science, University of Waterloo, Tech. Rep., 2000.

- [55] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih, “Image indexing using color correlograms,” in *CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*. Washington, DC, USA: IEEE Computer Society, 1997, p. 762.
- [56] S. Imai, “Cepstral analysis synthesis on the mel frequency scale,” in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '83.*, vol. 8, 1983, pp. 93–96.
- [57] Y. A. Ivanov and A. F. Bobick, “Recognition of visual activities and interactions by stochastic parsing,” *IEEE Transaction of Pattern Recognition and Machines Intelligence*, vol. 22, no. 8, pp. 852–872, August 2000.
- [58] A. Iyengar, M. S. Squillante, and L. Zhang, “Analysis and characterization of large-scale web server access patterns and performance,” *World Wide Web*, vol. 2, no. 1-2, pp. 85–100, 1999. [Online]. Available: citeseer.nj.nec.com/iyengar99analysis.html
- [59] A. K. Jain, M. N. Murty, and P. J. Flynn, “Data clustering: a review,” *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999. [Online]. Available: citeseer.nj.nec.com/jain99data.html
- [60] S. Jeannin and A. Divakaran, “Mpeg-7 visual motion descriptors,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 6, pp. 7720–7724, Jun 2001.
- [61] T. Jebara, R. Kondor, and A. Howard, “Probability product kernels,” *J. Mach. Learn. Res.*, vol. 5, pp. 819–844, 2004.

- [62] J. Jeon, V. Lavrenko, and R. Manmatha, “Automatic image annotation and retrieval using cross-media relevance models,” in *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval (SIGIR-03)*, Toronto, Canada, jul 28– aug –1 2003, pp. 119–126.
- [63] T. Joachims, “Transductive inference for text classification using support vector machines,” in *Proceedings of ICML-99, 16th International Conference on Machine Learning*, I. Bratko and S. Dzeroski, Eds. Bled, SL: Morgan Kaufmann Publishers, San Francisco, US, 1999, pp. 200–209. [Online]. Available: citeseer.nj.nec.com/joachims99transductive.html
- [64] M. I. Jordan and C. M. Bishop, “An introduction to graphical models,” to be published, 2003.
- [65] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. Prentice Hall, 2000.
- [66] J. Kender and M. Naphade, “Visual concepts for news story tracking: Analyzing and exploiting the nist trecvid video annotation experiment,” in *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, 2005.
- [67] L. Kennedy, A. Natsev, and S.-F. Chang, “Automatic discovery of query-class-dependent models for multimodal search,” in *ACM Multimedia*, Singapore, November 2005.
- [68] A. J. King and G. A. Calvert, “Multisensory integration: perceptual grouping by eye and ear,” *Current Biology*, vol. 11, pp. R322–R325, April

2001. [Online]. Available: <http://www.sciencedirect.com/science/article/B6VRT-430G2NG-H/2/57da3a59f23da8c2cc6e63bf88dc5995>
- [69] R. Kohavi, P. Langley, and Y. Yun, “The utility of feature weighting in nearest-neighbor algorithms,” in *Proc. European Conference on Machine Learning*, Prague, aug 25 1997.
- [70] D. Koller and M. Sahami, “Toward optimal feature selection,” in *International Conference on Machine Learning*, 1996, pp. 284–292.
- [71] C. E. Lawrence, S. F. Altschul, M. S. Boguski, J. S. Liu, A. F. Neuwald, and J. C. Wootton, “Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment,” *Science*, vol. 8, no. 262, pp. 208–14, October 1993.
- [72] N. D. Lawrence and B. Schölkopf, “Estimating a kernel Fisher discriminant in the presence of label noise,” in *Proc. 18th International Conf. on Machine Learning*. Morgan Kaufmann, San Francisco, CA, 2001, pp. 306–313. [Online]. Available: citeseer.nj.nec.com/lawrence01estimating.html
- [73] J. S. Liu, A. F. Neuwald, and C. E. Lawrence, “Bayesian models for multiple local sequence alignment and Gibbs sampling strategies,” *Journal of the American Statistical Association*, vol. 90, no. 432, pp. 1156–1170, 1995.
- [74] X. Liu and W. B. Croft, “Cluster-based retrieval using language models,” in *SIGIR '04: Proceedings of the 27th annual international conference on Research and development in information retrieval*. New York, NY, USA: ACM Press, 2004, pp. 186–193.
- [75] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins, “Text

- classification using string kernels,” *J. Mach. Learn. Res.*, vol. 2, pp. 419–444, 2002.
- [76] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [77] R. Luo, C.-C. Yih, and K. L. Su, “Multisensor fusion and integration: approaches, applications, and future research directions,” *IEEE Sensors Journal*, vol. 2, no. 2, pp. 107–119, 2002.
- [78] D. Marr, *Vision. A Computational Investigation into the Human Representation of Visual Information*. New York: Freeman, 1982.
- [79] H. McGurk and J. MacDonald, “Hearing lips and seeing voices,” *Nature*, vol. 264, no. 5588, 23–30 December 1976.
- [80] M. Meila and J. Shi, “Learning segmentation by random walks,” in *NIPS*, 2000, pp. 873–879. [Online]. Available: citeseer.nj.nec.com/meila01learning.html
- [81] —, “A random walks view of spectral segmentation,” in *AISTAT*, 2003. [Online]. Available: citeseer.nj.nec.com/388467.html
- [82] *Merriam-Webster Collegiate Dictionary*, 10th ed. Merriam-Webster Inc., 1998. [Online]. Available: <http://www.m-w.com/>
- [83] T. P. Minka, “A statistical learning/pattern recognition glossary,” 2005. [Online]. Available: <http://research.microsoft.com/~minka/statlearn/glossary/>

- [84] K. Murphy and M. Paskin, “Linear time inference in hierarchical HMMs,” in *Proceedings of Neural Information Processing Systems*, Vancouver, Canada, 2001.
- [85] I. J. Myung, V. Balasubramanian, and M. A. Pitt, “Counting probability distributions: Differential geometry and model selection,” *Proceedings of the National Academy of Sciences*, vol. 97, no. 21, pp. 11 170–11 175, 2000.
- [86] M. Naphade and T. Huang, “Discovering recurrent events in video using unsupervised methods,” in *Proc. Intl. Conf. Image Processing*, Rochester, NY, 2002.
- [87] H. Naphide and T. Huang, “A probabilistic framework for semantic video indexing, filtering, and retrieval,” *IEEE Transactions on Multimedia*, vol. 3, no. 1, pp. 141–151, 2001.
- [88] A. Nefian, L. Liang, X. Pi, X. Liu, and K. Murphy, “Dynamic bayesian networks for audio-visual speech recognition,” *EURASIP, Journal of Applied Signal Processing*, vol. 2002, no. 11, p. 1274, November 2002.
- [89] A. Ng, M. Jordan, and Y. Weiss, “On spectral clustering: Analysis and an algorithm,” in *Advances in Neural Information Processing Systems 14*, 2001. [Online]. Available: citeseer.nj.nec.com/ng01spectral.html
- [90] K. Nigam, A. K. McCallum, S. Thrun, and T. M. Mitchell, “Text classification from labeled and unlabeled documents using EM,” *Machine Learning*, vol. 39, no. 2/3, pp. 103–134, 2000. [Online]. Available: citeseer.nj.nec.com/nigam99text.html

- [91] N. Oliver, E. Horvitz, and A. Garg, “Layered representations for learning and inferring office activity from multiple sensory channels,” in *Proceedings of Int. Conf. on Multimodal Interfaces (ICMI’02)*, Pittsburgh, PA, October 2002.
- [92] S. Paek and S.-F. Chang, “Experiments in constructing belief networks for image classification systems,” in *invited paper to IEEE International Conference on Image Processing (ICIP-2000)*, Vancouver, British Columbia, Canada, September 2000.
- [93] P. Pérez, J. Vermaak, and A. Blake, “Data fusion for visual tracking with particles,” *Proceedings of IEEE (special issue on State Estimation)*, vol. 292, no. 3, pp. 495–513, 2004.
- [94] J. Platt, “Fast training of support vector machines using sequential minimal optimization,” in *Advances in Kernel Methods — Support Vector Learning*, B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds. Cambridge, MA: MIT Press, 1999, pp. 185–208.
- [95] M. Porat and Y. Y. Zeevi, “The generalized gabor scheme of image representation in biological and machine vision,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 10, no. 4, pp. 452–468, 1988.
- [96] M. F. Porter, “An algorithm for suffix stripping,” in *Readings in information retrieval*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997, pp. 313–316.
- [97] Y. Qi, A. Hauptmann, and T. Liu, “Supervised classification for video shot segmentation,” in *International Conference on Multimedia and Expo (ICME)*, Baltimore, MD, July 2003.

- [98] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*. Pearson Education POD, 1993.
- [99] L. R. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–285, feb 1989.
- [100] R. Radhakrishnan, Z. Xiong, A. Divakaran, and Y. Ishikawa, “Generation of sports highlights using a combination of supervised and unsupervised learning in audio domain,” in *Proc. Pacific Rim Conference on Multimedia*, 2003.
- [101] R. Radhakrishnan, “A content-adaptive analysis and representation framework for summarization using audio cues,” Ph.D. dissertation, Polytechnic University, december 2004.
- [102] D. Reynolds and R. Rose, “Robust text-independent speaker identification using gaussianmixture speaker models,” *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 1, pp. 72–83, 1995.
- [103] Y. Rui, A. Gupta, and A. Acero, “Automatically extracting highlights for tv baseball programs,” in *MULTIMEDIA '00: Proceedings of the eighth ACM international conference on Multimedia*. New York, NY, USA: ACM Press, 2000, pp. 105–115.
- [104] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. New York, NY, USA: McGraw-Hill, Inc., 1986.
- [105] E. Scheirer and M. Slaney, “Construction and evaluation of a robust multifeature speech/music discriminator,” in *Proc. ICASSP '97*, Munich,

- Germany, 1997, pp. 1331–1334. [Online]. Available: citeseer.nj.nec.com/scheirer97construction.html
- [106] G. Schwarz, “Estimating the dimension of a model,” *The Annals of Statistics*, vol. 7, pp. 461–464, 1978.
- [107] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000. [Online]. Available: citeseer.nj.nec.com/shi97normalized.html
- [108] M. Slaney, “A critique of pure audition,” in *working notes of the workshop on Comp. Aud. Scene Analysis at the Intl. Joint Conf. on Artif. Intel.*, Montreal, 1995, pp. 13–18.
- [109] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, “Content-based image retrieval at the end of the early years,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 12, pp. 1349–1380, 2000.
- [110] J. R. Smith, “Integrated spatial and feature image systems: Retrieval, analysis and compression,” Ph.D. dissertation, Graduate School of Arts and Sciences, Columbia University, 1997.
- [111] P. Smyth, “Clustering sequences with hidden markov models,” in *Advances in Neural Information Processing Systems*, M. C. Mozer, M. I. Jordan, and T. Petsche, Eds., vol. 9. The MIT Press, 1997, p. 648. [Online]. Available: citeseer.nj.nec.com/smyth97clustering.html
- [112] Y. Song, L. Goncalves, and P. Perona, “Unsupervised learning of human motion,” *PAMI*, vol. 25, no. 7, pp. 814–827, July 2003.

- [113] R. Srikant and R. Agrawal, “Mining generalized association rules,” in *VLDB '95: proceedings of the 21st International Conference on Very Large Data Bases, Zurich, Switzerland, Sept. 11–15, 1995*, U. Dayal, P. M. D. Gray, and S. Nishio, Eds. Los Altos, CA 94022, USA: Morgan Kaufmann Publishers, 1995, pp. 407–419. [Online]. Available: <http://www.vldb.org/dblp/db/conf/vldb/SrikantA95.html>
- [114] M. A. Stricker and M. Orengo, “Similarity of color images,” in *Proceedings of SPIE, Storage and Retrieval for Image and Video Databases III*, vol. 2420, March 1995, pp. 381–392.
- [115] G. Sudhir, J. C. M. Lee, and A. K. Jain, “Automatic classification of tennis video for high-level content-based retrieval,” in *CAIVD '98: Proceedings of the 1998 International Workshop on Content-Based Access of Image and Video Databases (CAIVD '98)*. Washington, DC, USA: IEEE Computer Society, 1998, p. 81.
- [116] H. Sundaram, “Segmentation, structure detection and summarization of multimedia sequences,” Ph.D. dissertation, Graduate School of Arts and Sciences, Columbia University, august 2002.
- [117] H. Sundaram and S.-F. Chang, “Determining computable scenes in films and their structures using audio-visual memory models,” in *Proceedings of the eighth ACM international conference on Multimedia*. ACM Press, 2000, pp. 95–104.
- [118] M. Szummer and T. Jaakkola, “Partially labeled classification with markov random walks,” in *Advances in Neural Information Processing Systems*, vol. 14, 2001. [Online]. Available: citeseer.nj.nec.com/szummer02partially.html

- [119] Y.-P. Tan, D. D. Saur, S. R. Kulkarni, and P. J. Ramadge, “Rapid estimation of camera motion from compressed video with application to video annotation,” *IEEE Transactions on Circuits and Systems for Video Technology (CSVT)*, vol. 10, no. 1, pp. 133–146, Feb 2000.
- [120] The HTK Team, “Hidden Markov model toolkit (HTK3),” September 2000, <http://htk.eng.cam.ac.uk/>.
- [121] The National Institute of Standards and Technology (NIST), “Topic detection and tracking (TDT),” 1998–2004, <http://www.nist.gov/speech/tests/tdt/>.
- [122] —, “TREC video retrieval evaluation,” 2001–2004, <http://www-nlpir.nist.gov/projects/trecvid/>.
- [123] The Oxford University Press, “The Oxford English Dictionary,” 1857–2004, <http://dictionary.oed.com/>.
- [124] the Wikimedia Foundation, 2005, http://meta.wikimedia.org/wiki/Stop_word_list.
- [125] H. Toivonen, “Sampling large databases for association rules,” in *Proceedings of the twenty-second international Conference on Very Large Data Bases, September 3–6, 1996, Mumbai (Bombay), India*, T. M. Vijayaraman, A. P. Buchmann, C. Mohan, and N. L. Sarda, Eds. Los Altos, CA 94022, USA: Morgan Kaufmann Publishers, 1996, pp. 134–145. [Online]. Available: <http://www.vldb.org/dblp/db/conf/vldb/Toivonen96.html>
- [126] S. Tong and E. Chang, “Support vector machine active learning for image retrieval,” in *MULTIMEDIA '01: Proceedings of the ninth ACM international conference on Multimedia*. New York, NY, USA: ACM Press, 2001, pp. 107–118.

- [127] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, 2001.
- [128] Y. Wang, Z. Liu, and J. Huang, "Multimedia content analysis using both audio and visual clues," *IEEE Signal Processing Magazine*, vol. 17, no. 6, pp. 12–36, November 2000.
- [129] R. M. Warren, *Auditory Perception*. Cambridge, UK: Cambridge University Press, 1999.
- [130] M. Weber, M. Welling, and P. Perona, "Unsupervised learning of models for recognition," in *ECCV (1)*, 2000, pp. 18–32. [Online]. Available: citeseer.nj.nec.com/weber00unsupervised.html
- [131] L. Xie, S.-F. Chang, A. Divakaran, and H. Sun, "Structure analysis of soccer video with hidden Markov models," in *Proc. Interational Conference on Acoustic, Speech and Signal Processing (ICASSP)*, Orlando, FL, 2002.
- [132] L. Xie, P. Xu, S.-F. Chang, A. Divakaran, and H. Sun, "Structure analysis of soccer video with domain knowledge and hidden markov models," *Pattern Recogn. Lett.*, vol. 25, no. 7, pp. 767–775, 2004.
- [133] E. P. Xing and R. M. Karp, "Cliff: Clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts," in *Proceedings of the Ninth International Conference on Intelligence Systems for Molecular Biology (ISMB)*, 2001, pp. 1–9. [Online]. Available: <http://www.cs.berkeley.edu/~epxing/Paper/ismb-cliff.ps>

- [134] P. Xu, L. Xie, S.-F. Chang, A. Divakaran, A. Vetro, and H. Sun, “Algorithms and systems for segmentation and structure analysis in soccer video,” in *Proc. IEEE International Conference on Multimedia and Expo (ICME)*, Tokyo, Japan, 2001.
- [135] R. Yan, J. Yang, and A. G. Hauptmann, “Learning query-class dependent weights in automatic video retrieval,” in *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*. New York, NY, USA: ACM Press, 2004, pp. 548–555.
- [136] S. Yantis, *Key Readings in Visual Perception*. Psychology Press, 2000.
- [137] D. Zhang, R. Rajendran, and S.-F. Chang, “General and domain-specific techniques for detecting and recognizing superimposed text in video,” in *Image Processing. 2002. Proceedings. 2002 International Conference on*, vol. 1, 2002.
- [138] H. Zhang, A. Kankanhalli, and S. W. Smoliar, “Automatic partitioning of full-motion video,” *Multimedia Syst.*, vol. 1, no. 1, pp. 10–28, 1993.
- [139] D. Zhong and S.-F. Chang, “An integrated approach for content-based video object segmentation and retrieval,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 8, pp. 1259–1268, December 1999.