

Probabilistic and Film Grammar Based Methods for Video Content Understanding

by

Dinh Quoc Phung
(Phùng Quốc Định)

Submitted to
the Department of Computing Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at

Department of Computing Science
CURTIN UNIVERSITY OF TECHNOLOGY

January 2005

Copyright© 2005 Curtin University of Technology

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university. To the best of my knowledge and belief this thesis contains no material previously published by any other person except where due acknowledge has been made.

Contents

Abstract	xi
Acknowledgements	xiii
Relevant Publications	xiv
Abbreviations	xvi
Notation	xvii
1 Introduction	1
1.1 Aims and Approach	2
1.2 Significance and Contribution	4
1.3 Structure of the Thesis	6
2 Related Background	8
2.1 Video Content Analysis: An Overview	8
2.1.1 Parsing and Segmentation of Videos	9
2.1.1.1 Primitive feature extraction	10
2.1.1.2 Temporal segmentation	14
2.1.2 Video Representation and Summarisation	18
2.1.3 Video Searching and Retrieval	20
2.1.4 Current Challenge – the Semantic Gap	21
2.2 Understanding Video Content through Film Grammar	22
2.2.1 Film Grammar for Educational Videos	22
2.2.2 Video Annotation with Film Grammar	24
2.2.3 Video Segmentation with Film Grammar	25
2.2.4 Computational Media Aesthetics and High Order Construct Extrac- tion	26
2.3 Probabilistic Approaches to Video Content Analysis	30
2.3.1 Bayesian Network Approaches to Video Analysis	30
2.3.2 Bayesian Networks and d -Separation	31
2.3.2.1 Bayesian Network approaches to scene content annotation .	33

2.3.2.2	Bayesian extraction of semantic concepts	34
2.3.2.3	Bayesian framework for video retrieval	35
2.3.3	Hidden Markov Model Based Approaches to Video Analysis	37
2.3.3.1	Hidden Markov Models	38
2.3.3.2	HMM-based video classification	40
2.3.3.3	HMM based video segmentation	42
2.3.3.4	Using a hierarchy of Hidden Markov Models	43
2.3.3.5	Hierarchical Hidden Markov Models	44
2.4	Closing Remarks	47
3	Identification of Hierarchical Narrative Structural Units	48
3.1	Understanding the Educational Film Genre	49
3.2	Uncovering the Hidden Structural Units	51
3.2.1	On-screen Narration	51
3.2.2	Voice-over Narration	53
3.2.3	Linkage Narration	53
3.2.4	Supportive Narration	54
3.3	Feature Extraction	55
3.3.1	Visual Content Analysis	55
3.3.1.1	Face detection and associated features	55
3.3.1.2	Text detection and associated features	57
3.3.2	Audio Content Analysis	59
3.4	Constructing the Classification System	61
3.4.1	Data and Groundtruth	61
3.4.2	Results: One-Layer Classification	61
3.4.3	Results: Two-Tiered Classification	63
3.5	Closing Remarks	64
4	Expressive Functions and Segmentation of Topical Content	65
4.1	The Nature of Topics and Subtopics	66
4.2	Partitioning Videos into Subtopics	67
4.2.1	The Content Density Function	67
4.2.2	Subtopic Boundary and the Curvature of the $\mathbf{Dn}(\cdot)$ Function	69
4.2.3	Heuristic Approach to Subtopic Detection	72
4.2.4	Probabilistic Approach to Subtopic Detection	72
4.2.5	Experimental Results	74
4.2.5.1	Results from the Heuristic Approach	74
4.2.5.2	Results from the Probabilistic Approach	75
4.3	Thematic and Dramatic Functions of Video Content	77
4.3.1	The Thematic Function	78
4.3.2	The Dramatic Function	79

4.3.3	Relating Media Elements to Thematic and Dramatic Functions . . .	79
4.3.4	Experimental Results	82
4.4	Hierarchically Partitioning Videos into Topics	84
4.4.1	Experimental Results	84
4.5	Concluding Remarks	87
5	Hierarchical Hidden Markov Models with Shared Structures	89
5.1	The Hierarchical HMM: Intuition and Definition	91
5.1.1	From the Regular HMM to the Hierarchical HMM	91
5.1.2	Hierarchical HMMs: Model Definition	93
5.1.3	Hierarchical HMMs with Shared Substructures	95
5.2	DBN Representation for the Hierarchical HMM	97
5.2.1	Network construction	97
5.2.1.1	Assumptions in the DBN Structure	97
5.2.1.2	Construction of the first slice	98
5.2.1.3	Construction at time t	99
5.2.1.4	The full DBN structure	100
5.2.2	Mapping Parameters from the DBN Structure	102
5.2.3	Conditional Independence in the DBN structure of the HHMM . . .	103
5.2.3.1	Symmetric independence theorem	103
5.2.3.2	Asymmetric independence theorem	105
5.2.3.3	The Start-to-End (STE) and Started-Idp (SI) lemmas . . .	106
5.3	Sufficient Statistics and Maximum-Likelihood for the HHMM	107
5.3.1	The Exponential Family	108
5.3.2	DBN as an Exponential Density and the Sufficient Statistics	108
5.3.3	Maximum-Likelihood Estimation	111
5.4	Expected Sufficient Statistics and EM Estimation	112
5.4.1	Expectation-Maximisation Algorithm	113
5.4.2	The Complete Log-likelihood Function and the M-step	114
5.4.3	Expected Sufficient Statistics and Auxiliary Variables	115
5.5	Inference in the Hierarchical HMM	117
5.5.1	The Fine-Singer-Tishby (FST) Method	117
5.5.2	The Asymmetric Inside-Outside Algorithm	119
5.5.2.1	The set of inside/outside auxiliary variables	120
5.5.2.2	Calculating the horizontal probability variables	121
5.5.2.3	Calculating the vertical-transition/emission probability variables	125
5.5.3	Computation of Other Auxiliary Variables	126
5.5.3.1	Calculating the asymmetric forward inside variable	126
5.5.3.2	Calculating the started-forward variable	126

5.5.3.3	Calculating the symmetric inside variable	127
5.5.3.4	Calculating the symmetric/asymmetric outside variables	128
5.6	Computing the Likelihood	130
5.7	Complexity Analysis and Some Numerical Results	131
5.7.1	Complexity of the AIO-algorithm	132
5.7.2	Some Numerical Results	132
5.8	Numerical Underflow and the Scaling Algorithm	134
5.8.1	Calculating the scaled inside variables	137
5.8.1.1	Calculating the scaling factor φ_r	138
5.8.2	Calculating the scaled outside variables	139
5.9	Closing Remarks	140
6	Semantic Analysis with the Hierarchical HMM	141
6.1	The Generalised Viterbi Algorithm for the HHMM	142
6.2	Modeling Continuous Observations	146
6.2.1	Mixture of Gaussians Emission Probability	146
6.2.2	Parameter Estimation	147
6.3	Subtopic Boundaries Detection with the HHMM	150
6.3.1	Incorporating Prior Knowledge into the Topology	150
6.3.2	Training the Model	151
6.3.3	Subtopic Detection Results and Discussion	153
6.4	Automatically Learning Structural Units with the HHMM	155
6.4.1	Semantic Mappings at the Production Level	156
6.4.2	Semantic Mappings at the Upper Level	157
6.5	Closing Remarks	158
7	Conclusions	159
7.1	Future Directions	161
A	Proof of Selected Theorems	164
B	Proofs of Formulas	165
C	Sufficient Statistics via Parameter Tying Transformation	175
C.0.1	Tying Transformation Theorem in the Exponential Family	176
C.0.2	Tying Parameters in the DBN Structure of the HHMM	176
C.0.2.1	The CSI-transformation	177
C.0.2.2	The Time-transformation	178
C.0.3	Sufficient Statistics in the BN (θ_{BN})	178

List of Figures

2-1	A typical structural decomposition of a video.	9
2-2	Steps in the NN-based face detection algorithm proposed in (Rowley <i>et al.</i> , 1998).	11
2-3	Steps in the text detection algorithm proposed in (Shim <i>et al.</i> , 1998)	12
2-4	A typical decomposition of an audio track in a movie.	13
2-5	The Computational Media Aesthetics framework	27
2-6	Rules for bouncing a ball in the Baye’s Ball algorithm	33
2-7	The Bayesian Network used in (Vasconcelos and Lippman, 1998c)	33
2-8	The simple BN used in(Ferman and Tekalp, 1999) to track the concept of ‘focus of attention’	35
2-9	The Bayesian Network used in Vasconcelos and Lippman (1998) to represent the relationship between the observations (feature variables X^k) and the sources (source content variables S_i) in the database.	36
2-10	The Dynamic Bayesian Network representation of a Hidden Markov Model.	38
3-1	The hierarchy of proposed narrative structures for educational videos. . . .	52
3-2	On-screen narration sub-hierarchy and examples.	53
3-3	Examples of voice-over narrative structures.	54
3-4	Examples of linkage and supportive narrative structures.	54
3-5	<i>Face-Content-Ratio</i> features (FCR) plotted for AN _{wS} , DN, VOTS, VO _{wT} , and E _{xLK} (values are sorted).	56
3-6	Coordinates (X_ϕ, Y_ϕ) of bounding box center are used as features to encode the movement of the narrator.	57
3-7	Example of FAR features plotted for Direction Narration vs. Assisted Narration.	58
3-8	Examples of scene texts (left) and superimposed texts (right). Scene texts are not considered in our work.	58
3-9	Examples of TCR features plotted within different structural categories. . .	58
3-10	Averaged values of the speech-ratio feature.	60
3-11	Averaged values of music-ratio features.	60

4-1	Content density functions (original and smoothed versions) plotted for educational video ‘Against All Odds - Part 4’ and training video ‘Eye-Safety’.	70
4-2	In the heuristic approach, the mid-point between C_{left} and C_{right} is labeled as the temporal index of a subtopic boundary.	71
4-3	The density function and likelihood for the video, ‘House-Keeping’. Vertical solid lines are actual subtopic boundaries and vertical dashed lines are detected boundaries.	73
4-4	Average precision and recall values for various window sizes computed from a set of 10 videos.	74
4-5	Magnitude of a candidate shot $\Phi_{c(i)}$ is computed as $\frac{p_1}{p_1+p_2}$	75
4-6	Establishing relationships between proposed media elements and their impact on thematic and dramatic nature of the educational content.	81
4-7	Analysis of thematic and dramatic functions for video ‘Electronics-Safety’.	83
4-8	Groundtruth and detection results for the ‘Eye-Safety’ video.	85
4-9	Plots of the thematic function, detected edges, topic and subtopic boundaries for instructional video ‘Eye-Safety’.	86
5-1	Representing a HMM with three states as a finite state automata (a), and its corresponding topological structure (b).	92
5-2	The HMM in Figure-(5-1) can be considered as children of a higher abstract state P	92
5-3	State transition and corresponding topology for the HHMM described in Example-(5.1) and Example-(5.2).	93
5-4	State transition and corresponding topology for the HHMM described in Example-(5.4)	96
5-5	A fully shared four-level HHMM (lattice) topology, and its equivalent expanded tree topology (when shared structure are not modeled).	96
5-6	Network construction at the first slice.	98
5-7	Subnetworks for the dependence structure during a transition	99
5-8	DBN representation for the HHMM, shaded nodes are observed variables. By definition, at the bottom level e_t^D is fixed to 1 and thus is removed from the network.	101
5-9	Four types of ‘family’ $\{z, \pi_z\}$ identified in the DBN of the HHMM.	102
5-10	Sub-network configuration for the event ‘start’ and ‘end’ of a state $x_t^d = p$.	103
5-11	Symmetric boundary event $SB_{t:r}^{d,p}$	104
5-12	Illustration of the asymmetric boundary event	105
5-13	Graphical representation for Lemma 5.1.	106
5-14	Graphical representation for Lemma 5.2.	107
5-15	Decomposition of the auxiliary variable $\xi_t^{d,p}(i, j)$ in the FST method.	118

5-16	Diagrammatic forms of recursive partitions for $\alpha_{l:r}^{d,p}(i)$ in Equation-(5.46b) when r strictly greater than l	119
5-17	Decomposition of the auxiliary variable $\xi_t^{d,p}(i, j)$ in our AIO (Asymmetric Inside-Outside) method.	120
5-18	Diagrams for the inside and outside variables	121
5-19	The HHMM used in the report of (Schlick, 2000) and in our experiment.	133
5-20	The training curve after 20 iterations.	133
5-21	Computation time of MP method (Murphy and Paskin, 2001) versus our proposed Asymmetric Inside-Outside (AIO) method.	134
6-1	Diagrammatic visualisation for the set of ‘maximum’ variables.	143
6-2	Visualisation of the recursive structure for routine <code>find-max-config(l, r, d, p)</code>	145
6-3	Graphical representation of the lowest level in the DBN structure with added mixture component z_t	147
6-4	Structure of subtopic generating process with assumed hidden ‘styles’; and its mapping to a topology for the HHMM. Shared structures are identified with an extra dotted circle.	151
6-5	Re-estimated parameters after training for the model at the top two layers.	152
6-6	Visualisation of the re-estimated mean value $\hat{\mu}_{21}$ for ‘introduction state’.	153
6-7	Further insight into the structure of an educational video.	154
6-8	The hierarchy of connected concepts which is used as topological specification for the corresponding HHMM.	155
6-9	Gray-scaled visualisation of the mean values of seven features learned at the production level.	156
6-10	Estimated transition probability at the upper level. Only dominant probabilities are shown.	157
C-1	Four main types of cliques identified in the DBN structure of the HHMM, which is essentially the same as Figure-(5-9).	178

List of Tables

2.1	The set of aural features described in (Phung, 2001) and used in this thesis.	14
3.1	Statistics of shots in the groundtruth for each structural type.	61
3.2	Results from evaluation on the training data (top) and 10-fold cross validation (bottom).	62
3.3	Classification results at each level of the hierarchy.	63
4.1	Average results for subtopic detection with probabilistic (\mathcal{P}^*) and heuristic (\mathcal{E}^*) approaches using a window size = 70s.	76
4.2	Aesthetics elements for expression in general films suggested by Foss (1992), and our suggested mappings to educational films.	77
4.3	Media elements and their utility in conveying thematic and dramatic functions of content in educational video.	78
4.4	Associated features for media elements in Table-(4.3).	80
4.5	Detection results of extrema points for dramatic and thematic functions. . .	82
4.6	Detection results for a set of ten education and training videos.	86
5.1	Parameters definition for a HHMM at level d for $1 \leq d \leq D - 1$	94
5.2	The initial parameters at the root level, and their estimated results from (Schlick, 2000) and from our experiment.	133
6.1	Detection results.	154
6.2	Deduced semantic mappings at production (shot) level.	157

List of Algorithms

4.1	Heuristic approach to subtopic detection	71
4.2	Two-tiered hierarchical topical segmentation	85
5.1	ML-estimation for the HHMM in the fully observed data case	112
5.2	EM estimation for the HHMM	115
5.3	Pseudo codes to compute the set of inside variables	127
5.4	Pseudo codes to compute the set of outside variables	129
5.5	The Asymmetric Inside-Outside Algorithm for the HHMM	130
5.6	Calculating the set of scaled inside variables	139
6.1	Backtracking for Generalised Viterbi.	145
6.2	Sub-routine $\{\mathbf{s}^*, \boldsymbol{\tau}^*\} = \text{find-max-config}(l, r, d, p)$	146

Abstract

In the last decade, we have truly entered the information age with an explosion in the amount of digital data produced every day. The growing complexity and inadequacy of existing tools for managing this data proliferation have highlighted the need for better tools and techniques. Towards this end, this thesis aims at exploring Film Grammar and formal probabilistic models for video content analysis, in which the educational videos are used as the domain of investigation.

To utilise Film Grammar, we base our work in a Computational Media Aesthetics framework as a systematic way of harnessing film theory in seeking meaningful descriptors. We study the aspects of ‘grammars’ that are peculiar to the educational genre to uncover the nature of semantics and structural information presented. In our initial study, a hierarchy of narrative structural units for this video genre is proposed and automatically recognised. The hierarchy is learned using a set of features extracted from audio and visual streams. The experimental results demonstrate the usefulness of the proposed hierarchy.

Next, we exploit film theory to extract *expressive* elements, computed from low-level features, that are useful for educational videos. First, we examine the *content density* function as a measure of the ‘rate of information delivered’, study its key contributing factors and propose a computational form for it. Observing that a drop or rise in this function reflects important clues about the structure of the video, we propose heuristic and probabilistic algorithms for the detection of subtopic boundaries. We then advance this study with the extraction of the *dramatic* and *thematic* functions. The thematic function reflects the ‘instructional’ or ‘informative’ portions in the video where the video-maker decides to interfere in the subject matter being presented. The dramatic function, on the other hand, provides information about the ‘dramatisation level’ of the video, such as a film segment where an event is dramatised. Combining information from the content density and thematic functions allows us to further segment an educational video into a two-level hierarchy of topical content, namely at main topic and subtopic levels.

In seeking formal probabilistic models, our key observation is that semantic concepts in the video domain possess a natural hierarchical decomposition, and more noticeably there is *tight* inheritance of semantics in the hierarchy. Here, we theoretically extend the tree-structure Hierarchical Hidden Markov Models (HHMM) in (Fine *et al.*, 1998)

to allow arbitrary sharing at any level in the topology of the model. We show how the exact inference and learning can be done in this general case with the same complexity as in (Fine *et al.*, 1998). To deal with long observation sequences, we propose a novel scaling algorithm to avoid numerical underflow. In addition, we also propose a new generalised Viterbi algorithm and address the issue of continuous observations for this model.

Following the theoretical extension to the HHMM, we present two applications exploiting the expressiveness of shared structures in the HHMM for the problem of semantic analysis and segmentation in educational videos. First, it is shown that subtopic boundary transitions can be detected in this framework. Second, we show that useful narrative structures can be learned automatically with the HHMM. In both applications, the domain knowledge is utilised as prior information to construct the topology of the HHMM.

Acknowledgements

For me, the last $3\frac{1}{2}$ years were, and will remain, unique and unforgettable. Not only the joy of working, but also the pressures especially from non-academic matters, added even more meaning to this thesis. In my deepest appreciation, I would like to thank my principal supervisor, Prof. Svetha Venkatesh, for her tireless guidance, encouragement and support that kept me going till the end. I find in her a *rare* quality that goes beyond a normal supervisor that one would imagine. I am thankful to whichever ‘magic’ made me fortunate enough to meet and work with her.

I also would like to thank my co-supervisor, Dr. Chitra Dorai, for her help during the course of this thesis. Although geographically far removed, email exchanges with her have helped in shaping many ideas during the initial period of this thesis. To my second co-supervisor, Dr Hung Bui, I would like express my deep gratitude for introducing me to the probabilistic world. His expertise in graphical models and his sharpness in thinking have both proven to be invaluable to me. I also thank him for the guidance that shaped the theoretical work in the Hierarchical Hidden Markov Models.

I am very grateful to the members of the CMA group, friends and the staff at the Department of Computing. Engaging in discussion with Brett, BaTu, and Simon and exchanging ideas on the topic of Film Grammar have been beneficial to me. Also, my thanks goes to Brett and BaTu for sharing their code that I have used in the initial phase of the thesis; to Sarah, Simon and Mary for helping out with the English; to all other fellows in the Department for enriching my life at Curtin or simply for chatting or having coffee/meals. I also would like to thank Curtin University and the Department of Computing for providing financial support for this thesis.

To my dearest wife Van Thi, I thank her for her love, understanding and endless support; for providing me with a warm shelter, the feeling of peace when you are around or simply in my mind; and of course, for the delicious food. My thanks is also extended to my Aunty Kha’s family and my brother Viet for making me feel that Australia is not so different from the hometown.

Finally, but surely not lastly, I’m indebted to my parents – a ‘debt’ that simple written words would by no means adequately describe. Even though half a world apart, their care and love can still be felt as if they are around. To them, this thesis is dedicated.

Perth, WA, January 2005

Relevant Publications

Part of this thesis and some related work has been published or documented elsewhere. The list of these publications is provided below.

The body of work dealing with the grammar of film (Chapter 3 and Chapter 4) has resulted in the following publications:

- Phung, D. Q., Dorai, C., and Venkatesh, S. (2002a). Narrative structure analysis with education and training videos for E-learning. In *International Conference on Pattern Recognition (ICPR'02)*, pages 835–839, Quebec, Canada.
- Phung, D. Q., Venkatesh, S., and Dorai, C. (2002b). High level segmentation of instructional videos based on the content density function. In *ACM International Conference on Multimedia (ACM'02)*, pages 295–298, Juan Les Pins, France.
- Phung, D. Q., Venkatesh, S., and Dorai, C. (2003a). On extraction of thematic and dramatic functions in educational films. In *IEEE International Conference on Multimedia and Expo (ICME'03)*, pages 449–452, Baltimore, New York, USA.
- Phung, D. Q., Venkatesh, S., and Dorai, C. (2003b). Hierarchical topic segmentation in instructional films based on cinematic expressive functions. In *ACM International Conference on Multimedia (ACM'03)*, pages 287–290, Berkeley, USA.

The work on the Hierarchical Hidden Markov Models (Chapter 5 and Chapter 6) has resulted in the following publications:

- Phung, D. Q., Venkatesh, S., and Bui, H. H. (2004a). Automatically learning structural units in educational videos using the Hierarchical HMMs. In *International Conference on Image Processing (ICIP'04)*, Singapore.
- Phung, D. Q., Bui, H. H., and Venkatesh, S. (2004b). Content structure discovery in educational videos with shared structures in the Hierarchical HMMs. In *Joint Int. Workshop on Syntactic and Structural Pattern Recognition*, pages 1155–1163, Lisbon, Portugal. (Also available in *Lecture Notes in Computer Science: Advanced*

in Statistical, Structural and Syntactical Pattern Recognition, Vol. 3138, p. 1155 Springer-Verlag).

- Bui, H. H., Phung, D. Q., and Venkatesh, S. (2004c). Hierarchical hidden Markov models with general state hierarchy. In D. L. McGuinness and G. Ferguson, editors, *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI'04)*, pages 324–329, San Jose, California, USA. AAAI Press / The MIT Press.

Although not directly related to the domain of educational videos investigated in this thesis, the application of the theory on the HHMM in Chapter 5 has resulted in the following collaborative work:

- Nguyen, T. N., Phung, D. Q., Venkatesh, S., and Bui, H. (2005). Learning and Detecting Activities from Movement Trajectories Using the Hierarchical Hidden Markov Models. *IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR'05)*, June 2005, San Diego, USA.
- Duong, V. T., Bui, H., Phung, D. Q., and Venkatesh, S. (2005). Activity Recognition and Abnormality Detection with the Switching Hidden Semi-Markov Model. *IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR'05)*, June 2005, San Diego, USA.

Abbreviations

Abbreviations	Meanings
iid.	independently and identically distributed
AIO	Asymmetric Inside-Outside
BN	Bayesian Network
CBIR	Content-based Image Retrieval
CMA	Computational Media Aesthetics
DBN	Dynamic Bayesian Network
DCT	Discrete Cosine Transform
EM	Expectation Maximisation
ESS	Expected Sufficient Statistics
FP	False Positive
FSM	Finite State Machine
FST	Fine-Singer-Tishy (referring to the algorithm in Fine <i>et al.</i> (1998))
GMM	Gaussian Mixture Model
LHS	Left-Hand-Side
HMM	Hidden Markov Model
HHMM	Hierarchical Hidden Markov Model
MBR	Minimum Bounding Rectangle
MCM	Multimedia Content Management
MCMC	Markov Chain Monte Carlo
ML	Maximum Likelihood
MP	Murphy-Paskin (referring to the algorithm in Murphy and Paskin (2001))
MPEG	Moving Picture Experts Group
NN	Neural Network
PCFG	Probabilistic Context Free Grammar
QVC	Query-by-Video-Clip
SCFG	Stochastic Context Free Grammar
SS	Sufficient Statistics
SVM	Support Vector Machine
RHS	Right-Hand-Side
TP	True Positive
VCA	Video Content Analysis

Notations

General Notations

Notation	Meanings
$\{\emptyset\}$	An empty set.
\mathbb{R}	The set of real numbers.
$ \mathcal{X} $	The cardinality of set \mathcal{X} .
\boxplus	End of an example.
\blacksquare	End of a proof.
$\perp\!\!\!\perp$	The conditional independence operator.
$\delta_x^{(i)}$	The Dirac delta function, which returns 1 if $x = i$ and 0 otherwise.
\triangleq	The definition operator, ie: it means ‘defined as’.
$T\langle\tau\rangle$	Sufficient statistics for the parameter τ .
$\langle\tau\rangle_Q$	Expected sufficient statistics of τ with respect to the distribution Q .

Notations Related to Video Analysis

V	Commonly used to denote a video.
S	Commonly used to denote a shot.
CS	Commonly used to denote a candidate shot.
ϕ	Commonly used to denote a video frame.
$\mathcal{F}(\phi)$	The face detector which returns 0 if face is found in ϕ , and 1 otherwise.
$\mathcal{T}(\phi)$	The text detector which returns 0 if no caption is found in ϕ , and 1 otherwise.
ON	On-screen Narration.
DN	Direct Narration.
AN	Assisted Narration.
VO	Voice-over.
LK	Linkage.
SN	Supportive Narration.
ANwT	Assisted Narration with Text.
ANwS	Assisted Narration with Scene.
VOwT	Voice-over with Text.
VOwS	Voice-over with Scene.
VOTS	Voice-over with both Text and Scene.
FvLK	Functional Linkage.
ExLK	Expressive Linkage.

Specific Notations

In Chapter 5, a heavy load of notations is used. We strongly suggest the readers to frequently refer to this page when reading the chapter for convenient and quick references.

Notations for Parameters of the HHMM

Symbol	Meanings
θ	Usually used to denote the set of parameters for a HHMM.
D	The depth of the HHMM.
ζ	The topology of a HHMM.
\mathcal{Y}	The observation space, ie: the set of all possible observed alphabets in the discrete case.
\mathcal{S}^d	The set of all states at level d .
$ \mathcal{S}^d $	The number of states at level d .
i, j	Commonly used to denote the children states.
p, q	Commonly used to denote the parental states.
$\text{pa}(i)$	The set of parental states of i specified by ζ .
$\text{ch}(p)$	The set of children states of p specified by ζ .
π	The initial probability matrix, where $\pi_i^{d,p}$ is the probability of i being called given the parent p at level d .
A	The transition probability matrix, where $A_{i,j}^{d,p}$ is the transition probability $i \rightarrow j$ given their parent p at level d .
B	The observation probability matrix, where $B_{v i}$ is the probability of observing v given the state i .

Notations Related to the DBN structure

Symbol	Meanings
\mathcal{H}	The set of hidden variables in the DBN.
\mathcal{O}	The set of observable variables in the DBN.
\mathcal{V}	The set of all variables in the DBN, ie: $\mathcal{V} = \mathcal{O} \cup \mathcal{H}$.
\mathcal{D}	The data set, which usually consists of K iid. observation sequences $\{\mathcal{O}^{(1)}, \dots, \mathcal{O}^{(K)}\}$.
t, l, r	Used to denote temporal indices. In a general, l is used for <i>left</i> index, r for <i>right</i> index, and t for general purpose.
d, d^*	Used to denote hierarchic indices.
x_t^d	The state variable at level at time t and level d .
e_t^d	The end-state variable at time t and level d . It takes on either 1 or 0 to indicate the ending status of x_t^d at time t .
y_t	The observation variable at time t .
$x_{l:r}^d$	Used to denote the set $\{x_l^d, x_{l+1}^d, \dots, x_r^d\}$.
$x_t^{d:d^*}$	Used to denote the set $\{x_t^d, x_t^{d+1}, \dots, x_t^{d^*}\}$.
$x_{l:r}^{d:d^*}$	Used to denote the set $\{x_l^{d:d^*}, \dots, x_r^{d:d^*}\}$ or equivalently $\{x_{l:r}^d, \dots, x_{l:r}^{d^*}\}$.
$\cdot x_t^d$	Equivalent to $\{x_t^d, e_{t-1}^d = 1\}$.
x_t^d	Equivalent to $\{x_t^d, e_t^d = 1\}$.
$\cdot \tau_t^d$	A random variable used to denote the time when state x_t^d has started.
τ_t^d	A random variable used to denote the finishing time of state x_t^d .
\mathcal{H}_t	The set of all hidden variables at time t .
\mathcal{O}_t	The set of all observed variables at time t .
$\mathcal{O}_{l:r}^{\text{in}}$	The set of all observed variables ‘inside’ the range $[l, r]$.
$\mathcal{O}_{l:r}^{\text{out}}$	The set of all observed variables ‘outside’ the range $[l, r]$, ie: $[1, t-1] \cup [r+1, T]$.
$\text{SB}_{l:r}^{d,p}$	The symmetric boundary event $\triangleq \{x_t^d = p, \cdot \tau_t^d = l, \tau_t^d = r\}$.
$\text{SI}_{l:r}^{d,p}$	The set of all variables ‘inside’ the boundary $\text{SB}_{l:r}^{d,p}$.
$\text{SO}_{l:r}^{d,p}$	The set of all variables ‘outside’ the boundary $\text{SB}_{l:r}^{d,p}$.
$\text{AB}_{l:r}^{d,p}(i)$	The asymmetric boundary event $\triangleq \{\cdot x_t^d = p, \cdot \tau_t^d \geq l, x_r^{d+1} = i\}$.
$\text{AI}_{l:r}^{d,p}(i)$	The set of all variables ‘inside’ the boundary $\text{AB}_{l:r}^{d,p}(i)$.
$\text{AO}_{l:r}^{d,p}(i)$	The set of all variables ‘outside’ the boundary $\text{AB}_{l:r}^{d,p}(i)$.




Notations for the Auxiliary Variables

Symbol	GUI	Meanings	Page
	\circ	Used to present a hidden variable in the DBN.	
	\bullet	Used to present a started state variable, ie: $\{x_t^d\}$.	
	\ominus	Used to present a terminated state variable, ie: $\{x_t^d\}$.	
	\oplus	Used to present a continuing state variable, ie: $\{x_t^d, e_{t-1}^d = 0, e_t^d = 0\}$.	
	\bullet	This drawing is used to present an observed variable.	
φ_t	NA	The scaling factor at time t .	135
$\xi_t^{d,p}(i, j)$	$\langle \overset{p}{\circ} \overset{i}{\bullet} \overset{j}{\bullet} \rangle$	The horizontal transition probability variable.	116
$\tilde{\xi}_t^{d,p}(i, j)$	NA	The scaled version of $\xi_t^{d,p}(i, j)$.	135
$\xi_t^{d,p}(i, \text{end})$	$\langle \overset{p}{\circ} \overset{i}{\bullet} \rangle$	The horizontal transition probability of ‘going to end-state’ variable.	116
$\chi_t^{d,p}(i)$	$\langle \overset{p}{\bullet} \overset{i}{\bullet} \rangle$	The vertical transition probability variable.	116
$\tilde{\chi}_t^{d,p}(i)$	NA	The scaled version of $\chi_t^{d,p}(i)$.	135
$\Gamma_t^D(i)$	$\langle \overset{i}{\bullet} \overset{p}{\bullet} \rangle$	The emission probability variable.	116
$\tilde{\Gamma}_t^D(i)$	NA	The scaled version of $\Gamma_t^D(i)$.	135

Notations for Auxiliary Inside/Outside Variables

$\alpha_{l;r}^{d,p}(i)$	$\left[\overset{p}{\bullet} \overset{i}{\bullet} \right]$	The asymmetric inside variable.	120
$\ddot{\alpha}_{l;r}^{d,p}(i)$	NA	The partially scaled version of $\alpha_{l;r}^{d,p}(i)$.	138
$\tilde{\alpha}_{l;r}^{d,p}(i)$	NA	The scaled version of $\alpha_{l;r}^{d,p}(i)$.	136
$\alpha_{l;r}^{d,p}(i)$	$\left[\overset{p}{\bullet} \overset{i}{\oplus} \right]$	The continuing-asymmetric inside variable.	130
$\ddot{\alpha}_{l;r}^{d,p}(i)$	NA	The partially scaled version of $\alpha_{l;r}^{d,p}(i)$	138
$\tilde{\alpha}_{l;r}^{d,p}(i)$	NA	The scaled version of $\alpha_{l;r}^{d,p}(i)$	136
$\underline{\alpha}_{l;r}^{d,p}(i)$	$\left[\overset{p}{\bullet} \right] \overset{i}{\bullet}$	The started-asymmetric inside variable.	120
$\tilde{\underline{\alpha}}_{l;r}^{d,p}(i)$	NA	The scaled version of $\underline{\alpha}_{l;r}^{d,p}(i)$.	136
$\Delta_{l;r}^{d,p}$	$\left[\overset{i}{\bullet} \cdots \overset{i}{\bullet} \right]$	The symmetric inside variable.	120
$\tilde{\Delta}_{l;r}^{d,p}$	NA	The scaled version of $\Delta_{l;r}^{d,p}$.	136
$\Delta_{l;r}^{d,p}$	$\left[\overset{i}{\bullet} \cdots \overset{i}{\oplus} \right]$	The continuing-symmetric inside variable.	131
$\lambda_{l;r}^{d,p}(i)$	$\left] \overset{p}{\bullet} \cdots \overset{i}{\bullet} \right[$	The asymmetric outside variable.	120
$\tilde{\lambda}_{l;r}^{d,p}(i)$	NA	The scaled version of $\lambda_{l;r}^{d,p}(i)$.	136
$\Lambda_{l;r}^{d,p}$	$\left] \overset{p}{\bullet} \cdots \overset{i}{\bullet} \right[$	The symmetric outside variable.	121
$\tilde{\Lambda}_{l;r}^{d,p}$	NA	The scaled version of $\Lambda_{l;r}^{d,p}$.	136

Notations Related to the Viterbi Algorithm

Symbol	GUI	Meanings	Page
$\delta_{l:r}^{d,p}(i)$		The maximum (asymmetric) inside variable.	143
$\tilde{\delta}_{l:r}^{d,p}(i)$		The started maximum (asymmetric) inside variable.	143
$\lambda_{l:r}^{d,p}$		The maximum symmetric inside variable.	143
$\psi_{l:r}^{d,p}(i)$	NA	The (best) switching time variable.	143
$\omega_{l:r}^{d,p}(i)$	NA	The (best) switching state variable.	144

Chapter 1

Introduction

The fast growing advances in electronics, computer, and communication technologies have greatly transformed our lives in many ways. Possibly, one of the most significant outcomes is the emergence of the field of ‘multimedia’ – which, perhaps an unknown term 50 years ago, has now become ubiquitous. Multimedia data such as digital music and videos (eg: news, movies, recorded footage) have been undoubtedly integrated into our daily lives. The problematic end, however, is the need to cope with this exponential growth of data made available everyday. This proliferation has also spawned the need for methods and technologies to bring about effective content management, including automatic indexing, browsing, searching and distribution of multimedia data. Traditional methods of manually examining data are time-consuming, expensive, unscalable, and conceivably, close to impossible.

One of the central research topics in multimedia content management is the *indexing* problem, which is analogous to the problem of providing a table of contents and keyword indices for a book. Hundreds of terabytes of video data would appear almost ineffectual, unless it is somehow organised, summarised and indexed. The ability to index data provides the *units* in which the data can be effectively summarised, represented, and subsequently made available for searching and retrieving. At the simplest level, mechanisms such as keywords for texts can be used. However, multimedia data contains far richer information, naturally embedded in several dimensions (temporal, spatial, conceptual, etc.), that simple keywords cannot describe. The pressing need is to move beyond simple keyword-based indexing to define richer *units* of indexation so that multimedia data can be better managed and accessed (eg: Smeulders *et al.* (2000); Jain (2001); Dorai and Venkatesh (2002); Wu and Kankanhalli (2000)). The driving aim is to shorten the *semantic gap*, which refers to the discrepancy in the semantics that users find comfortable with in the browsing and searching systems, and the simplicity of features that can be computed with existing tools.

Arguably, to understand and to build a semantic grid for a data domain, one needs to ex-

amine the *creating force* behind it, that is “to create tools for automatically understanding video, we need to be able to interpret the data with its maker’s eyes” (Dorai and Venkatesh, 2001b). It is in this fundamental view that we place the work in this dissertation. Two key questions are identified. The first is the quest for a *systematic way* to understand the ‘maker’s eyes’, and the second is the need to develop *advanced mechanisms for inference and interpretation* of the data. In answering the first question, one essentially needs to ‘return to roots’ and acquire specific domain knowledge in a *methodical manner* to help the investigation. Here we base our work on the *Computational Media Aesthetics* framework – a systematic method to bridge the semantic gap through Film Grammar recently pioneered in (Dorai and Venkatesh, 2001a,b, 2002). From the domain perspective, while many research attempts have sought solutions for a wide range of specific domains such as TV programs (news, sitcoms, commercials, reports) and entertainment films, none or very little attention has been paid to education-oriented videos, a rich class of film genre that has an extremely important role in building e-services for learning and training. Educational films, with their unique characteristics, therefore warrant further study and form the domain of investigation in this thesis. For readability, we shall often refer to this video genre as ‘educational films’. The terms ‘video’ and ‘film’ are also used interchangeably in this thesis.

While domain knowledge can help in identifying meaningful semantics and constructs, or distilling aspects of visual and aural appeal, *advanced methods to infer and interpret the data content is of no less importance*. Towards this end, we base our work in a probabilistic framework, and in particular the use of the Hierarchical Hidden Markov Models.

1.1 Aims and Approach

This thesis presents an investigation into the problem of content analysis for the domain of educational videos. Our objectives are:

- To construct meaningful semantic descriptions of the content through Film Grammar and apply them to the problem video annotation and segmentation.
- To advance knowledge in probabilistic methods for video annotation and segmentation.

Our approach, from the domain-specific perspective, is motivated by Film Grammar, a branch of film analysis that elucidates the conventions that are used by directors worldwide to convey ideas and meanings (Arijon, 1976; Monaco, 1977; Sobchack and Sobchack, 1987).

Here, we base our work on the Computational Media Aesthetics (CMA) framework. We concentrate on a more specific body of grammar for educational videos to shape our understanding of this domain. We seek to expose the semantic information embedded in the video production by focusing not merely on the representation of the perceived event, but also on the emotion and visual appeal of the content. The key difference in this framework is that to both define and extract meaningful semantic structures we seek guidance from Film Grammar. In particular, it tells us how production elements can be manipulated in order to achieve intended messages. Thus, complex constructs are both defined and extracted *only* if Film Grammar tells us that it is an element that the director crafts or manipulates intentionally. We use the grammar knowledge at two stages: (1) to provide insights into domain-specific, meaningful structural units and expressive functions that are computable from primitive features, and (2) to provide additional prior structural information to be incorporated in a probabilistic framework for semantic discovery and video segmentation.

To construct a probabilistic framework, we propose the use of the Hierarchical Hidden Markov Model (HHMM) with shared structures - a form of parameter tying at multiple levels. This is motivated by the fact that *semantic concepts in videos are not only naturally organised in a hierarchical manner, but also contain shared structures in the hierarchy*. This problem has not been addressed in the literature rigorously. In this thesis, we approach this issue by theoretically extending the original Hierarchical Hidden Markov Model proposed in (Fine *et al.*, 1998) to handle shared structures and apply it to the problem of semantic discovery and segmentation in educational videos. In particular, our specific aims are:

- A thorough examination of Film Grammar pertaining to the educational video genre, including the techniques used in the manipulation of visual and aural elements to convey messages.
- The identification of a hierarchy of meaningful narrative structural units in educational videos. The extraction of useful visual and aural features, and a classification system to map a shot into the narrative structure hierarchy.
- The identification and extraction of useful expressive functions that provide vital information about the content and structure of the video. In particular, for educational videos, we aim to extract the *content density* (a measure of the ‘rate of information delivered’), *thematic* and *dramatic* functions through the examination of aural and visual elements, their impacts, co-occurrences or co-existence that contribute to expressive functions.
- Development of the theoretical background to handle shared structures in the Hierarchical Hidden Markov Model. This includes the representation, inference and

learning for the model.

- The application of the Hierarchical Hidden Markov Model to the problem of segmentation and semantic discovery in educational videos.

1.2 Significance and Contribution

The significance of this work lies in two areas: (1) the development of Film Grammar based and probabilistic approaches for content management of education videos, and (2) the theoretical extension of the Hierarchical Hidden Markov Model. In particular, our contributions are:

- A hierarchy of narrative structural units in educational videos and the mechanism to learn and recognise these structures. We present commonly observed rules and conventions in educational media productions to manipulate the presentation of the content to match the learner's needs. Leveraging the production grammar to shape our understanding of the common structural elements employed in educational media, a hierarchy of narrative structures is proposed. Characteristics of each structural element manifesting as a sequence of shots are examined and a set of audiovisual features for capturing the differences between them is subsequently proposed. The implication of this piece of work is a system that can annotate educational videos to enable search and retrieval of informative content in a hierarchic manner.
- A continuous measure of the *content density* function that reflects the 'rate of information delivered' for educational videos, and its application to segment a video into subtopics. This function is a useful measure of the educational content in its own right, as demonstrated empirically to infer about the nature of subtopics. Close analysis of educational films shows that the flow in the amount of the material delivered over time reveals illuminating information about the structure of the video: a drop from high density to low density for a few shots is deliberate and used to draw attention. Based on this observation, we present two algorithms for the detection of subtopic transitions.
- A further examination of two useful expressive elements for the educational genre, namely the *thematic* and *dramatic* functions. These two functions are useful as they provide vital information about the 'mediation' or the involvement of the video maker. The thematic function reflects 'instructional' and 'informative' portions in the video, that is, the sections where the video maker decides to 'step in' and interject in the subject being shown. The dramatic function, on the other hand, reflects the

‘dramatic’ aspect of the video. Key contributing elements to these functions are studied empirically and computational forms for them are subsequently constructed. This study is significant as it allows the inferences relating to thematic and dramatic content to be made, which enables higher level annotation and segmentation of the content.

- An algorithm for the segmentation of topical content in educational videos based on the content density and thematic functions. In addition to information about subtopic boundaries from the dynamic changes in the content density function, we observe that a drop or rise in the mediation process is also deliberate to emphasise *main* topics. We hypothesise key factors that influence the correlation between subtopic and topic boundaries with these functions and develop an edge-based detection algorithm to segment a video into main topics and subtopics.

In seeking a probabilistic framework for annotation and segmentation of educational videos, our contributions are:

- A new theoretical extension to the original Hierarchical Hidden Markov Model proposed in (Fine *et al.*, 1998) to handle shared structures. The resulting model allows a form of the Hierarchical Hidden Markov Model with its most generalised form of state hierarchy. The direct modeling of shared structures results in practical savings in computation, and more accurate parameter learning with less training data. In addition, unproven technical details in the original work (Fine *et al.*, 1998) are also formally verified in this thesis.
- A novel Asymmetric Inside-Outside algorithm to perform inference in the presence of shared structures which has the same complexity as in the original model (Fine *et al.*, 1998), and the contribution of an Expectation Maximisation parameter estimation procedure for the problem of learning in the HHMM.
- A novel scaling algorithm for the HHMM to avoid numerical underflows. This is a very important issue that needs to be solved in order for the HHMM to be applied in real-world applications, a problem not addressed in the original paper (Fine *et al.*, 1998).
- A set of diagrammatic tools is developed to intuitively visualise the structure of complex computations during the inferencing process. Although formal derivation of several recursive algorithms appears very complicated, the tools simplify the derivation process enormously.
- A new generalised Viterbi algorithm and continuous observation modeling for our extended Hierarchical Hidden Markov Model and its application for the problem of

subtopic detection in educational videos. This approach is significant as it demonstrates a semi-supervised methodology towards the segmentation problem, in which the domain knowledge is incorporated as prior information about the topology of the Hidden Markov Model.

- Two new applications of the Hierarchical Hidden Markov Model for segmentation and semantic analysis in educational videos, in which we demonstrate how the HHMMs can be used to segment the video, and how to automatically learn useful structural units from the data.

In addition, the theoretical extension allows the HHMM to be readily applied to a wide range of potential applications apart from video content analysis, especially for the problem of video surveillance and complex behaviour recognition.

1.3 Structure of the Thesis

This thesis is structured as follows. In *Chapter 2*, we provide a literature review and related background to our work. Starting with an overview on video content analysis, we identify open issues and key developments in this area. We then review existing work on Film Grammar based and probabilistic methods for video analysis.

In *Chapter 3*, we present our first contribution to the understanding of the educational domain and the construction of a hierarchy of structural narrative elements. We study the audio and visual characteristics of these narrative classes and subsequently design a set of features used in the C4.5 algorithm to learn and recognise this hierarchy. Experimental results are also presented.

Chapter 4 presents our work on the extraction of the three high level constructs, namely the content density, thematic and dramatic functions. We first construct the content density function and then study its behaviour in relation to the nature of subtopic boundaries in educational videos. We then propose heuristic and probabilistic algorithms to detect subtopic transitions. The next part of this chapter deals with the extraction of dramatic and thematic functions. We then study aesthetic elements that influence these functions and empirically experiment to select key contributing factors. Computational forms for these functions are subsequently proposed and experiments are performed to evaluate their validity. Finally, we present a scheme to detect a two-level hierarchy of topical content using the content density and the thematic functions.

Chapter 5 presents our next major contribution in this thesis by rigorously revising the original Hierarchical Hidden Markov Model and extending it to handle shared structures. We first present the model definition, its Dynamic Bayesian Network (DBN) representation and formally derive a set of conditional independences in this DBN. We then address the problem of inference and learning in this model. For inference and learning in the HHMM with shared structures, we first explain the algorithm derived in (Fine *et al.*, 1998), highlight current problems when shared structures exist, and then present a novel Asymmetric Inside-Outside algorithm to overcome these issues. Complexity analysis and numerical results are also provided. In the next part of this chapter, a novel scaling algorithm to avoid numerical underflow when dealing with long observation sequences is presented. Three appendices at the end of the thesis are also included to provide several proofs and unexplained technical details developed in the chapter

In *Chapter 6*, we present a generalised Viterbi algorithm for the extended HHMM, and also address the issue of modeling continuous observations as a mixture of Gaussians. Next, we present two novel applications of the HHMM, one for the segmentation of subtopics and the other for automatically learning meaningful units in educational videos at multiple levels.

Finally, *Chapter 7* provides a summary of the work in this thesis and discusses some ideas and directions for possible future work.

Chapter 2

Related Background

In this chapter, we provide a review of relevant literature and related background for the work investigated in this thesis. As previously stated, this thesis aims at tackling the problem of content analysis for the domain of educational videos, in which we seek solutions from the systematic application of Film Grammar, and from formal probabilistic models. This chapter is organised into three main sections. In Section 2.1, we provide an overview of the video content analysis area. Section 2.2 provides a review on Film Grammar based methods, and Section 2.3 contains a review of probabilistic methods. Finally, the chapter ends with some concluding remarks in Section 2.4.

2.1 Video Content Analysis: An Overview

Recent advances in computing technologies have enabled video data to be captured, stored and transmitted efficiently. At the same time, it poses a fundamental problem of how this vast amount of data can be effectively indexed and organised in a meaningful manner to facilitate better management. This underlying challenge is broadly known in the research community as the problem of Multimedia Content Management (MCM), in which the field of *video content analysis* (VCA) is one of the main challenges (Lee *et al.*, 1992). The goal of VCA is to provide tools and techniques for automatically extracting, representing and understanding video structures to assist content indexing, browsing, searching, retrieval and distribution.

The field of video content analysis is broad and existing approaches can be categorised in different ways. From recent surveys (Ngo *et al.*, 2001; Dimitrova *et al.*, 2002; Snoek and Worring, 2000), we identify three main, yet inter-related, problems: (1) video parsing and segmentation, (2) video representation and summarisation and (3) searching and retrieval of video databases. Broadly speaking, video parsing and segmentation provides the basic

building blocks upon which higher level constructs are built. Summarisation of video data is concerned with providing a condensed (non-linear) summary of the content so that a lengthy video can be quickly reviewed without overlooking important details. Finally, the problem of search and retrieval lies in developing effective mechanisms to allow the user to search and query video clips of interest, similar to searching texts with Google.

We will, in turn, briefly review these three areas, and in doing so and where appropriate, we discuss what are the open problems and current directions. We then review Film Grammar based methods in Section 2.2 and probabilistic approaches in Section 2.3.

2.1.1 Parsing and Segmentation of Videos

Analysing a video, in the context of MCM, is to *understand* its content and subsequently give it structure, whereby *abstract* indices are built to be used in the video system. The first step in this process is the parsing and segmentation of the videos, to which much of the existing research efforts have been devoted. Video parsing and segmentation can be viewed as a hierarchic process. At the bottom level, the goal is to move beyond meaningless raw pixels or audio signals, providing a new dimension in which the video can be alternatively characterised. This transformation is commonly referred to as the *primitive features* extraction stage. Visual and aural features such as colour, texture, facial descriptors, loudness, frequency subband descriptors are computed. At higher levels, based on these features, the video is *partitioned* into meaningful and coherent units such as shots, or scenes.

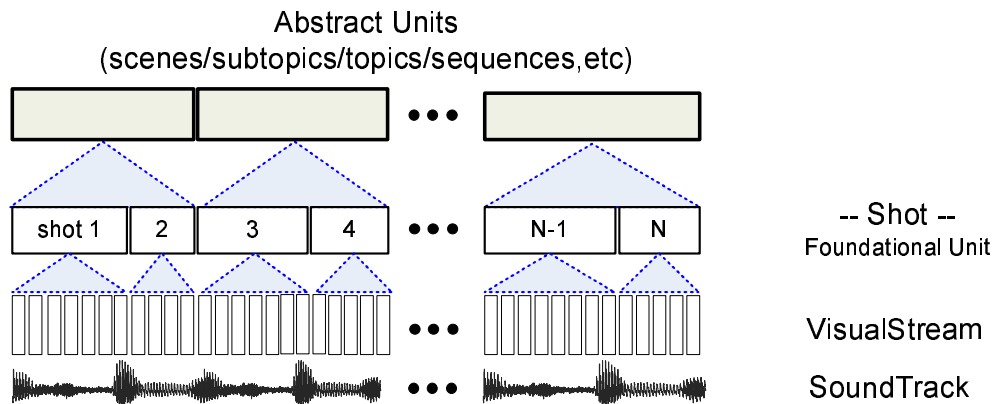


Figure 2-1: A typical structural decomposition of a video.

Figure-(2-1) depicts a general structural decomposition of a video when it is parsed and segmented.

2.1.1.1 Primitive feature extraction

Mining meaningful features to characterise the video is one of the most basic problems in video analysis. Raw pixels or aural signals provide no useful information. Both visual and aural features need to be meaningfully formed and extracted. In our discussion, primitive features refer to static features, and are extracted from a *still* picture or a *fixed* segment of audio signal.

Low-level image features

Techniques and algorithms to extract low-level features from the still image are mostly inherited from the computer vision community. These methods are utilised mainly in the area of content-based image retrieval (CBIR) to provide alternative *visual-based* retrieval systems as opposed to the *text-based* systems in the early 70's (Rui *et al.*, 1999; Li *et al.*, 2001). Recent surveys (Li *et al.*, 2001; Antani *et al.*, 2002) show that the three most popular feature spaces used in CBIR systems are based on: *colour*, *shape*, and *texture*. Colour-based features (eg: colour histograms, colour moments) are most widely used for their two main advantages: computational efficiency, and independence from view (scaling, translation, rotation) and resolution (Li *et al.*, 2001). Texture-based features are mainly concerned with providing the visual regularities in the whole image or homogeneous regions (Antani *et al.*, 2002). Shape-based features provide an alternative method for retrieval by matching shapes; one popular approach, for example, is the use of extracted contours from a set of deformable images to match users' sketches (eg: Pala and Santini (1999)). Good surveys abound to provide details about what low-level features are computable and their scope of applications in CBIR. Comprehensively reviewing them, however, is of little relevance to our work. We refer readers to (Rui *et al.*, 1999; Smeulders *et al.*, 2000; Li *et al.*, 2001) and references therein for further information.

Mid-level images features – facial and textual information

Besides low-level features which are usually computed directly from raw signals, mid-level features provide useful information. With respect to educational videos, facial and textual information provide vital clues about the content and structure of the video.

Face Detection

Detecting human faces provides useful information about the content of videos, especially for the class of the educational/informative genre such as news, documentaries, training/teaching videos or lecture videos. In this type of video, the presence of a human face provides vital clues about the structure of the video; for example, in a teaching video, a frontal view of the teacher usually implies a topic is either being introduced or being summarised; or in a news report, the arrival of the anchor often demarcates important segments. Existing approaches to face detection can be broadly classified into two categories:

(1) feature-based approaches, and (2) classification-based approaches. In the former group, facial features such as eyes, nose and mouth are first detected and then geometric relationships are established to search for a face. For example, Kotropoulos and Pitas (1997) search for hair, eyebrows, eyes, mouth and chin by simple horizontal and vertical profiles of the image; (Xu and Sugimoto, 1998; Yang *et al.*, 1998) search for eyes, pupils and nostrils from the dark regions; Hsu *et al.* (2001) detect eyes and mouth by observing differences in intensity signatures for the two regions. Other feature-based approaches also include: colour-based face detection (Yang and Waibel, 1996; Xu and Sugimoto, 1998; Bojic and Pang, 2000; Chai and Ngan, 1999; Chai and Bouzerdoum, 2000), texture-based face detection (Bonet and Viola, 1998; Garcia and Tziritas, 1999), and shape-based face detection (Sirohey, 1993; Govindaraju, 1996; Menser and Wien, 2000). Face detection in the classification-based approach typically involves the training of a machine learning algorithm to distinguish between face and non-face regions. A search window is constructed, possibly at multiple scales, and then the whole image is scanned, searching for face patterns. These methods include the use of the Karhunen-Loeve transform (Kriby and Sirovich, 1990), which is later extended to eigenfaces (Turk and Pentland, 1991); the use of K -means to cluster face and non-face regions (Sung and Poggio, 1998); the use of naive Baye’s classifier for face and non-face regions (Moghaddam and Pentland, 1997; Pham *et al.*, 2001); the use of Support Vector Machines to detect vertically oriented and non-occluded frontal faces (Osuna *et al.*, 1997); and finally the use of neural networks (NN) (Rowley *et al.*, 1998; Feraud *et al.*, 2001).

We refer readers to the surveys of (Hjelmas and Low, 2001; Yang *et al.*, 2002) and references therein for further details. In our work, we use the neural network based face detection developed in (Rowley *et al.*, 1998) to detect frontal faces, which we shall describe now¹. The basic steps for this algorithm are shown in Figure-(2-2). Neural networks are trained

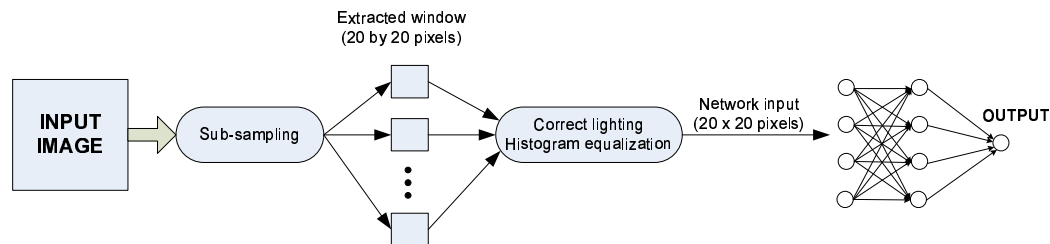


Figure 2-2: Steps in the NN-based face detection algorithm proposed in (Rowley *et al.*, 1998).

on a window size of 20×20 and an output of between -1 and 1 is reported to indicate the presence of a face in each window. To handle faces larger than the pre-defined window size, the original image is repeatedly subsampled by a factor of 1.2 and the sampled images

¹We gratefully acknowledge Henry A. Rowley for generously sharing his face detection library used in this work.

are then input to the neural networks. To further enhance the detection process, the authors propose two post-processing techniques, namely merging overlapping detections from a single network and arbitration of detection results from multiple networks. Using the CMU face database, the proposed algorithm is reported to achieve a detection rate between 77.9% and 90.3%.

Text Caption Detection

Another important and integral part of video that potentially contains rich information about the content structure is *text*, which is either embedded or superimposed within visual frames. Extraction of textual information involves localisation, detection and recognition of text from a given image. The problem is usually complicated by the variances in size, font, style, orientation, and so on. Embedded texts, also referred to as *scene* texts, are therefore generally harder to detect compared with superimposed texts. We refer readers to (Jung *et al.*, 2004) for a more comprehensive survey on text extraction in images and video. In this work, we employ the algorithm proposed in (Shim *et al.*, 1998)² for the task of detecting superimposed texts. This text detection system, as shown in Figure-(2-3), consists of three main steps: (1) isolating candidate regions that may contain texts, (2) separating a character region from its surroundings, and finally (3) verifying the presence of texts by a consistency analysis procedure.

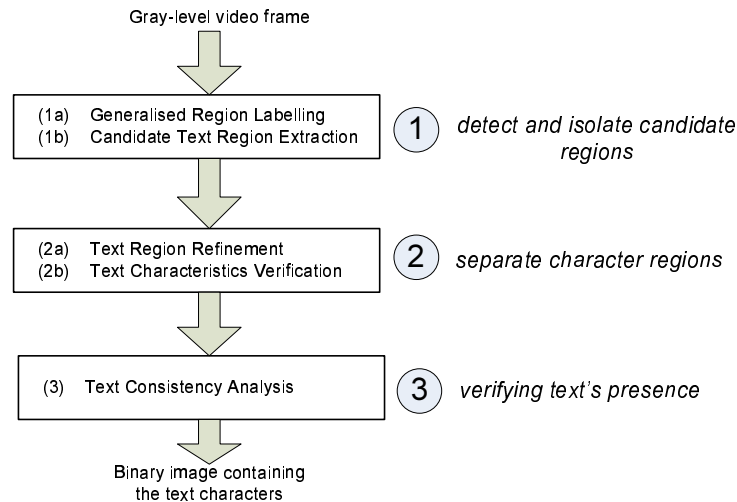


Figure 2-3: Steps in the text detection algorithm proposed in (Shim *et al.*, 1998)

To detect text candidate regions, a generalised region labeling algorithm using chain-code is proposed to partition the input image into non-overlapping homogeneous regions followed by a candidate-text-region extraction. Selecting a candidate text region is based on their geometric characteristics; for example a region is removed if the width and height

²We gratefully acknowledge Shim *et al.* (1998) for providing us with the source code of their text detection algorithm.

of its MBR (minimum bounding rectangle) are greater than 24 and 32 pixels respectively for a 320×240 image. Next, each character region is isolated from its surroundings by a text-region-refinement procedure which uses a local threshold selection scheme to extract character segments within each candidate regions. This is followed by a text-characteristics-verification procedure to further enhance the character region by applying a number of heuristics (eg: remove a character region if its area is less than 12). Finally, a text-consistency-analysis procedure is performed to verify the consistency between neighbouring text regions to eliminate false positive regions. Three consistency criteria are used, including: (a) position analysis (checking inter-character spacing), (b) horizontal alignment analysis of characters, and (c) vertical proportions analysis of adjacent character regions. The final output from the entire process is a binary image containing the text characters.

Audio Processing

Besides the visual stream, most of the existing video data nowadays is accompanied with a soundtrack, which provides another rich source of information³. Recent research in content-based video indexing systems have started to incorporate the audio information at many phases ranging from annotation, segmentation, to searching and retrieval. In these applications, the audio track is commonly handled by dividing it into a sequence of overlapping clips. Each clip is in turn subdivided into a sequence of overlapping frames as shown in Figure-(2-4). Feature extraction is first carried at the frame level and then

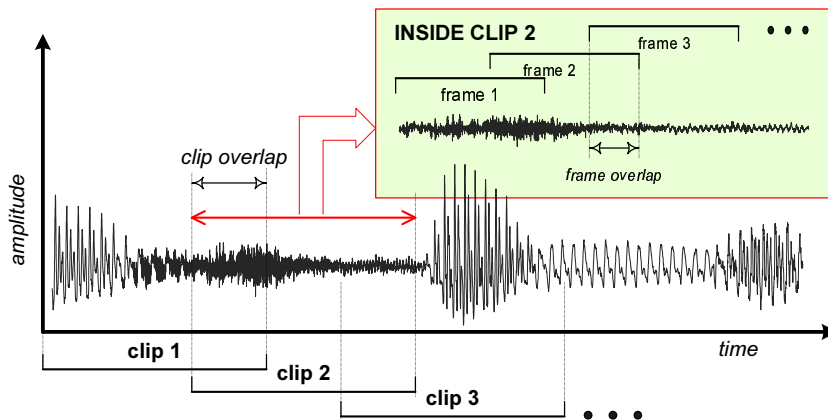


Figure 2-4: A typical decomposition of an audio track in a movie into a sequence of overlapping clips and frames (adapted from Phung (2001)).

combined (eg: by taking the average) to form clip-level features. In this thesis, unless otherwise explicitly stated, the audio is digitised at 44.1 kHz, the clip size is 1 second with an overlap of 0.5 second; the frame size is 512 sampling points (≈ 12 ms) with an overlap of 256 samples. The feature extraction system described in (Phung, 2001) is used

³Audio information indexing and retrieval is itself a mature research topic for decades concerning the management of audio data available from sources such as music collections or telephone messages.

in this thesis. Essentially, this feature set is extracted based on: (1) raw signals (*time*), (2) Fourier analysis, and (3) wavelet analysis. Table-(2.1) summarises this feature set. For more complete details, we refer readers to (Phung, 2001).

DOMAIN	FEATURE DESCRIPTION
Time	Short-time Energy & Average Magnitude Loudness (mean and std.), Loudness Dynamic Change Zero Crossing Rate, Silence Interval Distribution
Fourier	Frequency Bandwidth and Brightness Subband Energy Ratio Mel-Cepstrum
Wavelet	Subband Energy Subband Variance Frequency Centroid and Bandwidth

Table 2.1: The set of aural features described in (Phung, 2001) and used in this thesis.

2.1.1.2 Temporal segmentation

Video data moves beyond still images with its fourth dimension: *time* – and thus, implies temporal relations. The extra temporal characteristics make content-based video systems more complicated than CBIR systems. In addition to *intra-frame* features to describe an image, the system needs to capture the temporal characteristics of the video via its *inter-frame* features. These features provide descriptions of the dynamics of the video and the goal is to capture *temporal regularities*, whereby the video can be segmented into temporally⁴ coherent units. This can be done at multiple resolutions, of which the *shot* remains the most fundamental unit.

Shot Segmentation

A shot is a segment of video resulting from a single run of the camera. A shot boundary can be broadly categorised into either a *cut* or an *optical* transition. A cut is an instantaneous change from one shot to another and by far the most common transition. An optical transition is when two shots are joined by special effects such as a *dissolve*, a *wipe*, or a *fade*.

Detecting cuts has been tackled intensively in the early periods of video analysis research. Cut detection is a relatively simple problem due to the sudden abrupt change at the boundary. There are numerous methods to capture this ‘sudden change’, ranging from the direct raw pixel-comparison to comparisons of the DCT coefficients from MPEG

⁴The term ‘temporal’ here should be understood broadly as an umbrella term for both ‘temporal’ and ‘spatial’ since it means whatever regularities can be extracted with the extra information provided with time.

compressed frames. The central idea is the selection of an appropriate colour space to build the histograms and the construction of a dissimilarity measure between two histograms. Colour histogram comparisons are used in (Yeu and Liu, 1995; Patel and Sethi, 1996), improved with twin-comparisons in (Zhang *et al.*, 1993). A study in Lienhart (2001) shows that local histogram comparison is the most effective and accurate method for shot detection. Improvements to local histogram-based methods include (Yong, 1999). Despite the difference in the methods or domains, cut detection is generally regarded as a solved problem (Hanjalic, 2002). For example, in a recent experiment (Hanjalic, 2002), the author reports an accuracy and precision of 100% for cut detection when experimenting on a complex set of data including movies, soccer games, news and commercial documentaries. Detecting optical transitions is harder than cuts and is still an active research issue. The main difficulty here is how to model the temporal statistics due to the variances in camera operations and object movements.

Most of the existing approaches tackling optical transition rely on statistical models to characterise the transition. Hanjalic and Zhang (Hanjalic and Zhang, 1999b) formulate shot transitions in a probabilistic framework, in which a Poisson distribution is assumed for the distribution of shot length and the statistical measure is based on the motion compensation features. The detection is then performed based on the threshold criteria that minimises the average detection error probability. Similar ideas are used in (Vasconcelos and Lippman, 2000b) where Erlang and Weibull distributions are investigated for shot distributions. In (Liu and Chen, 2002), an eigenspace method is employed to capture the non-stationary statistics of the features. In (Kankanhalli and Chua, 2000), the authors employ a temporal multi-resolution analysis (TMRA)-based algorithm to locate both abrupt (cuts) and gradual transitions (optical). The authors in (Truong *et al.*, 2000b,c) model a dissolve as a combination of chromatic or intensity changes resembling a fade-in with a concurrent fade-out. In (Altunbasak, 2000), Altunbasak models the area where a shot transition occurs, or not, by two Gaussians or two Gamma distributions, based upon which, a threshold to detect shots is formed. Another noteworthy direction is the casting of the shot transition problem into a stateful framework, such as in (Boreczky and Wilcox, 1998; Sanchez *et al.*, 2002). In this setting, the video is assumed to enter different ‘states’ during its running time, which is modeled by some state-spaced probabilistic models, most popularly by the Hidden Markov Model (HMM) and its variants. In (Boreczky and Wilcox, 1998), the authors jointly fuse the features extracted from differences in both visual and aural streams based on a HMM, whose state space is designed to be {shot, fade, dissolve, cut1, cut2, zoom, pan}. The states {cut1, cut2, fade, dissolve} are used to model the transition segments between shots, and {zoom, pan} are used to model the camera operation. After training, an unseen video is segmented into shots based on a Viterbi decoding algorithm given the trained HMM.

In brief, detection of shot transitions is one of the most fundamental problems in video analysis to which most of the early research efforts have been devoted. Several comprehensive surveys and papers for this topic abound (eg: Zhang *et al.* (1993); Ahanger and Little (1996); Gu *et al.* (1997); Lienhart (2001); Hanjalic (2002)). With respect to the chosen domain of educational videos investigated in this dissertation, we found that the task of shot detection adequate for our work using the Webflix software (Mediaware-Company, 1999), in which nearly 100% of detection accuracy is achieved for cuts and around 90% – 97% for optical transitions. Detection of optical transitions, which comprises a very small portion, can be manually corrected where necessary. This helps us to alleviate the errors at shot level being inherently propagated up the hierarchy of semantic constructs.

Scene Segmentation

For many applications, segmenting videos into shots is largely inadequate for the task since a shot alone does not convey much meaning with respect to the semantics of the video. A dialogue scene, for example, due to the frequent switching of the camera, would consist of many shots, where each shot alone will carry little meaning unless all shots are somehow grouped together. Such an issue of mining abstract semantics to move beyond the shots has been the centre of much recent research. There is a fast growing body of work to solve this problem, and depending on the domain of investigation, different names appear such as *scene*, *story* or *episode* for motion pictures; *topics*, *macro segments*, or *story units* for information-oriented videos such as news or training and educational videos; or general term such as *logical story units*. In this review, unless explicitly stated, we shall use ‘*scene*’ as a unified term for all of the aforementioned names.

In its most general definition: ‘*a scene is a sequence of consecutive shots whose contents are unified in terms of time, locale and dramatic structures*’ (Truong *et al.*, 2002b). Clearly different from the problem of shot detection where no *contextual knowledge* (eg: locale or dramatic structures) is required, the problem of scene segmentation is more difficult. Sundaram (2002), for example, mentions the following three challenges in modeling the knowledge of scenes: (a) the variety in directorial styles, (b) the semantic relationship of neighbouring scenes, and (c) the knowledge of the viewer about the world. Because scene segmentation relies on domain knowledge, therefore in many cases, it is solved for specific domains such as motion pictures, news/sitcoms, documentaries, or instructional and lecture videos. As remarked by (Truong, 2004), there is no universal framework that works across all domains: methods to detect topics in news, for example, will generally be inapplicable to detect scenes in a movie and vice versa.

Research into scene boundary detection is growing rapidly and many comprehensive sources on this topic are available (eg: Sundaram and Chang (2000b); Vendrig and Worring (2002); Truong *et al.* (2002b); Snoek and Worring (2000)). What we discuss next

is a general review in this area, focusing on specific domains and general methodology. Our coming reviews on Film Grammar based and probabilistic approaches shall further expand this review in a more appropriate context.

There is a large body of research focusing on extracting scene-level concepts in broadcast programs, in particular news videos (eg: Ariki *et al.* (1997); Ide *et al.* (1998); Huang *et al.* (1999b); Walls *et al.* (1999); Shearer *et al.* (2000); Bertini *et al.* (2000); Snoek *et al.* (2004)). Early work in this domain has focused on extracting *meaningful* events, which is usually cast as a classification problem. In (Shearer *et al.*, 2000), for example, the authors combine a number of visual and aural low-level features together with the concept of shot-syntax to group shots in news videos into different narrative structures and label them {anchor shots, voice-overs, or interview}. Liu *et al.* (Liu *et al.*, 1997; Liu and Huang, 1999) propose an integrated approach to segment news reports from other categories in broadcast programs (eg: commercials, shows) based on the fusion of audio and visual information with different types of classifiers (simple threshold, fuzzy, Gaussian mixture, support vector machine). Ide *et al.* (1998) propose an automatic indexing scheme for television news video, where shots are indexed based on the image content and keywords. Shots in this work are classified into five different categories: speech/report, anchor, walking, gathering, and computer graphics. Caption text information is then used with the classified shots to build the indices.

Segmentation of the *news story* is the second major theme exploited in the broadcast domain. The common underlying method used in these works is the use of explicit ‘rules’ about the structure of news programs to detect the transitions between news story. For example, a commonly accepted rule, as remarked by (Truong, 2004), is that a news story often starts and finishes with anchor-person shots; Aigrain *et al.* (1998) observe that a start of a news story is usually coupled with music; or Wang *et al.* (2003) notice that a relatively long silence period is the indication of the boundary between two news stories. (Zhu *et al.*, 2001) utilise the results from anchor-person and caption detection to form a set of rules for news story boundary detection (eg: if the same text caption is used in two consecutive anchor-person shots, then they belong to the same news story). There is also a body of work which casts the segmentation problem of news story in a HMM framework where the domain knowledge can be explicitly used. Iurgel *et al.* (2001), for example, propose the news story segmentation as a problem of decoding maximum state sequence of a trained HMM whose topology is designed by incorporating explicit knowledge about news program. Discussion on this approach is further included in Section 2.3.3.2 and Section 2.3.3.3 when we review the HMM-based approaches for video analysis in Section 2.3.3.

Recently, there has been growing research interest in extracting scenes in motion pictures (eg: Zhang *et al.* (1993); Yeung *et al.* (1996); Rui *et al.* (1998); Sundaram and Chang

(2000b); Wang *et al.* (2001); Adams *et al.* (2001); Truong *et al.* (2002b)). Detecting scenes in motion pictures is challenging and there are three main existing approaches:

- *Temporal clustering-based detection.* In this approach, shots are clustered into scenes based on visual similarity and temporal closeness, and includes the work of (Yeung and Liu, 1995; Yeung and Yeo, 1996; Rui *et al.*, 1999; Hanjalic *et al.*, 1999a,b; Lin and Zhang, 2000).
- *Rule-based detection.* Scene breaks in this method are detected based on the semantic analysis of audiovisual characteristics and in some cases further enhanced with cinematic rules and conventions (Huang *et al.*, 1998; Wang *et al.*, 2001). Wang *et al.* (2001) attempt to extract scenes in movies using visual similarity and then further improve it with guidance from cinematic rules. Another interesting piece of work is of (Adams *et al.*, 2000, 2002b), in which a *tempo* function is computed, and then used to segment a movie into story units based on the ebb and flow of this function.
- *Memory-based detection.* In (Sundaram and Chang, 2000a,b, 2002), the authors integrate visual and audio clues for scene detection. Visual shot similarity is determined based on whether or not a group of shots is consistent chromatically (colour), and audio features are combined to detect “audio scenes”. Visual and aural data are then fused within the context of a memory and attention span model to find likely segmentation or singleton events.

2.1.2 Video Representation and Summarisation

A video could consist of hours of running time, and thus it is crucial that the video can be compactly described automatically without overlooking important details, much analogous to providing abstracts and keywords for text documents. This process is generally known as video summarisation or abstraction. Methods for video summarisation are categorised in (Truong, 2004) into two main groups: (1) keyframe-based abstraction, and (2) video skimming.

Keyframes-based Summarisation

In this scheme, the video content is summarised by a set of keyframes, also known as representative frames. At the simplest level, keyframes can be constructed by uniformly sampling the video sequence (Taniguchi, 1995), or selecting the first frame of every shot as the representative (Smoliar and Zhang, 1994). Clearly, frames selected in these ways do not richly reflect the video content, and more advanced methods have been developed. In (Ngo *et al.*, 2001; Truong, 2004), the authors summarise three main mechanisms for keyframes extraction:

- *Dissimilarity in visual content.* In this framework, starting from the first frame, a new keyframe is selected sequentially as soon as a new frame is declared to be significantly different from the previous keyframe. The main component here is the similarity function to measure the content change, eg: histogram difference (Xiong *et al.*, 1997; Günsel and Tekalp, 1998), change in the number of objects and region histogram (Zhang *et al.*, 1997), or a combination of both (Kim and Hwang, 2000).
- *Clustering-based extraction.* In this approach, a feature space is constructed, in which a point in the feature space corresponds to a frame. These points are then clustered into groups and keyframes are selected from those groups (Hanjalic and Zhang, 1999a; Drew and Au, 2000; Yu *et al.*, 2004). There is also a slight modification to this approach called *curve simplification*, in which instead of forming clusters, a set of points is constructed in a way that the “removal of remaining points least changes the shape of the curve connecting all points through their temporal ordering” (Truong, 2004). Works that follow this direction include (Hanjalic *et al.*, 1998; Doulamis *et al.*, 1999; Divakaran *et al.*, 2002).
- *Important events based extraction.* Rather than looking for visual similarity, this approach selects a keyframe based on the trigger of semantically important events such as when the motion curve reaches a local minima (Wolf, 1996; Calic and Izquierdo, 2002) or the attention curve reaches the local maxima (Ma *et al.*, 2002).

Besides the issue of which mechanism is to be used, keyframes can be extracted at the shot or clip levels. In the former, shot indices are first detected, and keyframes are extracted to reflect the content of each individual shot. The advantage of this approach is that it can preserve the *temporal visual progression* and represent mainly *local neighbourhood consistency* (Truong, 2004) such as the works of Xiong *et al.* (1997); Kim and Hwang (2000). Extraction of keyframes at the clip level does not require shot indices in advance, and tends to generate a smaller number of keyframes while preserving the global consistency such as the clustering-based keyframes extraction methods in (Zhuang *et al.*, 1998; Yu *et al.*, 2004).

Video Skimming

The objective of video skimming is to ‘collect’ important and relevant events from the video to generate a (much) shorter version of a video without overlooking the crucial details. Movie trailer are examples of video skimming. However, in many cases, commercial movie trailer generation is not the objective of video skimming since they are *subjectively* generated for advertising purposes, and thus do not necessarily reflect the content of the videos. Moreover, they are rarely generated automatically⁵. The main goal of video skim-

⁵There are few works that seek to generate movie trailer automatically, eg: the *VAbstract* system (Pfeiffer *et al.*, 1996)

ming, on the hand, is to automatically *summarise* and *represent* the content succinctly. While simple video skimming mechanisms such as uniform sub-sampling of the video, or the juxtaposition of the keyframes and its neighbouring frames (eg: Wu and Kankanhalli (2000)) are efficient, they are still ‘missing’ the coherence in the content and are hard to comprehend. The problem of video skimming is thus generally regarded as difficult *since it requires high level content analysis*, and their methods can be categorised into three main groups (Truong, 2004): (a) features-based skimming; (b) redundancy elimination skimming; and (c) events-based skimming. We refer readers to (Ngo *et al.*, 2001; Truong, 2004) and references therein for a complete treatment of video skimming.

2.1.3 Video Searching and Retrieval

As a subproblem of information retrieval, searching and retrieval of videos are concerned with algorithms to retrieve *relevant* content given a query input from the user. Ngo *et al.* (2001) identify two popular retrieval problems:

- *Retrieval by similar videos.* This problem is cast as the video genre identification and present the highest level of video classification. The objective is to retrieve high-level content description of the videos, which usually organised in hierarchy. The top level may include general genres such as {movies, news, cartoons, documentaries, etc.}. At the lower level, the movie genre, for example, is further classified into subgenres such as action, comedy, violent, etc. Examples of work in genre identification include (Fischer *et al.*, 1995; Truong *et al.*, 2000a; Phung *et al.*, 2002; Taskiran *et al.*, 2003).
- *Retrieval of similar clips in a video.* A clip in this context is either: a) a single shot, or b) a set of shots describing a unified event (and thus similar to the concept of scene). Retrieval of videos at clip level is also known in the community as Query-by-Video-Clip (QVC) paradigm (Jain *et al.*, 1999; Liu *et al.*, 1999; Tan *et al.*, 1999; Yuan *et al.*, 2004). The key question is how to define a similarity measure between the query clip input from the user and a large collection of video segments in the database. As noted in (Ngo *et al.*, 2001), the similarity measure can be defined either locally (eg: temporal alignment and matching via dynamic programming in Tan *et al.* (1999)) or globally (eg: distance between two feature vectors defined globally for two clips in Yuan *et al.* (2004)).

Ngo *et al.* (2001) and Adams (2003b) cover a fairly detailed discussion of similarity measures in video databases, and we refer the reader to these references for further details. To conclude this overview on video searching and retrieval, we emphasise that this area

is still widely acknowledged as a difficult problem, and as remarked in (Ngo *et al.*, 2001), video retrieval possesses the same challenge as image retrieval, that is “low-level features for retrieval do not match human perception well”; in other words, again *the ‘gap’ between low-level features and high-level concepts is the main obstacle.*

2.1.4 Current Challenge – the Semantic Gap

In summary then, the most fundamental challenge in video content analysis lies in ‘closing the semantic gap’ – the final frontier identified in several papers, eg: Smeulders *et al.* (2000); Dorai and Venkatesh (2001a); Jain (2001); Naphade and Huang (2002); Dorai *et al.* (2002); Adams (2003b). As we have reviewed, while shot indices can be effectively detected, the problem of scene segmentation still remains difficult and is challenged by the semantic gap. All other facets including summarisation, representation, searching, browsing or retrieval of video data, again rely on the solutions to bridging-the-semantic-gap problem. It is the problem of “connecting data with meaning” (Adams, 2003a, p.30).

How then can the semantic problem be approached? Naturally, to understand the data one needs to examine the *creating force* behind it. Adams (2003a) remarks: “to define one’s domain is no less than identifying (or at least constraining) the candidate forces responsible for the data under investigation”, or stronger: “to create tools for automatically understanding video, we need to be able to interpret the data with its maker’s eye” (Dorai and Venkatesh, 2001b). This philosophy embarks on a new direction in seeking solutions for the semantic gap. Two key components are identified:

- a systematic method to understand the “maker’s eyes” and
- mechanisms for interpretation of the data.

In the problem of ‘understanding the maker’s eyes’, there are arguably certain ‘rules’ and conventions that govern the generating process of a video production – more formally known as *Film Grammar*, a branch of film analysis that elucidates the practices used by directors worldwide to convey ideas and meanings. Our first approach towards understanding video content is based on Film Grammar, which we will henceforth refer to as Film Grammar based methods. Section 2.2 provides a review of this area.

The second main research effort towards the semantic problem is in seeking advanced methods to interpret data. In this area, advanced statistical methods are harnessed, mainly using probabilistic models. We will review probabilistic methods in video analysis in Section 2.3.

2.2 Understanding Video Content through Film Grammar

There is growing interest in using Film Grammar as the underlying principle to understand video content. So, what is Film Grammar? The literature on film analysis has several slightly different definitions for Film Grammar⁶. Arijon (1976) summarise its essence:

Film Grammar – A collection of rules and conventions that are the product of experimentation, an accumulation of solutions found by everyday practice of the craft (Arijon, 1976, p.2).

In other words, Film Grammar provides us with a corpus of commonly used techniques, or guidelines, that convey purpose and meaning. These rules and conventions in media production have intensively been utilised and exploited, mainly in seeking high level semantic descriptions, annotation and segmentation. We thus organise the review in this section as follows. In Section 2.2.1 we discuss some ‘grammars’ that are peculiar to educational films. Next, in Subsection 2.2.2, we review a body of work that uses Film Grammar to assist video content annotation. This is followed by Subsection 2.2.3 in which the problem of segmentation using Film Grammar is reviewed. Finally, we present the Computational Media Aesthetics framework and discuss previous works that have used this framework as an underlying principle to approach the semantic gap problem in Subsection 2.2.4.

2.2.1 Film Grammar for Educational Videos

Feature films and educational films have many differences. The axiomatic distinction is that the purpose of feature films is to *entertain* whereas the goal of education/training films is to *teach* and *train*. Thus, a rich array of techniques and grammars is usually exploited in feature films to achieve ‘entertaining’ elements such as story dramatisation, evoking emotions (fear, sadness, anger), horror sound effects, etc. Grammars used in educational films are in somewhat more contained set, and while sharing certain aesthetic elements with feature films (eg: event dramatisation, special sound effects), they must be used in a way that conforms with the desirability of *clarity* and *unambiguities* for instructional and educational purposes (Herman, 1965). To provide further background on education and training films, we shall now outline key aesthetic elements that constitute the grammar

⁶There are also several debates on whether the grammar in film is the same as the usual grammar in languages such as English. While this topic is still debatable, our view towards Film Grammar is that there are common rules and techniques used in visual arts to achieve specific effects. We refer to these ‘rules’ as the ‘grammar’, and whether or not these rules conform with strict syntactic checking as in linguistics is a different issue.

for this genre. More specific production grammars, in relevant contexts, used in our work are further discussed in Chapter 3 and Chapter 4.

Visual and Aural Ingredients

In the visual stream, educational films share many similarities with news and documentaries. While it is almost impossible to enumerate all types of visual modes in feature films, the visual information in education and training films can be mainly categorised into the following main components (Rabiger, 1998; Herman, 1965):

- *Anchor section* – instructions are delivered in a direct manner such as the camera showing the teacher talking (similar to anchor shots in news).
- *People talking* – to each other, either acknowledging the presence of the camera whilst talking or assuming that the camera is absent during their verbal exchange, such as interviews or voice-of-experience sections (where people talk about their experiences in safety/training videos.)
- *Action footage* – people or creatures doing things, carrying on their daily activities, work, play, and so on; shots of landscapes and inanimate objects. In training videos, action footage (in the form of re-enactments) is also often used to dramatise, or intensify a safety message.
- *Superimposed texts* – artificial texts in video frames. Superimposed text is an important element in conveying messages in educational videos.
- *Graphics and artwork*: still photos, line art, cartoons, and other graphics.
- *Library footage* – can be uncut archive material or sequences recycled from other films.
- *Blank screen* – causes us to reflect on what we have already seen or to give heightened attention to the soundtrack.

The secondary information channel in the film medium is the soundtrack. Its significance is even more emphasised in educational films. The audio in these films typically falls into two classes: one is *speech* to convey instructions and teaching concepts, the other, such as *music* or *expressive silence*, is to create ‘mood’ with either ambient background audio or sound effects.

Editing Elements in Educational Films

There are certain basic principles and practices that must be adhered with in order to edit shots into cinematically effective scenes, subtopics, topics, sequences and the film as

a whole. This is known in film analysis as *montage*. In an educational context, it is the act of using definite grammar of accepted editing concepts and principles to organise the content meaningfully and cohesively. Herman (1965) outlines the following principles in educational films to maintain the professionalism in editing:

- **Continuity.** The editor must first organise the content for continuity, that is, according to Herman (1965), it is “only with this element of continuum that a film can possess the compelling motion and movement unique to a motion picture”.
- **Flow and inter-relationships.** The succession of shots/scenes/sequences must flow smoothly and continuously to create a unified, coherent and consistent quality for a teaching film. To achieve the smooth flow, the editing process must conform to inter-relationship organisation, that is, for example, shots within a topic segment must be *meaningfully* juxtaposed to convey and form a *complete* discussion of that topic. Three building principles to maintain the smooth flow are: (a) *progression*, (b) *contrast*, and (c) *repetition*.

The *progression* principle requires that filmic images move in an orderly sequence, from beginning to end, in a coherent unified flow. This principle, for example, would imply that a topic or teaching lesson in educational films should be crafted by having an introduction, a body, and a conclusion. Besides being progressive, the flow should depict image action by *contrasting* the effects of motion and movement, for example by “opposing one quality with another so that the sense of action is suggested rather than effected” (Herman, 1965). Motion and movement, when not actually depicted, can also be implied by “simply *repeating* certain qualities, the sense of action resulting from the dynamics of mere repetition” (Herman, 1965).

2.2.2 Video Annotation with Film Grammar

The first major application of Film Grammar is the exploitation of its ‘rules’ and ‘principles’ in a specific domain for content annotation and labeling. Yoshitaka *et al.* (1997) exploit Film Grammar to annotate the semantic content of scenes for video retrieval. In this work they define the grammar of film as “the accumulation of knowledge in order to express or emphasise a certain semantic content effectively. It relates to camera framing, editing techniques, and the dynamics of the shots”. Three types of semantics were considered {conversation, tension rising, and action}. Three kinds of ‘grammar’ are utilised: editing patterns for a combination of shots, the visual dynamics of shots, and shot combinations of similarity. Based on this set of rules, a heuristic algorithm is proposed to

detect the repetition of similar shots, and then grouped into scenes of conversation, tension rising or action. No results, however, are reported to evaluate the detection scheme. While Film Grammar is exploited, this work is restricted only to the grammar of the three chosen dramatic events. This work is then concerned with developing a retrieval scheme for the content, in which the experiments report around 70.8% correct retrieval, 29.2% wrong retrieval and 10.5% ‘miss’ retrieval for conversation scenes. These figures are {61.5%, 38.5%, 5.8%} and {47.7%, 52.6%, 0%} for tension rising and heavy actions scenes respectively. Rasheed *et al.* (2003a,b), inspired by Film Grammar, exploit a set of low-level features including shot length, colour variance, motion content and lighting key to provide a mapping to four types of films, namely comedy, action, dramas and horror. Moncrieff *et al.* (2001b,a) study aspects of audio ‘grammar’ to annotate violent and car chase scenes in movies.

2.2.3 Video Segmentation with Film Grammar

The second major exploitation of Film Grammar is to solve the problem of video segmentation. Ferman and Tekalp (1997) exploit a set of editing cues to generate meaningful semantics from the video stream. In Aigrain *et al.* (1998), the authors employ a set of ‘macro-knowledge-based’ rules to detect macro segments (scenes) from the video stream. They exploit six types of rules, namely: transition effect rules, shot repetition rules, continuous shot setting similarity rules, editing rhythm, soundtrack rules, and camera work rules. The rules used are stated explicitly and unambiguously. A camera work rule, for example, states that “if there is a succession of three or more simple shots with the same camera work (other than still shot) then they belong to the same sequence”. Based on these rules, the macro-segmentation is based on three steps: (a) *merging* – identify starting and ending points for segment and break points, (b) *precedence* – resolve conflicts between segment and break points in the previous step using precedence rules; and (c) *hole filling* – adding segments for any missing parts. This piece of work is interesting in the sense that it presents a nice exploitation of explicit grammar to group shots into meaningful, macro segments. The paper, however, only presents an example of their scheme on one documentary and a few TV shows, and no evaluation was rigorously conducted to validate its usefulness as well as its generalisation to other domains.

Radev *et al.* (1999) combine film theory with the graph-based object oriented data model and propose a general film model that represents structural, semantic, and syntactic elements of film. This model, in essence, is a conceptual schema graph whose nodes are the basic film structural elements and features derived from an analysis of film theory studied in (Monaco, 1977). Logical relationships between these elements are represented by edges between corresponding nodes. While no experimental results are represented and

no methods are proposed for automatic classification or segmentation with their model, their study is highly original, showing the possibility of deriving rigorous frameworks for video modeling from the film literature.

In (Chen and Ozsu, 2002), the authors utilise a set of editing rules to extract simple *dialogue* and *action* scenes. Further constraining the case, they consider at most two actors in the scene. The authors propose three types of shots found in a dialogue scene between actors X and Y : (A) a shot in which only the face of X is visible, (B) only the face of Y is visible, and (C) both faces of X and Y are visible. Augmented with the fact that an ‘insert’ or ‘cut-away’ shot (type D) is occasionally introduced during the scene, the vocabulary of shot styles is represented as $\{A, B, C \text{ or } D\}$. The same set of shot styles is used for simple action one-to-one fighting scenes. Next, the authors use the following two-step ‘grammar’ for a dialogue scene to construct a Finite State Machine (FSM): (1) setting up the dialogue scene, and (2) expanding the dialogue scene. Different ‘rules’ are applied to constrain the arrangement of shot types during these two steps. For example, in the expanding-scene stage, A may only be followed by B or C . A FSM is then constructed based on these editing rules for a regular language that consists of all possible video shot strings (arbitrary juxtaposition of shot labels that follow the editing rules) for the dialogue scene. Similar techniques are used for action scenes. A dialogue scene is then detected if a path, starting from the ‘start’ state and ending at one of ‘end’ states, can be found. Even though it appears from the FSM that there is no ambiguity in the language (ie: there is a unique parsing path for the given string), it is not formally addressed by the authors. Nevertheless, using a data set consisting of three movies *Gladiator*, *Crouching Tiger/Hidden Dragon* and *Patch Adams*, their experiments report an average precision of 87.3% and recall of 94.8% for dialogue scene detection, and 80.3% of precision and 82.8% for action scene detection.

Using similar ideas, Zhai *et al.* (2004) propose the use of a FSM for detecting and classifying three types of movie scenes: conversation, suspense, and action. Utilising a mid-level feature derived from a face detector, their FSM is much simpler than that of (Chen and Ozsu, 2002). Similar approaches using FSM are found in (Merlino *et al.*, 1997).

2.2.4 Computational Media Aesthetics and High Order Construct Extraction

In the work reviewed above, although the knowledge of Film Grammar is utilised and this generally results in an improvement in performance, it has not been systematically utilised and exploited. Recently, the *Computational Media Aesthetics* (CMA) framework was proposed to systematically use Film Grammar as its foundation (Dorai and Venkatesh,

2001b, 2002). Inspired by the pioneering works of Zettl (1999) on *applied media aesthetics*, the authors define an algorithmic framework in which Film Grammar is systematically exploited:

Computational Media Aesthetics – “The algorithmic study of a number of image and aural elements in media and the computational analysis of the principles that have emerged underlying their use and manipulation, individually or jointly, in the creative art of clarifying, intensifying, and interpreting some event for the audience.” (Dorai and Venkatesh, 2001b).

By placing a strong focus on emotional, aural and visual appeal of the content, the CMA framework allows the extraction of semantic and semiotic information by the investigation of the relations between cinematic elements and narrative form (Dorai and Venkatesh, 2002). Figure-(2-5) depicts the CMA framework. At the *primitive feature extraction* stage,

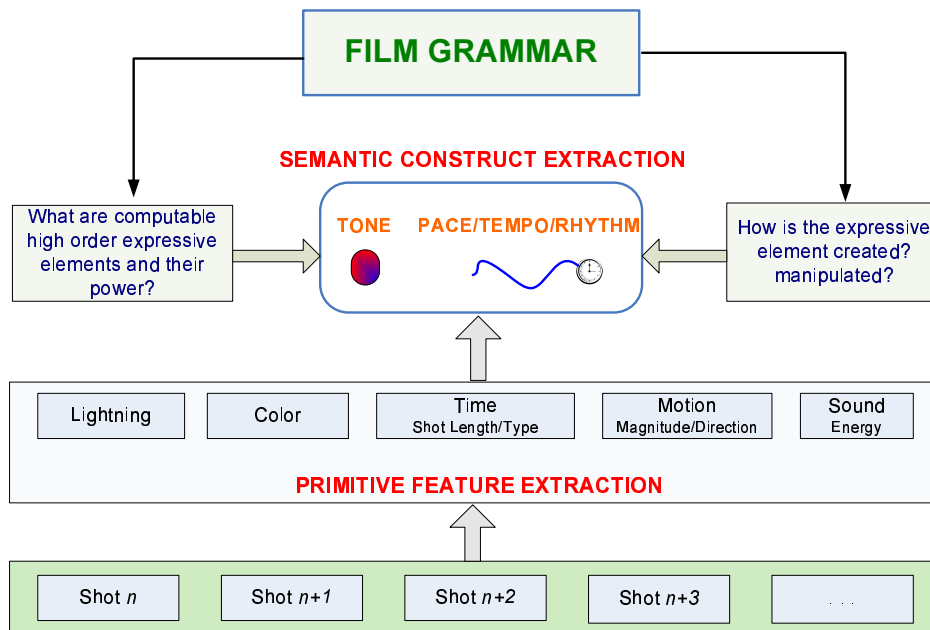


Figure 2-5: The Computational Media Aesthetics framework, adapted from (Dorai and Venkatesh, 2002, ch.1).

computable audio and visual elements are extracted such as lighting, colour, motion, sound energy and so on. What sets CMA apart from other existing approaches is at the *semantic construct extraction* phase: “that is, to both define what to extract and how to extract these constructs we seek guidance and logic from film grammar. We do so because directors create and manipulate expressive elements related to some aspect of visual or emotional appeal in particular ways to have maximum impact” (Dorai and Venkatesh, 2002). Film Grammar plays the central role at this stage. It provides insights into the video creation

process, and tells us not only which expressive elements are manipulated by the filmmaker, but also why and what the intended meaning and impact is. One of the main efforts in exploiting Film Grammar is to seek *expressive functions*, or high order semantic constructs, to describe the content.

Closely related to our work on the extraction of high order constructs for educational and training videos in Chapter 4, we shall now review two expressive functions that have been exploited in previous work, namely the visual pace function (Adams *et al.*, 2000, 2002b), and the audio pace function (Moncrieff, 2004).

The visual pace function

One of the important aspects of film expression is the *tempo* or *pace* function which was systematically investigated in the works of Adams *et al.* (2000, 2002b). Initially, the author used a set of three long feature movies⁷ with a *manual* list of story sections groundtruthed as “fast” or “slow” as the indication of the tempo. Average shot motion and shot length were used as the contributing elements for the pace function. Half of the data was used to train a decision tree classifier and the rest was used as unseen data. The conclusion drawn from this initial study was that while the learned decision tree can well separate between “fast” and “slow” categories (23 correctly classified out of 26) in this case-study, the resolution issue is a problem with this classification scheme, ie: the boundaries between fast and slow sections will break down upon the addition of the remainder of data from the film, for example the sections that are neither fast nor slow (Adams, 2003a). Another drawback is that it is unable to provide an ‘intuitive’ and ‘smooth’ feel for the pace. Resolving these two issues, the author proposed a continuous measure function for the pace, constructed upon two contributing factors: *motion* and *shot length*, and formulated as:

$$P(n) = \frac{\alpha (\text{med}_s - s[n])}{\sigma_s} + \frac{\beta (m[s] - \mu_m)}{\sigma_m} \quad (2.1)$$

where s is the shot length (in number of frames), m is motion magnitude, and n is the shot number; σ_s and σ_m are the standard deviation of shot length and motion respectively; μ_m and med_s are the motion mean and shot length median respectively; and the weights α and β have the values of 1, which essentially means that both shot length and motion contribute equally to the perception of pace. The pace function is usually smoothed with a Gaussian filter. The contributing factor *shot length* corresponds to a cinematic technique of *montage*, or editing, which is used by the director to manipulate the speed at which the viewer’s attention is directed. Motion, on the other hand, is influenced by the on-screen dynamics and, thus, affects the perception of tempo. Short shot length and high motion content will therefore result in higher pace, and vice versa. The tempo function is then exploited to detect story sections and dramatic events (Adams *et al.*, 2000, 2002b), and the

⁷Titanic, Lethal Weapon 2, and The Color Purple.

automated extraction of film *rhythm* is exploited for scene content understanding (Adams *et al.*, 2001, 2002a).

The Audio Pace Function

In parallel with the works of Adams (2003a) on visual pace, Moncrieff (2004) investigate the aspect of *audio pace* in films. The concept of audio pace is derived from the elements of Film Grammar ‘that influence the perceived pace of the film through the properties of the audio’ (Moncrieff, 2004, p.119). The audio is usually coupled with visual aspects to maintain the synchronisation of the audio visual elements, and thus attributes more meaning to the visual domain. When disparity between audio and visual elements exist, this corresponds to cinematic techniques used to intensify the significance of certain events. Moncrieff (2004), therefore, remarks “it is the parity and disparity between the audio and visual aspects of film, guided by Film Grammar techniques, that influences the audio pace”.

To construct the audio pace function, the author considers two types of features. The first consists of a set of features derived from the energy and frequency of the audio signals, and the second set is constructed to encode the dynamic behaviour of the audio. In the former, two audio features are developed at the shot level including the *audio energy*, calculated as a smoothed version of the average audio shot energy, and the *fundamental frequency* computed by tracking the cepstral peaks. In the second set of features, to capture the dynamic characteristics of the audio content, Moncrieff (2004) proposes the use of *audio entropy* and *audio type*. In essence, the audio entropy⁸ is calculated by the similarity, or dissimilarity, between two successive audio frames of fixed duration, analogous to the visual frame by frame difference used to determine motion. Each audio frame is treated as a vector of features and the difference between two audio frames is captured by the cosine value of the angle between the two vectors. Equation-(2.2) shows the difference function and the computation for the entropy, where ς_{i-1} and ς_i are feature vectors representing two consecutive audio frames:

$$\begin{aligned} \Delta(\varsigma_i, \varsigma_{i-1}) &\triangleq \frac{\varsigma_i \cdot \varsigma_{i-1}}{\|\varsigma_{i-1}\| \times \|\varsigma_i\|} \\ \text{Entropy}(\varsigma_i) &\triangleq 1 - \Delta(\varsigma_i, \varsigma_{i-1}) \end{aligned} \quad (2.2)$$

Audio entropy is first calculated uniformly for the entire film and then mapped into a shot-based feature by averaging audio frames for that shot, excluding speech frame. When examining the dynamic behaviour of the audio content by audio type, Moncrieff (2004) classifies each audio segment into one of the four labels: music, speech, low sound energy and other, and then combines them to construct shot-based features. One of the combi-

⁸We note that the term ‘entropy’ used in (Moncrieff, 2004) has a different meaning to the term ‘entropy’ used in information theory.

nation methods, for example, is by counting the number of changes in the audio labels within a shot as a feature for that shot. Given the set of primitive features, the audio pace function is constructed as a linear combination of the features:

$$\text{AudioPace}(n) = \alpha f_1(n) + \beta f_2(n) \quad (2.3)$$

where α and β are the weights. Using the audio pace function, Moncrieff (2004) proposes a framework for narrative detection by examining the significant changes in the audio pace.

Related work to the extraction of expressive elements also include the works of (Colombo *et al.*, 1999; Truong *et al.*, 2002a; Truong, 2004). These authors investigate the aspect of colour semantics in videos. Colombo *et al.* (1999) study the expressive and emotional functions through colour and construct a framework for semantic retrieval of art paintings and commercial videos. Truong *et al.* (2002a) investigate the contribution of colour semantics to express two aesthetic elements for visual appeal, namely *adding excitement/drama*, and *establishing mood*. Colour is represented in HSV space at the shot level and linearly combined to form a continuous measure for the colour expressiveness. Edge detection is then performed to segment the movie into scenes.

2.3 Probabilistic Approaches to Video Content Analysis

In a broader context, statistical pattern recognition (PR) techniques are used almost at every phase from low-level processing to abstraction and retrieval. We refer readers to the surveys of (Jain *et al.*, 2000; Antani *et al.*, 2002) for more a comprehensive review on the application of pattern recognition techniques to the field of video content analysis. Closely related to our study, we shall review existing research attempts that base their work on two most popular probabilistic models used in video content analysis, namely the Bayesian Network in Section 2.3.1 and the Hidden Markov Model in Section 2.3.3.

2.3.1 Bayesian Network Approaches to Video Analysis

The Bayesian Network (BN) (also known by other names such as *belief networks*, *casual probabilistic networks*, or *influence diagrams*), pioneered by Judea Pearl, is a widely used probabilistic model, most popularly in the area of Artificial Intelligence. It provides a formal probabilistic framework for modeling *casual* interactions (without loops) among many random variables, in particular for building models of domains with inherent uncertainty (Jensen, 1996). The applications of BNS are vast and we do not attempt to cover

them here, instead we refer readers to Pearl’s latest book on this topic (Pearl, 2004) for further details. The rest of our review on Bayesian approaches for video analysis is structured as follows. We briefly introduce the Bayesian Network in Subsection 2.3.2. Our goal is to familiarise readers with a necessary background in BN, especially the *d-separation* procedure, which will be used frequently to provide proofs for many theorems put forward in Chapter 5. The applications of BNs for video analysis are subsequently reviewed in Subsections 2.3.2.3, 2.3.2.1, and 2.3.2.2.

2.3.2 Bayesian Networks and *d*-Separation

A Bayesian Network (BN) is an acyclic directed graph $G = \{V, E\}$, where each node $x_i \in V$ represents a random variable (RV), and each directed edge $u_{ij} \in E$ from node x_i to node x_j represents a casual relationship between the two variables, in which x_i is the parent of x_j . A node x_i in a BN is thus associated with a set of its parental nodes $\text{pa}(x_i)$, and parameterised by the conditional probability $\Pr(x_i \mid \text{pa}(x_i))$. Assume that there are N variables associated with the network, then the BN represents a *joint probability distribution* over the set of all variables $\{x_1, \dots, x_N\}$, which can be conveniently factorised into a product of the *local* conditional probability forms:

$$\Pr(x_1, \dots, x_N) = \prod_{i=1}^N \Pr(x_i \mid \text{pa}(x_i)) \quad (2.4)$$

One of the central issues in the BN is the *inference* problem, that is to compute marginal or conditional marginal distributions such as $\Pr(U)$ or $\Pr(U \mid H)$ where U and H are subsets of V . For example in the retrieval problem, H is the set of observed variables and U is the semantic concept being queried. The most useful feature of BN is its local factorisation form (given in Equation-(2.4)) that enables the inference problem to be done in a standard *junction tree* algorithm, which typically involves the following steps: (a) the moralisation of the BN to convert it to an undirected graph, which is then (b) triangulated and converted to a junction tree, and finally (c) a message passing procedure is performed on the clique tree. At the end of the algorithm, marginals and conditional marginals can be conveniently obtained from the clique and separator potentials on the junction tree. The complexity of this algorithm is in the order of the maximum clique size in step (b). We refer readers to (Pearl, 1998; Jensen, 1996) for further details on this algorithm and related discussions.

For learning or parameter estimation in BNs, two cases are considered⁹. In the *fully observed case*, the maximum likelihood (ML) parameter estimation solution for a local

⁹Note that we are implicitly referring to the discrete case.

conditional probability $\hat{\Pr}(x_i \mid \text{pa}(x_i))$ becomes the *normalised marginal counts* over all possible configurations of $(x_i, \text{pa}(x_i))$. In the *hidden variables* case, the parameter estimation is achieved by the Expectation Maximization (EM) algorithm. The E-step is viewed as an inference problem in the BN, and informally can be viewed as the process of ‘filling’ a missing variable by its expectation, which is then treated as the fully observed case. We refer readers to (Jordan, 2004) for further details. In Chapter 5, we will touch on these issues again, but will specifically target the problem of parameter estimation for the Hierarchical HMM.

Another useful property of Bayesian Networks is that conditional independencies among variables can be asserted directly from the graph topology without the need to examine the parameters. This is commonly known as the *d-separation* assertion in BN. In Chapter 5, this assertion criteria will frequently be used in many proofs, we therefore briefly describe it here. For more complete details and examples, readers are referred to (Pearl, 1998; Jensen, 1996). In essence, *d-separation* is a set of analytical rules for asserting the independence relations among variables on a BN purely by examining the properties of the graph. To assert a conditional independence property P :

$$X_A \perp\!\!\!\perp Y_B \mid Z_C,$$

(ie: X_A is conditionally independent with Y_B given the knowledge of Z_C), the *Baye’s Ball* algorithm is used. That is, first all variables in Z_C are shaded (indicate that they are observed), and if a ball starts from any variables in X_A can reach one of the variables in Y_B , then property P is invalidated, otherwise it is validated. The rules for bouncing the ball (eg: see Jordan (2004), Jensen (1996) (ch.2)) in the *canonical* case are shown in Figure-(2-6), which consists of four canonical graphs: (a) *Explaining-Away* – or converging connection, (b) *Hidden Cause* – or diverging connection, (c) *Markov-Chain* – or serial connection, and the last graph (d) is the special case when the ball reaches one of the leaves. In each case, the direction of the arrow indicates if the ball can pass through or will be bounced back. For further details and examples of this algorithm, we recommend (Jensen, 1996, ch.2), and (Jordan, 2004).

Next, we organise the review of existing research attempts that use the Bayesian framework for video analysis into three main groups. In Subsection 2.3.2.1, we review existing Bayesian Network approaches to video annotation problem, followed by the BN-based extraction of semantic concepts in Subsection 2.3.2.2. Finally, we discuss the use of BNs in the problem of video retrieval in Subsection 2.3.2.3.

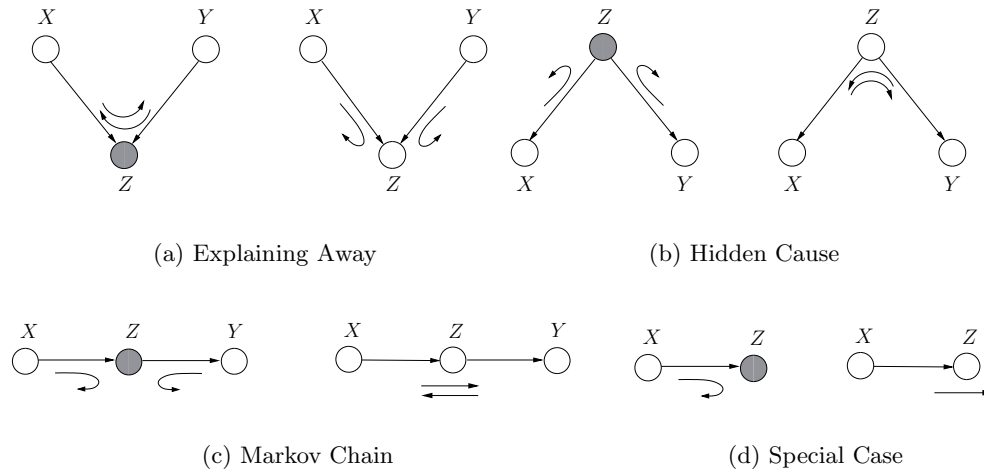


Figure 2-6: Rules for bouncing a ball in the Baye's Ball algorithm, adapted from (Jordan, 2004).

2.3.2.1 Bayesian Network approaches to scene content annotation

In the BMoViES¹⁰ system, Vasconcelos and Lippman (1998c) aims to use the Bayesian Network to infer four types of content: *Action*, *Close-up*, *Crowd*, and *Setting* from a set of three low-level visual features including *motion energy*, *skin colour*, and *texture energy*. The framework thus consists of two layers. At the bottom layer, simple visual sensors are used to detect visual features, and each is modeled as a random variable. At the top level, a probabilistic network is constructed to infer the state of a content type. Prior knowledge about the structure of BNs, ie: the relationships between a content type and low-level feature set, is incorporated directly as shown in Figure-(2-7).

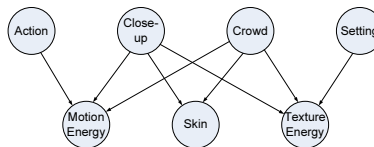


Figure 2-7: The BN used in (Vasconcelos and Lippman, 1998c) to infer high-level content types from low-level sensory information.

Given the observation O from the visual sensors, the detection of a content type S is simply translated into an inference problem of computing $\Pr(S | O)$ in a BN. This framework is revised and extended in (Vasconcelos and Lippman, 1998b), in which the authors report an accuracy of 98.7% for *Action*, 88.2% for *Close-up*, 85.5% for *Crowd*, and 86.8% for *Setting*

¹⁰Bayesian Modeling of Video Editing and Structure.

when testing on a dataset of around 100 video clips extracted from the movie ‘Circle of Friends’. One drawback of this framework, however, is that it is not robust under lighting conditions and its strong association with the skin model and the presence of humans (which would be invalid, for example, if a character is painted).

In (Jasinschi *et al.*, 2001), the authors describe a Video-Scout system, in which a Bayesian framework is formulated at the centre of the system to infer high-level content descriptions of TV programs such as ‘Talk’, ‘Financial news’, or ‘Commercials’. The core of the Video-Scout system is the Segmentation-and-Indexing module, which is formulated in a three-layered probabilistic Bayesian Network, called the *Bayesian Engine*. Similar Bayesian approach is used in (Nitta *et al.*, 2002) in the sports domain to classify football shows into *live, replay*. The game is then segmented into logical unit by identifying the first live shots in a sequence of contiguous live shots.

2.3.2.2 Bayesian extraction of semantic concepts

Ferman and Tekalp (1999) propose a probabilistic framework, combining Hidden Markov Models and Bayesian Networks, for mapping low-level visual features to a set of semantic descriptors. The framework architecture consists of two components: HMMs are used at the shot and sequence levels to model the temporal characteristics of video and group shots into coherent sequences; and a number of BNs are used to map shots into a set of semantic concepts. A set of 11 low-level features is extracted from the visual stream and concatenated to form a single feature vector for each shot. A Hidden Markov Model with five states is then used to model a dialogue scene, and given the trained HMM, a shot is annotated based on the Viterbi decoding results. Using this scheme, they claim that the assigned labels, after a Viterbi decoding, accurately reflect the nature of the shots (eg: establishing, master shots, etc.) when training and testing on dialogue scenes derived from the dataset consisting of *TV Drama*, *NYPD Blue*, and the movie *Days of Thunder*. However, apart from showing the result for only one dialogue scene from *NYPD Blue*, there are no other results reported to validate their shot annotation scheme. In the semantic extraction stage, Ferman and Tekalp (1999) propose the use of a BN for two specific applications. In the first, a Bayesian Network is used to map a semantic description for each shot after class labels have been assigned by the HMM, and thus provide more descriptive semantic labels than the HMM states. Unfortunately, no further details are given in Ferman and Tekalp (1999) as to what the structure of the BN is, nor which semantic concepts are used. In the second application, they use a very simple BN (Figure-(2-8)) to extract the concept of ‘focus of attention’ based on the properties (size, position, motion) of the object tracked within the shot.

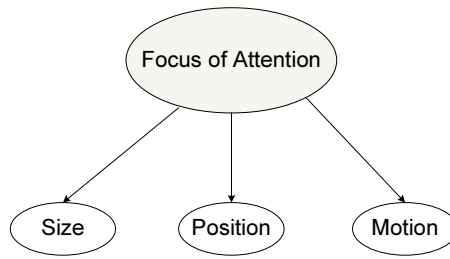


Figure 2-8: The simple Bayesian Network used in (Ferman and Tekalp, 1999) to track the concept of ‘focus of attention’ based on low-level properties of the tracked object.

Three levels of ‘focus of attention’, namely ‘*is*’, ‘*maybe*’, or ‘*is not*’ are used. The experimental results report that 92% of the time the BN assigns the same label as the human operator. Although the Bayesian Network used in their work is very simple, it nevertheless shows the initial feasibility of using such a probabilistic model to ‘map’ semantic concepts.

2.3.2.3 Bayesian framework for video retrieval

Approaches using BNs for the problem of retrieval typically treat features and semantic concepts S_i as random variables, whose relationships are captured in a Bayesian Network. The BN represents a joint probability distribution over all variables and trained using annotated data. A query is then translated into an inference problem in the BN. Using this approach are the works of Vasconcelos and Lippman (1998a, 2000a); Naphade *et al.* (1998); Naphade and Huang (2000b). As an example to show the mechanism behind these schemes, we discuss the work of (Vasconcelos and Lippman, 1998a). In this work, a generic Bayesian framework is proposed in which a query Q is answered by the closest match source S^* given by:

$$\begin{aligned}
 S^* &= \operatorname{argmax}_i \Pr(S_i = 1 \mid Q) \stackrel{(a)}{=} \operatorname{argmax}_i \frac{\Pr(Q \mid S_i = 1) \Pr(S_i = 1)}{\Pr(Q)} \\
 &\stackrel{(b)}{=} \operatorname{argmax}_i \{ \log \Pr(Q \mid S_i = 1) + \log \Pr(S_i = 1) \}
 \end{aligned} \tag{2.5}$$

where items in the content database are assumed to be observations drawn from a set of M content sources, and the boolean-valued variable S_i ($i = 1, \dots, M$) is defined as 1 if Q was drawn from the i th source¹¹. In this equation $\Pr(S_i = 1)$ is the *prior* for source S_i , and $\Pr(Q \mid S_i = 1)$ is formulated as an inference problem in a Bayesian Network as follows. In a generative process, Vasconcelos and Lippman (1998a) assume that each observation \mathbf{X} from a given source S_i is composed of K features $\mathbf{X} = \{X^1, \dots, X^K\}$,

¹¹and in step (a), simple Baye’s theorem is applied and in step (b), the constant $\Pr(Q)$ is ignored.

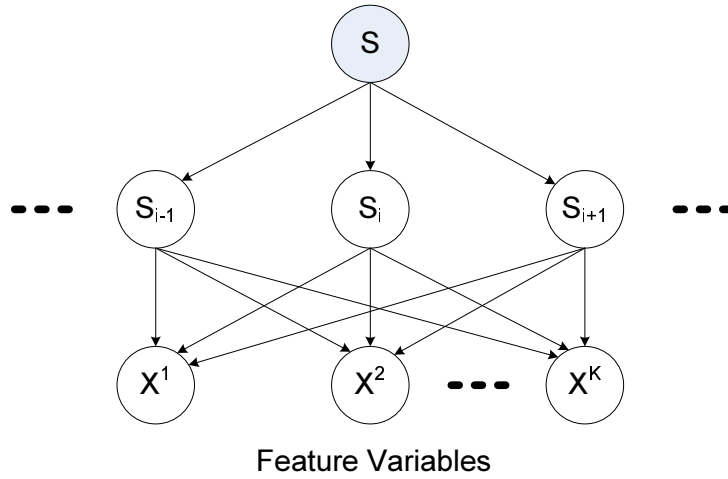


Figure 2-9: The Bayesian Network used in Vasconcelos and Lippman (1998) to represent the relationship between the observations (feature variables X^k) and the sources (source content variables S_i) in the database.

which are conditionally independent given S_i , that is:

$$\Pr(X^1, \dots, X^K | S_i) = \prod_{k=1}^K \Pr(X^k | S_i)$$

The Bayesian Network representing their casual relationship is depicted in Figure-(2-9), in which the authors introduce the extra source state multinomial variable S to keep track of which source is currently generating the observation, for example¹²: $\Pr(S_i = 1 | S) \triangleq \delta(S, i)$. Assuming that the parameters for the model have been properly trained, a query Q provided by the user is treated as consisting of a set of observed variables O , and a set of hidden (not instantiated) variables H , where O and H are disjoint and $O \cup H = X$. The likelihood of this query Q is then translated into a simple inference problem in the Bayesian Network by marginalising out un-instantiated variables:

$$\Pr(Q | S_i) = \sum_H \Pr(O, H | S_i) = \Pr(O | S_i) \quad (2.6)$$

Next, Vasconcelos and Lippman (1998a) consider a specific deployment of this framework when the set of feature variables X consists of both textual attributes T and visual attributes V , where $X = T \cup V$. The prior $\Pr(S_i)$ is treated as uniform; the conditional probabilities $\Pr(T^k | S_i)$ are modeled as a Bernoulli distribution; and $\Pr(V^k | S_i)$ are modeled as a mixture of Gaussians. Given this parameterisation for the model, the authors present a maximum likelihood estimation for the parameter. In principle the estimation follows the standard method used in the Bayesian Network literature. That is, it includes an explicit

¹²Note that in the original paper, the authors use slightly different notation. They represent S as an M -vector, whose elements are zero everywhere except 1 at the i th position to indicate $S_i = 1$.

expression of the log-likelihood, which will decouple the parameterisation for each local conditional probability $\Pr(X \mid \text{parent of } X)$, and each will then be maximised separately, eg: using the EM algorithm for the case of the mixture of Gaussians parameterisation, and taking the first derivative for the case of the Bernoulli parameterisation¹³. Even though no empirical results are given, the probabilistic framework developed by Vasconcelos and Lippman (1998a) is novel and scalable to large databases. For example, Equation-(2.6) reveals that the complexity of retrieval only depends on the number of features specified by the user K and not on the number of content sources M known to the systems, and thus M can be made arbitrarily large. Furthermore, prior knowledge such as the belief about a certain source in the database can be easily incorporated. A similar approach, called *Bayesian multiject* and *multinet*, is used in a series of work by (Naphade *et al.*, 1998) and (Naphade and Huang, 2000b,a).

2.3.3 Hidden Markov Model Based Approaches to Video Analysis

The Hidden Markov Model (Rabiner, 1989), and its variants, is perhaps one of the most widely and successfully used stochastic models in a diverse set of domains ranging from behaviour recognition (Yamato *et al.*, 1992), hand-writing recognition (Park and Lee, 1996) to genome structure discovery (Churchill, 1992), and most popularly in speech processing and recognition (Rabiner, 1989). Hidden Markov Models are also popular in video analysis, mainly in video surveillance and video content analysis. Video surveillance (eg: tracking, activity/behaviour recognitions, etc), however, is itself a complete self-contained theme and not directly relevant to the context of video content analysis presented in this dissertation, and thus will not be reviewed here.

We organise the rest of this section as follows. In the next subsection, we introduce the Hidden Markov Model as a special case of the Dynamic Bayesian Network, as well as the inference and learning problems in this model. Many excellent tutorials abound such as that of (Rabiner, 1989; Heckerman *et al.*, 1997); however, since our theoretical development for the Hierarchical HMM in Chapter 5 will use the HMM as the starting point, and address similar problems in the HMM (eg: numerical underflow), we shall therefore revise this model with sufficient detail in Subsection 2.3.3.1. This is followed by our review on the use of the HMM for video classification in Subsection 2.3.3.2, and video segmentation in Subsection 2.3.3.3. Next, we summarise the literature that attempts to use of the HMM in a hierarchical manner for video analysis in Subsection 2.3.3.4. Finally, we briefly review the Hierarchical Hidden Markov Model and their applications

¹³Alternatively, we note that the second layer of the BN in Figure-(2-9) can be completely removed due to the deterministic relationship between variables S and S_i . With this removal, this model will essentially reduce to the well-known *Naiïve Bayes* classifier.

in Subsection 2.3.3.5.

2.3.3.1 Hidden Markov Models

The Hidden Markov Model (HMM) (Rabiner, 1989) is a simple generative model which can be viewed as a special case of the Dynamic Bayesian Network as shown in Figure-(2-10) (when unrolled T times). Each time slice has a simple BN structure consisting of a single state variable x_t which generates the observation y_t . In general, the state variables $x_{1:T}$ are hidden and the observation $y_{1:T}$ are observed, where we write $x_{1:T}$ to represent the sequence of variables $\{x_1, \dots, x_T\}$.

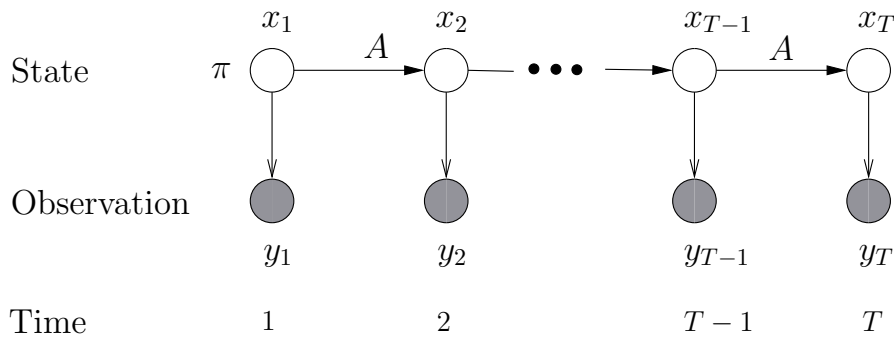


Figure 2-10: The Dynamic Bayesian Network representation of a Hidden Markov Model.

In the discrete case, together with the implicit information about the number of states N and number of observation alphabets M , a HMM is parameterised by $\lambda = \{\pi, A, B\}$ where $\pi_i \triangleq \Pr(x_1 = i)$ is the initial probability, $A_{ij} \triangleq \Pr(x_t = j \mid x_{t-1} = i)$ is the transition probability and $B_{v|i} \triangleq \Pr(y_t = v \mid x_t = i)$ is the emission probability. In the continuous observation case, the emission probability is usually modeled by a mixture of Gaussians. Three basic problems associated with the HMM outlined in (Rabiner, 1989) are:

- (1) computing the likelihood: $\Pr(y_{1:T} \mid \lambda)$
- (2) determining the best sequence state: $x_{1:T}^* = \operatorname{argmax}_{x_{1:T}} \Pr(x_{1:T} \mid y_{1:T}, \lambda)$, and
- (3) estimating the parameter λ^* that maximises the likelihood (or equivalently the log likelihood): $\lambda^* = \operatorname{argmax}_{\lambda} \Pr(y_{1:T} \mid \lambda)$.

For details on these problems and related discussions, readers are referred to (Rabiner,

1989). We briefly discuss here the problem of inference and learning in this model, which are essentially equivalent to three problems outlined above.

Inference in the HMM refers to the computation of the forward¹⁴ variable $\alpha_t(i) \triangleq \Pr(x_t = i, y_{1:t})$, and the backward variable $\beta_t(i) \triangleq \Pr(y_{t+1:T} | x_t = i)$, which is then used to compute two smoothing distributions needed in the learning: $\gamma_t(i) \triangleq \Pr(x_t = i | y_{1:T})$, and $\xi_t(i, j) \triangleq \Pr(x_{t-1} = i, x_t = j | y_{1:T})$. The forward/backward variables can be efficiently computed recursively via dynamic programming based on the conditional dependency from Markov property. The recursion for the forward variable, for example, can be calculated as:

$$\begin{aligned} \alpha_{t+1}(j) &\triangleq \Pr(x_{t+1} = j, y_{1:t+1}) = \sum_i \Pr(x_t = i, x_{t+1} = j, y_{1:t+1}) \\ &= \sum_i \Pr(y_{t+1} | x_{t+1} = j, \overbrace{y_{1:t}}) \Pr(x_{t+1} = j | x_t = i, \overbrace{y_{1:t}}) \Pr(x_t = i, y_{1:t}) \\ &= B_{y_{t+1}|j} \sum_i A_{ij} \alpha_t(i) \end{aligned}$$

Upon the computation of α_T , the likelihood is readily available: $\Pr(y_{1:T}) = \sum_i \Pr(x_T = i, y_{1:T}) = \sum_i \alpha_T(i)$. The backward variable β_t can also similarly be calculated recursively. For learning, the well-known EM algorithm (also known as the Baum-Welch algorithm) is used. During each iteration, the expected sufficient statistics (ESS) for parameter $\langle \lambda \rangle$ are computed in the E-step. In the M-step, the newly estimated parameter, resulting from the process of maximising the log-likelihood, is set to the normalised expected sufficient statistics. For example, assume we know how to compute $\xi_t(i, j)$ from $\alpha_t(i)$ and $\beta_{t+1}(j)$, the ESS for the parameter A_{ij} and its newly ML-estimated solution are:

$$\langle A_{ij} \rangle = \sum_{t=1}^{T-1} \xi_t(i, j) \quad \hat{A}_{ij}^{ML} = \frac{\langle A_{ij} \rangle}{\sum_i \langle A_{ij} \rangle}$$

To avoid repeating lengthy derivations, we refer readers to the classic reference of (Rabiner, 1989) for the computational details of β_t, γ_t, ξ_t and the Baum-Welch algorithm for the learning problem. Alternatively, an elegant way to perform inference and learning in the HMM is to consider the DBN structure of the HMM as a Bayesian network, which can then be used with the junction tree algorithm mentioned in Subsection 2.3.2. For further references on this work, we recommend the work of (Smyth *et al.*, 1997; Jordan, 2004).

Before concluding the background on the HMM, we wish to mention the problem of numerical underflow, an imperative issue in order for the HMM to have real-world applications. This issue also needs to be addressed for the Hierarchical HMM, which we shall address in Chapter 5. Underflow problems occur when T becomes large. Since the

¹⁴A version of un-normalised filtering distribution.

forward variable $\alpha_t \triangleq \Pr(x_t, y_{1:t})$ is the joint of probability of a large number of variables when $t \rightarrow T$, it will eventually become too small to be represented in the computer. To avoid this problem, a scaled version of α_t is defined and computed instead:

$$\tilde{\alpha}_t(i) \triangleq \Pr(x_t = i \mid y_{1:t})$$

Apart from being a scaled version of α_t , this probability also has the meaning of the *filtering distribution*, that is the filtered probability of a state given the observation up to time t . This variable can be computed by introducing two extra variables:

$$\begin{aligned} \text{the partially scaled variable:} & \quad \ddot{\alpha}_t(i) \triangleq \Pr(x_t = i, y_t \mid y_{1:t-1}) \\ \text{and the scaling factor:} & \quad \phi_t \triangleq \Pr(y_t \mid y_{1:t-1}) \end{aligned}$$

Assuming that $\tilde{\alpha}_t, \ddot{\alpha}_t, \phi_t$ are computed at time t , the recursions to roll over these variables to the next time slice $t + 1$ are as follows:

$$\begin{aligned} \ddot{\alpha}_{t+1}(j) & \triangleq \Pr(x_{t+1} = j, y_{t+1} \mid y_{1:t}) = \sum_i \Pr(x_{t+1} = j, x_t = i, y_{t+1} \mid y_{1:t}) \\ & = \sum_i \Pr(y_{t+1} \mid x_{t+1} = j) \Pr(x_{t+1} = j \mid x_t = i) \Pr(x_t = i \mid y_{1:t}) \\ & = B_{y_{t+1}|j} \sum_i A_{ij} \tilde{\alpha}_t(i) \\ \phi_{t+1} & \triangleq \Pr(y_{t+1} \mid y_{1:t}) = \sum_j \Pr(x_{t+1} = j, y_{t+1} \mid y_{1:t}) = \sum_j \ddot{\alpha}_{t+1}(j) \\ \tilde{\alpha}_{t+1}(j) & \triangleq \Pr(x_{t+1} = j \mid y_{1:t+1}) = \frac{\Pr(x_{t+1} = j, y_{t+1} \mid y_{1:t})}{\Pr(y_{t+1} \mid y_{1:t})} = \frac{\ddot{\alpha}_{t+1}(j)}{\phi_{t+1}} \end{aligned}$$

In Chapter 5, we will use a similar strategy (ie: introduce the partially scaled variable and the scaling factor) to solve the numerical problem for the Hierarchical HMM. For the rest of the review on HMM-based approaches in video analysis, we identify and organise the literature into four main areas: *video content classification/annotation*, *video segmentation*, *hierarchical use of HMMS*, and *using HMMS for fusion of multi-modal data*.

2.3.3.2 HMM-based video classification

The most popular use of the Hidden Markov Model in video analysis is for the task of *supervised* content classification or annotation, which is essentially comprised of two stages: *training* and *recognition*. In the training stage, based on an implicit assumption that different content genres pose distinct *temporal regularities*, a HMM is trained to capture the regularities in temporal structure for each content class. Assuming that there

are M classes of content types C_i (for $i = 1, \dots, M$), each is trained with a Hidden Markov Model parameterised by $\lambda^{(i)}$, then, in the recognition stage, a new segment D is classified into the k th class label C_k that maximises the likelihood, that is:

$$k = \underset{i}{\operatorname{argmax}} \Pr(D \mid \lambda^{(i)})$$

There is large body of work using this approach to characterise TV programs. In (Iyengar and Lippman, 1998), Iyengar and Lippman apply the Hidden Markov Model to classify sports and news programs and movies trailers. Training on a set of 24 news and sports programs, the authors report an average classification accuracy of 90% for a testing case consisting of 13 news and 13 sports programs. Next, the authors investigate the problem of classifying ‘action’ and ‘character’ movie trailers. A set of 24 movie trailers were used, and 3 trailers for each type of movie were randomly selected to train a HMM for that category. Using motion energy and shot length as features, a classification result of 78% was reported. In Wei *et al.* (2000), using HMMs, the authors attempt to classify a broadcast segment into one of four TV programs, namely news, commercials, sitcoms, and soaps. Different from the study of (Iyengar and Lippman, 1998) in which only low-level features are used, Wei *et al.* (2000) utilise a set of mid-level features from face and text detection as input to the HMMs. Their observation is that different TV programs have certain temporal regularities in the ‘trajectories’ of face and caption texts, and thus can be captured efficiently by the Hidden Markov Model. Gibert *et al.* (2003) use HMMs to classify four types of sports, namely hockey, basketball, football and soccer. Using colour and motion as features, a HMM is trained for each class and then used in the classification stage based on the maximum likelihood criteria. An accuracy of 93% was reported when testing on 220 minutes of sports video.

Noting that building a HMM from raw features at frame-level (eg: in Iyengar and Lippman (1998); Wei *et al.* (2000)) is computationally expensive, Lu *et al.* (2003) perform an extra keyframe extraction step for each training segment and features are instead extracted from the set of keyframes to fit the HMM. Using only audio information, Liu *et al.* (1998) construct five ergodic HMMs to characterise five types of TV programs, including commercials, basketball games, football games, news reports and weather forecasts. This work is later extended in Huang *et al.* (1999a) by combining both audio and visual information. One of these methods uses a two-stage HMM. First, using audio information alone, the authors separate the video into three broad categories: commercials, games, and reports. Visual information is then used to train a number of HMMs to separate ‘games’ into ‘basketball game’ or ‘football game’; and similarly to separate ‘reports’ into either ‘news report’ or ‘weather report’.

Despite the variance in performance, approaches that use Hidden Markov Models to char-

acterise broadcast content generally perform well since TV programs are relatively unique in their aural and visual temporal characteristics. The main drawback in this approach, as with any other HMM-based framework, is the determination of the number of states, which is typically resolved with the aid of domain knowledge. Another disadvantage is the requirement of pre-segmented and labeled data during the training period, which is usually a very time-consuming process. Moving beyond the domain of TV programs, there are also a few studies attempting to use the Hidden Markov Model to characterise richer content type (Kang, 2003). In (Kang, 2003), for example, Kang uses Hidden Markov Models to model three types of emotional content including {fear, sadness, joy} from low-level features such as colour, motion, and shot length.

2.3.3.3 HMM based video segmentation

Existing methods using Hidden Markov Models for video segmentation can be broadly categorised into two main approaches, which we term as *window-based* and *Viterbi-based* approaches. In the former, assuming a finite set of content types exist, a HMM-based classification process is performed *prior* to the segmentation stage. A sliding window is then scanned through the entire video, in which at each step, the content within the window is input to a pool of HMMs to compute the likelihood it belongs to one of the content types. The likelihood values are then used to determine the segmentation points, most widely by using dynamic programming to determine an optimal likelihood path for the entire video. In the *Viterbi-based* approach, instead of training a series of HMMs for a pre-defined set of content classes, this approach uses a *single* Hidden Markov Model to model the entire video, usually with some specific domain knowledge such as typical syntax of the video. After training, a video is segmented based on a Viterbi decoding on the video.

Window-based HMM approach to video segmentation

Huang *et al.* (2000) use the HMM for the problem of scene segmentation and classification. Their approach consists of two phases. First, a series of HMMs are trained for different types of content, which are used to compute the likelihood of a shot belonging to a particular class. Assuming that the video \mathbf{V} consists of T shots $\{\mathbf{S}_t\}_{t=1}^T$, and there are M content types, each parameterised by a Hidden Markov Model $\lambda^{(m)}$, then at any arbitrary time t , M likelihood values for each shot $\{\Pr(\mathbf{S}_t | \lambda^{(m)})\}_{m=1}^M$ are evaluated, and thus the likelihood values for the entire video with respect to each specific content type are maintained. In a dynamic programming approach, an optimal path $P^*(\mathbf{V})$ consisting of a sequence of shot labels that maximise the accumulated likelihood is computed. The video \mathbf{V} is then segmented (and also annotated) into ‘blocks’ (of shots) at the points where a content change is observed in $P^*(\mathbf{V})$. An experiment conducted for five content types

{commercial, live basketball game, live football game, news, and weather forecast} , each trained with a five-state ergodic HMM are reported in (Huang *et al.*, 2000). Similar ideas are used in Xie *et al.* (2002b); Barnard *et al.* (2003); Kijak *et al.* (2003b,a); Xie *et al.* (2004). Xie *et al.* (2002b, 2004), for example, use two HMMs to model ‘play’ and ‘break’ scenes in soccer videos. Using dominant-colour ratio and motion intensity as the feature, the authors train two models separately using pre-segmented data. The transition matrix from ‘play’ to ‘break’ or vice versa are constructed by manually counting the transitions in the training data. The likelihood values computed over the sequence of segments for the ‘play’ and ‘break’ concepts are used as ‘observations’ for the optimal path determination as in Huang *et al.* (2000). The authors report a classification accuracy of over 83.5% and a comment that ‘most of boundaries are detected’.

Decoding-based HMM approach to video segmentation

In Iurgel *et al.* (2001), Hidden Markov Models are employed to combine audio and visual features for topic boundary detection in the domain of TV news. Boykin and Merlino (2000) extend the finite state machine in their previous work (Merlino *et al.*, 1997) to model news programs by a HMM consisting of four states {start-topic, commercial-break, end-topic, and others}. The HMM is trained and an unseen news video is segmented into topics by observing when the start-topic or end-topic is reached during the Viterbi decoding. Similar approaches are used in (Wolf, 1997; Kijak *et al.*, 2003b); and in some of the works we have reviewed earlier for shot detection (Boreczky and Wilcox, 1998; Sanchez *et al.*, 2002) (page 15). Another closely related approach is the use of the finite state machine to segment a video as we have reviewed in the Film Grammar based approach, including the works of (Merlino *et al.*, 1997; Chen and Ozsu, 2002; Zhai *et al.*, 2004) (page 26).

2.3.3.4 Using a hierarchy of Hidden Markov Models

The major drawback of using a regular Hidden Markov Model in video analysis is that the strong Markov assumption limits its expressiveness to model long-term dependencies in the video. The aforementioned Viterbi-based approach, for example, may work well for short video clips whose genre description is clear in its syntax (eg: news), however it would appear to completely fail to model a long video (eg: full length movie), in which long-term temporal correlations exist. Modeling the video in such situations is generally a very difficult problem. A rigorous method to tackle this problem is by using a *hierarchy* of Hidden Markov Models to provide semantic descriptions and segmentation at multiple levels. Existing methods generally use pre-segmented training data at multiple levels, and hierarchically train a pool of HMMs, in which HMMs at lower levels are used as input to

the HMMs at the upper levels¹⁵. In principle, some fundamental units are recognised by a sequence of HMMs, and then likelihood values (or labels) obtained from these HMMs are combined to form a ‘stack’ of HMMs to capture the interactions at a higher semantic level.

Some of the methods reviewed previously (Huang *et al.*, 2000; Xie *et al.*, 2002b) have also used HMMs at the lower level as input to the upper level for semantic mining. However, the upper level in these studies is not strictly modeled as a Hidden Markov Model. More rigorous use of HMMs at multiple levels include the works of Kijak *et al.* (2003b,a); Naphade and Huang (2002). Targeting the sports domain, Kijak *et al.* (2003b,a) propose the use of HMMs at two levels to provide a two-tiered classification of tennis videos. At the bottom level, the authors use HMMs to model four classes, namely ‘first missed serve’, ‘rally’ (to represent game phases), ‘replay’, and ‘break’ (to represent non-game phases). Each HMM is then trained separately, and subsequently connected to a high-level HMM which represents the syntax of the tennis video with three states of the game {sets, games, and points}. Parameters for the high-level HMM are, however, completely manually specified. A generic two-level hierarchy of HMMs is proposed in (Naphade and Huang, 2002) to detect recurrent events in videos. Their idea is to use an ergodic HMM at the top level, in which each state is another (non-ergodic) sub-HMM representing a type of the signal with certain stationary properties. Even though no details are given, the authors claim to use the EM algorithm to train the model¹⁶. This model is then applied in (Naphade and Huang, 2002) to detect recurrent events in movies and talk shows. In the former case, the top level HMM has six states, each is, in turn, another three-state non-ergodic HMM. The observations are modeled as a mixture of Gaussians. After training, the authors claim that interesting events can be meaningfully detected such as ‘explosion’, ‘male speech’, and so on. Another noteworthy approach is the use of the multiple HMMs for fusion of multi-modal data. Instead of using multiple HMMs to provide hierarchical descriptions of the data, these approaches use HMMs multiply as a framework to fuse data such as that of (Naphade *et al.*, 2001) and (Sanchez *et al.*, 2002).

2.3.3.5 Hierarchical Hidden Markov Models

While being able to overcome the limitation of the regular HMM in modeling long-term dependencies, approaches that use HMMs at multiple levels still suffer from two major

¹⁵We note that this method of combining HMMs into a hierarchy closely resembles the cascaded HMM (Brants, 1999), and the embedded HMM (Murphy and Nefian, 2001). All of these models are indeed a special case of the Hierarchical Hidden Markov Models we develop in Chapter 5 when the time indices (ending status) of all sub-HMMs are observed.

¹⁶In principle, this can be done by simply ‘flattening’ this two-level HMM into a regular HMM with the bigger product state space, and it then can be trained as with the regular HMM.

problems: (1) pre-segmented and annotated training data are needed at all levels, and (2) parameters at higher levels, in most existing work, have to be hand-crafted. In many cases, preparing training data at multiple levels is extremely tedious and may not be possible. In the second problem, each semantic level has to be modeled separately, and thus the underlying problem is that the framework does not capture the essential nature of the interactions across semantic layers, and these interactions are not part of the learning process.

One framework that integrates the semantics across layers is the Hierarchical Hidden Markov Model¹⁷ proposed recently in (Fine *et al.*, 1998). The hierarchical HMM extends the standard HMM in a hierarchic manner to allow each state to be recursively generalised as another sub-HMM, and thus enabling the ability to handle hierarchical modeling of complex dynamic processes, in particular “the ability to infer correlated observations over long periods in the observation sequence via the higher levels of the hierarchy” (Fine *et al.*, 1998). The original motivation in (Fine *et al.*, 1998) was to seek better modeling of different stochastic levels and length scales presented in language (eg: speech, handwriting, or text). To provide the relevant background, at the same time maintaining the readability, we will briefly describe this model here using the original explanation and notations in (Fine *et al.*, 1998) and shall revise this model in greater detail in our own presentation in Chapter 5.

Being a structured multi-level stochastic process, a HHMM allows each of its hidden states another sub-HHMM on its own. An observation is generated by a HHMM via a recursive activation of a state, then one of its substates, and so forth, in which the activation ends when it reaches the *production state* at the bottom of the hierarchy. By definition, only the production states¹⁸ emit observations, and those hidden states that do not emit observations directly are termed as *internal* (or *abstract*) states¹⁹. A transition from an internal state to another internal substate at the lower level is termed as a ‘vertical transition’. Once the vertical activation is complete, control returns to the calling state, which then makes a ‘horizontal transition’ within the same level. Let D be the depth of the hierarchy, then the entire set of parameters defined for a HHMM is denoted by:

$$\lambda = \underbrace{\{A^{q^d}, \pi^{q^d}\}}_{\lambda^{q^d}} \cup \{B^{q^D}\} \quad \text{for } d = 1, \dots, D - 1$$

We can think of this parameter set as follows: each internal state q at level d is associated with $\lambda^{q^d} = \{A^{q^d}, \pi^{q^d}\}$ which parameterises the vertical (π) and horizontal (A)

¹⁷We note here the distinction between hierarchical use of HMMs and the Hierarchical Hidden Markov Models. In the former, a series of HMMs are employed and combined in a hierarchic manner, whereas the HHMM is itself a complete stochastic model that allows a state of a normal HMM to be another sub-HMM, and thus much harder to be represented, inferred, and learned.

¹⁸We can think of production states as usual states in a regular HMM.

¹⁹The term ‘internal’ is used in (Fine *et al.*, 1998), and ‘abstract’ is used in (Murphy, 2001).

transition probabilities for its substates; and B^{q^D} is the emission probability defined at the production level. Three fundamental problems for the regular HMMs in (Rabiner, 1989) (discussed in Subsection 2.3.3.1) are similarly addressed in (Fine *et al.*, 1998). Inspired by the Inside/Outside algorithm for the problem of parameter estimation in SCFG, the authors in (Fine *et al.*, 1998) have developed a method for inference and learning in this model with complexity of $O(T^3N)$ where T is the observation length, and N is the total number of hidden states. This algorithm shall be further discussed in Chapter 5. In (Murphy, 2001), the author presents an alternative algorithm to perform inference for the HHMM by representing this model in a DBN and subsequently applying standard forward/backward inference in the usual DBN structure. The algorithm in (Murphy, 2001) is thus linear in T , but still exponential in depth D .

For applications, the HHMM has found initial applications in hand-writing recognition (Fine *et al.*, 1998), robot navigation (Theocharous and Mahadevan, 2002), behaviour recognition (Luhr *et al.*, 2003), and information retrieval (Skounakis *et al.*, 2003). Relevant to the area of video analysis, the HHMM offers a unified probabilistic framework to model temporal dependencies in video at multiple scales, and at the same time *prior* structural information, such as the number of states at each level, can easily be incorporated. Its application is, however, very limited to only the work of (Xie *et al.*, 2002a), which is then extended in (Xie and Chang, 2003). In this work, the authors use the HHMM to detect the events of ‘play’ and ‘break’ in soccer videos. For inference and learning, a HHMM is ‘flattened’ into a regular HMM with a very large state space, which can then be used in conjunction with the standard forward/backward inference in a normal HMM. Converting the HHMM to a flat HMM suffers from many disadvantages (Murphy, 2001): (a) it cannot provide multi-scale interpretation, (b) it loses modularity since the parameters for the flat HMM get constructed in a complex manner, and (c) it may introduce more parameters, and most importantly it does not have the ability to reuse parameters, in other words parameters for shared sub-models are not ‘tied’ during the learning, but have to be replicated and thus we lose the inherent strength of hierarchical modeling. The work of (Xie *et al.*, 2002a) is however novel and interesting in that the structure of the model is learned in a completely unsupervised manner in a combination of a Markov Chain Monte Carlo (MCMC) model selection and a procedure for automatic feature selection. The search over the model space is done with reverse-jump MCMC in a split and merge procedure combined with a Bayesian Information Criteria as the model prior. The feature selection is done in a joint of filter and wrapper methods, in which clusters of feature groups are first discovered based on the information gain criteria, and then an approximate Markov blanket for each group is constructed. Finally, the resulting models and feature sets are ranked based on a *a posteriori* fitness test. In an experiment designed to detect the ‘play’ and ‘break’ events in soccer video, Xie *et al.* (2002a) compared four methods: (1) Supervised HMMS, in which each category is trained with a separate HMM (similar to

techniques reviewed in Subsection 2.3.3.2), (2) Supervised HHMMs, in which bottom level HMMs are learned separately and parameters for the upper levels are manually specified (similar to approach reviewed in Subsection 2.3.3.4), (3) Unsupervised HHMMs without model adaptation, and (4) Unsupervised HHMMs with model adaptation. In (3) and (4), two-level HHMMs are used. Their results report a very close match between unsupervised and supervised methods in which the completely unsupervised method with model adaptation performs best. These figures are 75.5%, 75.0%, 75.0% and 75.7% respectively for the four above methods. While presenting a novel contribution to the feature selection and model selection procedure, the application of the HHMMs in this work is still limited both for learning and for the exploitation of the hierarchical structure.

2.4 Closing Remarks

This chapter has provided related background information for this thesis. We have started with an overview of the field of video content analysis in which the semantic gap has been identified as the final ‘frontier’, whether it is for video annotation, segmentation, summarisation or retrieval. Towards this end, we have made the case for the use of Film Grammar based and probabilistic methods. We then review the body of literature that based their work on Film Grammar and probabilistic methods respectively. In the Film Grammar based approaches, we have emphasised the role of the Computational Media Aesthetics framework as a formal and systematic way of analysing video content. In the probabilistic methods, we directed our review to the two most popular and widely used models, namely the Bayesian Network and the Hidden Markov Model. We identify the need for the use of the Hierarchical Hidden Markov Models together with some relevant background and its applications.

In the next chapter, we present our first contribution to the analysis of educational videos and provide the first exploitation of the specific Film Grammar for this film genre to construct a meaningful hierarchy of structural units.

Chapter 3

Identification of Hierarchical Narrative Structural Units

As widely acknowledged in the video indexing research community, one pressing problem in content management is to develop methods to automatically structuralise multimedia data in order to bridge the semantic gap through high-level semantics-based video partitioning, event extraction, and content tagging. This problem is crucial for automatic media searching and browsing processes to become more effective. Reviews in Chapter 2 have shown that much research in this field has targeted broadcast videos and lately motion pictures, however little attention has been devoted to instructional media, in particular the domain of education videos.

This chapter addresses the first step towards automatic structuralisation of educational and training videos, a domain gaining notice with recent interest in e-learning media technologies (eg: Dorai *et al.* (2001)). We present commonly observed rules and conventions in instructional video productions to manipulate the presentation of content in media to match learners' needs. Leveraging upon production grammar to shape our understanding of the common structural elements employed in instructional media, we propose a hierarchy of narrative structures used. Characteristics of each structural element which manifest as a sequence of shots are examined and a set of audiovisual features for capturing the differences between them is proposed. The C4.5 algorithm is used in our hierarchical classification system. Experimental results are presented to demonstrate the richness and effectiveness of the feature set proposed and the resulting classification of narrative structures.

The remainder of this chapter is organised as follows. Section 3.1 describes techniques that contribute to the “grammar” of educational videos. Next, the hierarchy of narrative structural elements is developed in Section 3.2. Extraction of features is addressed in

Section 3.3, followed by experimental results in Section 3.4.2. Finally, the chapter summary is provided in Section 3.5

3.1 Understanding the Educational Film Genre

Chapter 2 has outlined the principle of Film Grammar and its relation to computational media aesthetics. In this section we shall discuss specific aspects of “grammar” that are used by creators of media to make educational videos. This is also known as the *expository* class of film. Our aim is to distil specific elements peculiar to this film genre and gain insight into the extraction of useful structural units. For example, Braddeley (Braddeley, 1975, p.171) remarks:

“The aim of documentary or story-film editor is the creation of mood, the dramatisation of events. To the editor of educational films, these considerations are largely irrelevant. The purpose of his films is to teach and his aims must be clarity, logical exposition and a correct assessment of the audience’s receptivity.”

So how then can an educational film be formally distilled? We first subdivide this genre into two types: *instructional* films and *teaching* films. Instructional films encompass a class of the so-called “how-to” films (Herman, 1965; Braddeley, 1975). In the simplest case, they are recorded to guide the audience through a defined set of steps where the footage can be filmed with a single narrator. At a more complex level, the instructional film starts to teach a mixture of specific skills. This could pertain to techniques in a situation or a workplace, for example, the use of heavy equipment for farming, or a set of rules for handling certain situations, for example, teaching an employer to handle customers at a reception desk. These kinds of films need to capture the complexity for the set of situations or rules, and present each in a sequential fashion. Typically, there will be an introduction to the whole video, followed by an elaboration of each situation or rule. In such films, the order of the material is more or less fixed.

Sharing certain similarities with instructional films, teaching films, however, are usually more complex as they cover less concrete material such as the explanation of a theory in science or a phenomenon in physics. They are mainly used for classroom purposes¹. The topics they cover are more complex, and thus logical inference has to be made in order

¹In this thesis, we are dealing with professionally created videos, *excluding* those lecture videos recorded by hand-held camera (which are used in a number of works in the literature, eg: Mahmood and Srinivasan (2000))

to understand the material. These films are more complicated to craft in terms of both order and presentation, and need to use an array of media techniques.

How then is the material presented in an educational film? The literature (Herman, 1965; Davis, 1969; Braddeley, 1975) offers a range of techniques. In this thesis, we concentrate on a specific and most commonly used structure, namely the *expository rundown*.

In planning a video with expository rundown, the structure must be designed to determine both the content and the order of the material. The structure can be logical, or functional depending on what is being shown. This structure often forms the core of the video, and the resulting framework is reinforced in many ways: the narrator telling the audience about the structure with text summarising the video thus far, or outlining in text what is going to follow. Once the structure is determined, there is a general introduction to the material by the narrator. Occasionally to draw attention, some expressive material is used. The purpose of the film is identified, and the structure is clearly defined. This then leads to the main material being presented. The main ideas should be reinforced, and shown from many points of view. Each subtopic should terminate with a summary of what has been shown. At the end, the conclusion summarises the main points again, either directly through the narrator and/or with texts. To shape the content within the framework of expository rundown, a number of sub-structures are used by the professionals. We identify four important techniques: description, the use of point of view, substantiation, and demonstration.

- *Description*. This is to present to the users a description of the subject under discussion. Typically, this can be done with both video and descriptive commentary by the narrator that explains the visual content of the video.
- *The point of view*. In educational films, there is a strong point of view, and it is generally the viewpoint of the narrator. The narrator weaves the audience in and out of the planned structure, appearing either through the voice-over or in person at the end of the topics or subtopics.
- *Substantiation*. This involves showing examples or comparisons that clarify the discussion. This can be done in several ways: elaborating footage, interviews, other points of view footage, re-enactments and so on.
- *Demonstration*. This is an integral part of “how-to” films and it involves the narrator demonstrating the steps in the video, in a step-by-step manner. Since this technique by itself can be quite boring, directors intersperse this with other techniques to sustain attention. This can be either through expressive methods such as the use of music, or other pieces of expressive linkage.

Given this knowledge, in the next section we formally identify a set of structural units for the educational genre.

3.2 Uncovering the Hidden Structural Units

Jain and Hampapur (Jain and Hampapur, 1994) stress that the structure of a medium is greatly influenced by the nature of information in it and the purpose for which it is used. An educational video is not an exception, and indeed far from being a haphazard juxtaposition of content, it is a highly structured medium aimed at having maximum impact on the viewer.

Compared with other video genres - eg: feature film - educational videos exhibit many fundamental differences. Every video is built for a specific *purpose* and this is where educational videos differ the most. The purpose of a feature film is to *entertain* the audience, while the objective of an educational film is to *teach* and *train*. A well-crafted clip that moves viewers to action or instils a long-lasting message in their memory requires directing skills. Not only does the organisation of the material matter, but numerous aesthetic choices have to be made. Most importantly, not only is *what* to be shown important but also *how* it is to be presented. In the quest for the answer to the *how*, we find the building blocks that will serve as our structural units. Guided by these production principles, we propose a hierarchy of narrative structures which are considered crucial constituent elements for effective presentation. Figure-(3-1) depicts the proposed hierarchy. Four main categories are identified: *on-screen* narration, *voice-over* narration, *linkage* narration and *supportive* narration. Each category is further subdivided into finer classes of narration. We shall now describe each of these elements in detail.

3.2.1 On-screen Narration

On-screen narration (ON) refers to segments in the video containing appearances of the narrator. The purpose of these sections is typically to speak to viewers with the voice of authority and such sections are used to introduce a new topic (subtopic), to clarify a concept or to lead the viewers through a procedure with examples. We further distinguish on-screen narration into two types:

- *Direct Narration* (DN). This involves *eye-to-eye* contact where the narrator speaks to the viewers directly. The face of the narrator is shown from the beginning of the shot and there is little movement through the shot.

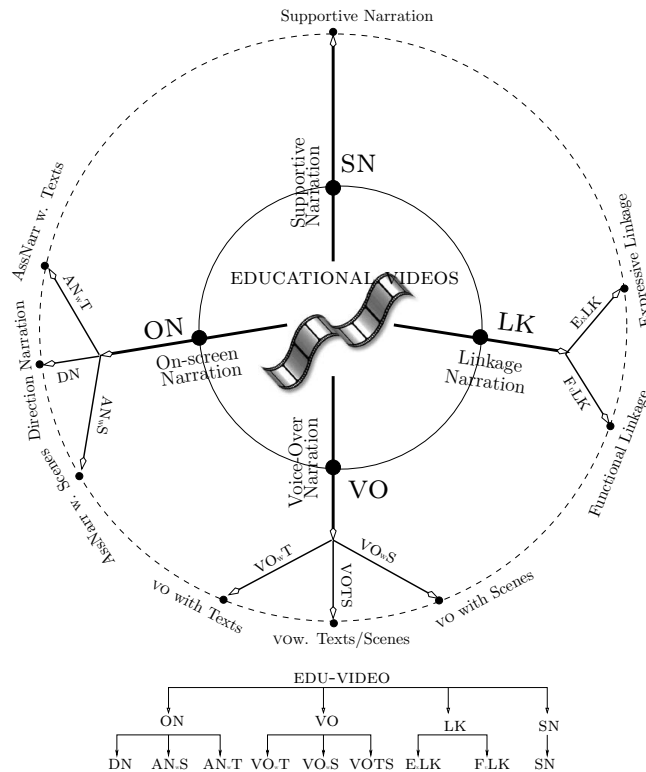


Figure 3-1: The hierarchy of proposed narrative structures for educational videos.

- *Assisted Narration* (AN). By “assisted” we mean that although the narrator is present in the segment, the attention of the viewers is *not necessarily* focused on the narrator(s). Here, the purpose is not only to talk to the viewers but also to *emphasize* a message by means of text captions and/or to convey *experience* via background scenes. We further refine this category into two sub-structures:
 - *Assisted Narration with Text* (ANwT). Here, the narrator communicates with the viewers using superimposed text captions and these may be accompanied by illustrative scenes. For example, in a safety film about eye protection, the narrator holds a model of the eyes to show the area in the eye with text displayed on the screen.
 - *Assisted Narration with Scenes* (ANwS). Typical film segments found for this category are shots of the narrator walking around in the scene. A training film, for instance, shows the scene in which the narrator speaks about clearance of the footpath and at same time walks into a factory to show viewers the visual ‘experience’ of a satisfactory footpath.

Some visual examples to illustrate these sub-structures are shown in Figure-(3-2). The audio track typically contains speech and sometimes expressive silence.

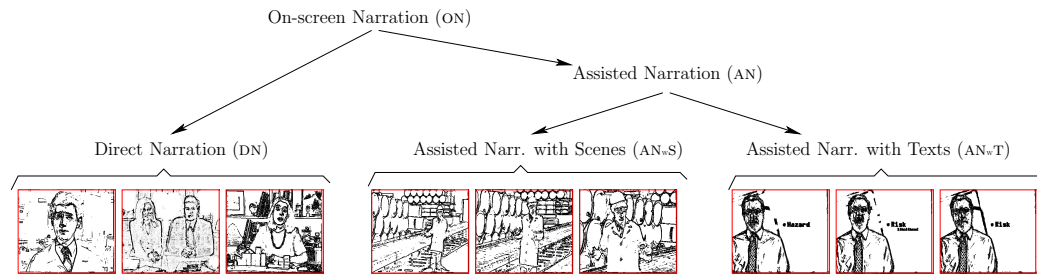


Figure 3-2: On-screen narration sub-hierarchy and examples.

3.2.2 Voice-over Narration

Voice-Over (VO) sections are identified as segments where the audio track is dominated by the voice of the narrator, *but without his or her* appearance. The purpose of these segments is to communicate with the viewers using the narrator's voice and adding pictorial illustrations in the visual channel. Voice-over sections are further categorised as:

- *Voice-Over with Texts only* (VO_T). Here, along with the voice of the narrator, there is *only* superimposed text on the screen. The text is usually presented on a simple background.
- *Voice-Over with Scenes only* (VO_S). This is the most commonly encountered structure in educational films, and it contains voice-over in the audio channel and scenes in the background but no text captions are displayed.
- *Voice-Over with both Text and Scenes* (VO_{TS}). This structure captures voice-over sections where both text and scenes are found in the visual channel. While in the former category the film director chooses to communicate with viewers using superimposed texts or scenes only, a narrative structure with both text and scenes not only gives direct messages in text but also communicates shared experiences.

Figure-(3-3) shows some visual examples for the voice-over category. Speech is typically found dominant in the audio track.

3.2.3 Linkage Narration

The linkage narration (LK) structure is captured as sections of film, whose purpose is to maintain the continuity of a story line, but there is *neither on-screen nor voice-over narration* involved. We subdivide this category into the following two groups:

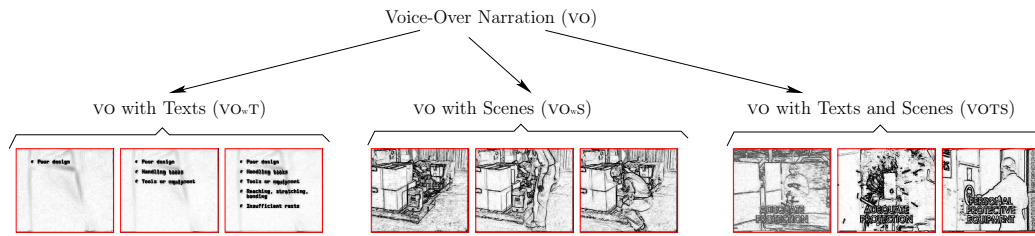


Figure 3-3: Examples of voice-over narrative structures.

- *Functional Linkage Narration* (F_{VLK}). This structure contains transition shots encountered in switching from one subject to the next. Usually, large superimposed text captions are used and the narration stops completely.
- *Expressive Linkage Narration* (E_{XLK}). This structure is used to create ‘mood’ for the subject being presented. For example, in a safety film presenting the fire safety issue, there is a segment in which the narration is completely stopped and a sequence of burning houses is depicted. These scenes obviously do not give any instructions or convey any concepts in a direct way, but create a ‘mood’ that helps the film to be more interesting and facilitates the presentation of fire safety issues.

3.2.4 Supportive Narration

This category encompasses all interviews, sections of “voices of experience”, and sections that cannot be classified into any of the categories described above and are abbreviated as SN.

Figure-(3-4) shows some visual examples of LK and SS structures. The type of audio observed is various (music, silence, background noise, etc).

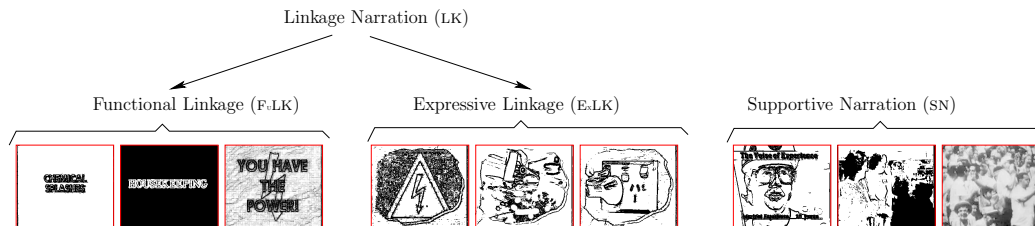


Figure 3-4: Examples of linkage and supportive narrative structures.

3.3 Feature Extraction

Designing ‘good’ features is crucial to the recognition task. In this section, we describe the feature extraction process that underpins the recognition of narrative elements.

3.3.1 Visual Content Analysis

The first stage in the visual analysis phase is grouping frames into shots. In this work, we use a commercial software (Mediaware-Company, 1999) to detect *cut* and *dissolve* transitions. After shot indices are computed, the shots are analysed for the presence of face and text captions, in which the face detection algorithm proposed in (Rowley *et al.*, 1998), and the text detection algorithm in (Shim *et al.*, 1998) are used².

3.3.1.1 Face detection and associated features

The space and time chosen for a narrator to appear in an educational video is obviously not arbitrary but rather deliberate to deliver the director’s messages, in particular to capture the viewers’ attention. The first feature set is constructed based on the detection and tracking of the face, which is assumed to be that of the narrator within a shot. A narrator might appear in a variety of angles and points of view. However, we are only interested in detecting frontal faces that look directly at the camera and dominate the frame. This is further justified by the knowledge of Film Grammar, for instance in (Davis, 1969, p.54):

“[Rule] 24d. Never let a performer look straight into the lens of the camera unless it is necessary to give the impression that he is speaking directly to the viewer personally.”

To perform the face detection within a shot, one can simply exhaustively search every single frame. This is a computationally expensive process. To achieve faster computation, we select two representative sequences of frames from each shot for analysis. A straightforward and simple selection is a sub-sampled version of the original frame sequence. The second sequence includes selected keyframes based on a well-known mechanism – the difference in the colour histogram function such as in (Truong *et al.*, 2002b). Features are then extracted from each sequence and averaged to form one set of features. For example, if $f_A^{(1)}, f_A^{(2)}$ are the values for feature A extracted from the two sequences, then $f_A = (f_A^{(1)} + f_A^{(2)})/2$

²These algorithms were also previously discussed in Section 2.1.1.1 of the background chapter.

is used as the value for feature A . Given a representative sequence $\Gamma(\mathbf{S}) = \{\tau_i\}_{i=1}^M$ for a shot \mathbf{S} , we define the following features:

(1) The *Face-Content-Ratio* (FCR) feature. This feature is designed to capture a shot in which we observe the dominant appearance of the narrator. It is computed as the normalised number of frames where faces are detected:

$$\text{FCR}(\mathbf{S}) \triangleq \sum_{i=1}^M \mathcal{F}(\tau_i) / M \quad (3.1)$$

where $\mathcal{F}(\tau_i)$ is the face detection indicator function, which returns 1 if there is face in frame τ_i and return 0 otherwise. An on-screen narration section, for example, would return a high value for the FCR feature compared to other categories due to the dominance of the narrator. Figure-(3-5) plots this feature for different structural elements obtained from a mixture of 15 instructional and teaching films, where features are all sorted before plotting. A long horizontal line centered around zero during the first 220 shots for the E_{xLK} category, for example, indicates that most of the FCR values computed for this category are very small with some exceptions (outliers). We further observe that in most of the cases, except the on-screen category (ie: AN_{wS} and DN), all others have their FCR values of around zero.

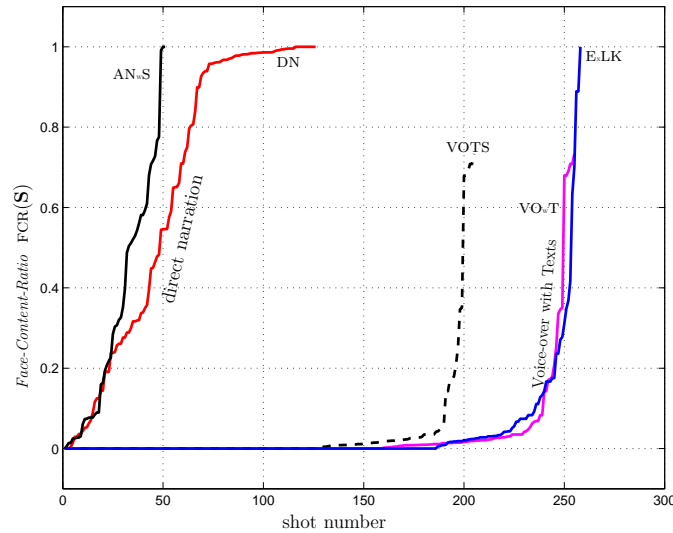


Figure 3-5: *Face-Content-Ratio* features (FCR) plotted for AN_{wS} , DN , VOTS , VO_{wT} , and E_{xLK} (values are sorted).

(2) The *Face-Bounding-Box* features. Within the on-screen group, although there is a dominance of a face in both the direct- and assisted-narration categories, we observe more variations in the detected face across successive frames in the AN category. We capture this fact by measuring the changes in the coordinates of the detected face. Let (X_ϕ, Y_ϕ) be

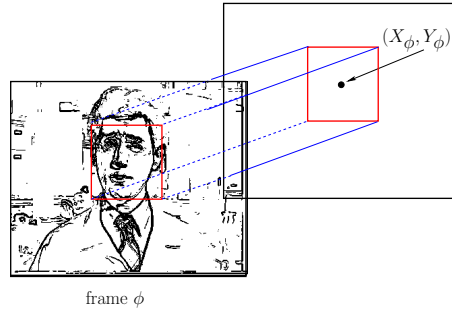


Figure 3-6: Coordinates (X_ϕ, Y_ϕ) of bounding box center are used as features to encode the movement of the narrator.

the coordinates of the center of the face bounding box in frame ϕ (see Figure-(3-6)). The means $\mu_X(\mathbf{S}), \mu_Y(\mathbf{S})$ and variances $\sigma_X^2(\mathbf{S}), \sigma_Y^2(\mathbf{S})$ computed from two sequences $\{X_{\tau_i}\}_{i=1}^M$, and $\{Y_{\tau_i}\}_{i=1}^M$ are used as the face bounding box features for shot \mathbf{S} .

(3) The *Face-Area-Ratio* feature. In direct-narration shots, it is common that the narrator is filmed as a close-up, and therefore the area of the detected face is generally larger than that of assisted-narration shots. We encode this information by a normalised measure of the area of detected face. Let K_Γ be the number of frames in $\Gamma(\mathbf{S})$ where a face is detected and $\{v_1, v_2, \dots, v_{K_\Gamma}\}$ be the areas of the bounding box in these frames respectively, then this feature is given as³:

$$\text{FAR}(\mathbf{S}) \triangleq \frac{\sum_{k=1}^{K_\Gamma(\mathbf{S})} v_k}{K_\Gamma \times Z} \quad (3.2)$$

where Z is the area of a whole frame (352×288). Figure-(3-7) plots this feature for DN vs. AN, where a high ratio is observed for the DN category when compared with AN.

3.3.1.2 Text detection and associated features

Another valuable channel of information in educational videos is the text embedded in the frame. There are mainly two types of texts: *scene* texts and *superimposed* texts (see Figure-(3-8)). The former refers to texts that appear and are part of the scene. These texts are recorded within the scenes such as street names, warning signs, or texts on a notice board outside a classroom. Such texts are difficult to detect reliably due to the unconstrained nature of their appearance (Shim *et al.*, 1998) and are less important to our analysis. Superimposed texts, on the other hand, are easier to detect and more meaningful to our analysis. They give signals as to when a film starts, demarcating sections, emphasising key points, and drawing attention to specific details. They usually appear in bold form

³Note that we normalise according to K (number of frames in which faces are detected), not to M (number of frames in the sequence).

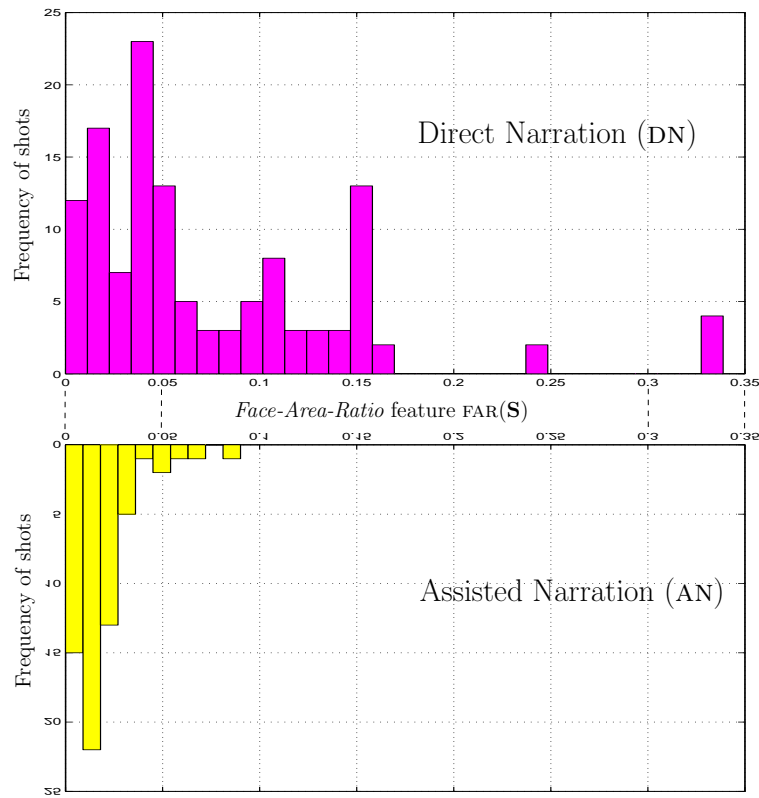


Figure 3-7: Example of FAR features plotted for Direction Narration vs. Assisted Narration.



Figure 3-8: Examples of scene texts (left) and superimposed texts (right). Scene texts are not considered in our work.

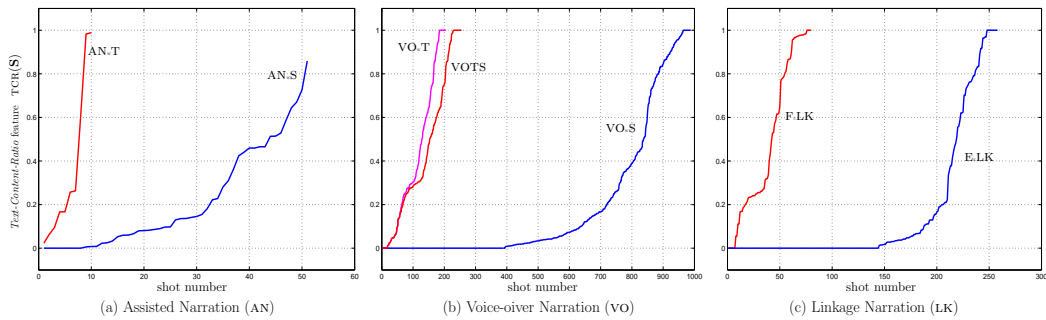


Figure 3-9: Examples of TCR features plotted within different structural categories.

and are, therefore, relatively easy to detect. In this work, the algorithm described in (Shim *et al.*, 1998) is used for detecting the texts in the still frame.

However, if this algorithm is applied directly, the error rate is high, as it was originally designed for general purpose use and made scalable to variations such as font size/style, complex backgrounds. The results from this detector are therefore further enhanced by applying the following three rules. First, with the dimension of a frame being 352×288 , a detected text line height or width must be greater than 10 pixels to remain valid. Secondly, the width must be more than twice the height; and lastly, the level of contrast between the letters and its background must exceed a threshold. For a representative sequence $\Gamma(\mathbf{S}) = \{\tau_i\}_{i=1}^M$, the following features are defined:

- (1) The *Text-Content-Ratio* (TCR) feature: $\text{TCR}(\mathbf{S}) \triangleq \frac{\sum_{i=1}^M \mathcal{T}(\tau_i)}{M}$, where $\mathcal{T}(\tau_i)$ is the text detection indicator function, which returns 1 if captioned texts are detected in frame τ_i , and return 0 otherwise.
- (2) The *Text-Bounding-Box* features. These are the means, $\{\mu_X(\mathbf{S}), \mu_Y(\mathbf{S})\}$, and variances, $\{\sigma_X^2(\mathbf{S}), \sigma_Y^2(\mathbf{S})\}$, of the text bounding box center (similar to the face bounding box discussed before).
- (3) The *Text-Area-Ratio* feature: $\text{TAR}(\mathbf{S}) \triangleq \frac{\sum_{k=1}^{K_{\Gamma(\mathbf{S})}} v_k}{K_{\Gamma(\mathbf{S})} \times Z}$, where v_k is now redefined as the area of the text bounding box instead.

Figure-(3-9) shows an example of the feature TCR plotted for different narrative categories. It is evident from this figure that structures, those are defined based on the assumption of existence of texts (eg: ANwT), significantly have higher values of TCR compared to other structures (eg: ANwS).

3.3.2 Audio Content Analysis

For the classification of a shot into direct-narration, voice-over, linkage, etc., information from the face and text detection alone do not suffice. We further extract features from the audio track. The audio track is sampled at 44.1kHz with mono channel and divided into a sequence of non-overlapping clips of 0.25s duration. Audio features are then extracted using the energy in seven sub-bands of the discrete wavelet transform with 6 levels of decomposition and 14-order cepstral coefficients obtained from the LPC coefficients. Details for the extraction of this feature can be found in (Phung *et al.*, 2002). This forms an aural feature vector consisting of 21 elements from each clip.

Let $\Lambda(\mathbf{S}) = \{\varsigma_i\}_{i=1}^Q$ be the sequence of Q consecutive audio clips that make up the sound track of the shot \mathbf{S} . In a supervised mode, we classify each ς_i into one of the following labels: *vocal speech*, *music*, *silence*, and *non-literal* (NL) sound. The ratios of the number of clips classified in each category to the total Q are used as the associated audio features. We thus have four aural features namely: speech-ratio, music-ratio, silence-ratio and NL-ratio. That is, for example, if there are P clips classified as *music*, then the music-ratio(\mathbf{S}) is $\triangleq P/Q$.

Figure-(3-10) and Figure-(3-11) plot the averaged values of the speech-ratio and music-ratio features respectively. The dominance of speech-oriented classes: ON, VO, and SN when compared to the linkage narration (LK) is evident for the speech-ratio. However, with respect to the music-ratio (Figure-(3-11)), the expressive linkage (E_xLK) dominates all other categories.

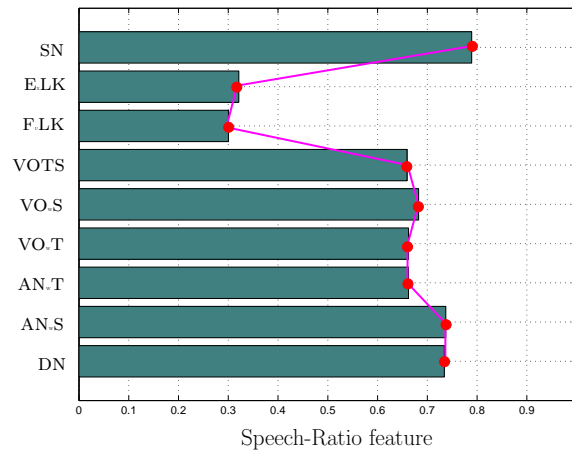


Figure 3-10: Averaged values of the speech-ratio feature.

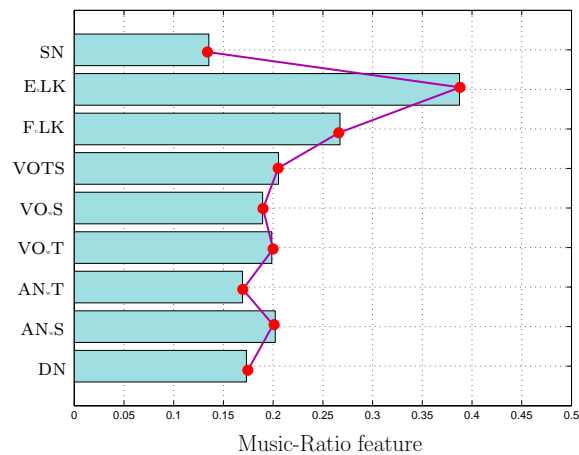


Figure 3-11: Averaged values of music-ratio features.

3.4 Constructing the Classification System

Given that we have tailored a meaningful set of structural units, what is required next is the means to recognise these structures automatically. In this section, we present a supervised approach using the well-known C4.5 algorithm⁴ to construct a decision tree system to recognise the structural hierarchy at multiple levels.

3.4.1 Data and Groundtruth

The data set is collected from 15 training and classroom teaching videos. Shot indices are first derived, then each shot is manually labeled as one of structures shown in Figure-(3-1) to form the groundtruth. The statistics for the groundtruth are shown in Table-(3.1). It

BOTTOM LEVEL			
DN	126		
AN _w T	10		
AN _w S	51		
		TOP LEVEL	
E _x LK	258	ON	187
F _v LK	80	LK	338
SN	10	VO	1243
VO _w T	50	SN	10
VO _w S	988		
VOTS	205		

Table 3.1: Statistics of shots in the groundtruth for each structural type.

is worth drawing attention to the difference in the distribution of various structural units in educational videos from the groundtruth statistics. This data set reflects empirical statistics where we observe the dominance of the VO category, then the LK narration, and lastly the ON narration. We note very few cases of the SN narration. Among them, ON is the most meaningful group in terms of carrying semantics across structures as it highlights important events.

3.4.2 Results: One-Layer Classification

We report two experimental results corresponding to the two levels in the proposed hierarchy. In the first experiment, the C4.5 algorithm (Quinlan, 1993) is used to train a

⁴During the experiment, we have used algorithms other than C4.5 including Support Vector Machines (SVM) and K -nearest neighbour (K -NN). However, C4.5 far outperforms its counterparts, therefore, we only include results from C4.5 in this chapter.

decision tree for across all nine classes: { DN, AN_{wT}, AN_{wS}, VO_{wT}, VO_{wS}, VOTS, ExLK, FuLK, SN} (cf. Figure-(3-1)). The results when testing on the training data show an accuracy of 90.3%, leading to a confusion matrix shown in Table-(3.2). To yield an unbiased result on the unseen data, we use 10-fold cross validation on this data set. The average accuracy obtained from this validation process across all nine structures is 73.3%. The True Positive (TP) and False Positive (FP) rates within each class is also reported in Table-(3.2). We observe low TP-rates for classes AN_{wT}, AN_{wS}, and SN. This can be explained by the data set as these classes are small compared with other categories, and thus dominated by the larger classes. However, despite the fact that it suffers from overfitting, the TP-rate of the most important structure, namely the DN is high. Overall, we note that the mis-

ExLK	FuLK	DN	AN _{wT}	AN _{wS}	VO _{wS}	VO _{wT}	VOTS	SN	← CLASSIFIED AS	
200	3	1	0	0	52	0	2	0	ExLK	Expressive Linkage
5	69	0	0	0	5	1	0	0	FuLK	Functional Linkage
0	0	121	0	1	4	0	0	0	DN	Direct Narration
0	0	2	6	1	0	0	1	0	AN _{wT}	Ass. Narr. w. Texts
0	0	1	0	46	3	0	1	0	AN _{wS}	Ass. Narr. w. Scenes
19	4	2	2	2	947	4	8	0	VO _{wS}	VO w. Scenes
1	0	1	0	0	1	45	2	0	VO _{wT}	VO w. Texts
4	2	2	0	0	31	0	166	0	VOTS	VO w. Text/Scenes
1	0	0	0	0	2	0	1	6	SN	Supportive Narration

ExLK	FuLK	DN	AN _{wT}	AN _{wS}	VO _{wS}	VO _{wT}	VOTS	SN	←	TP rate	FP rate
138	9	1	2	3	100	0	5	0	ExLK	53.49	46.51
11	45	0	0	0	20	2	1	1	FuLK	56.25	43.75
0	0	97	3	5	18	1	1	1	DN	76.98	23.03
1	0	2	2	3	2	0	0	0	AN _{wT}	20.00	80.00
4	0	9	2	17	13	1	4	1	AN _{wS}	33.33	66.67
40	9	14	5	10	861	17	32	0	VO _{wS}	87.15	12.85
1	3	0	0	2	13	26	5	0	VO _{wT}	52.00	48.00
7	5	1	2	3	70	4	113	0	VOTS	55.12	44.88
0	0	2	0	1	3	0	1	3	SN	30.00	70.00

Table 3.2: Results from evaluation on the training data (top) and 10-fold cross validation (bottom).

classification between ExLK and VO_{wS} accounts for the main source of errors (off-diagonal coloured cells). This may be attributed to the inaccuracy in separating vocal speech from music as this is the only clue to distinguish between the two classes. Further examination of the videos shows that confusion between on-screen narration and voice-over segments is generally due to the following scenarios:

- A voice-over shot with the presence of many faces, such as people in a meeting, is misclassified as on-screen narration.
- Errors in face detection leads to a misclassification. For example, if a face in a DN shot fails to be detected, then it will be classified as VO.

Given the fact that the classification process is heavily affected by the accuracy of the face/text detection process, the results have shown an adequate accuracy, especially in the classification among similar structures such as between AN_wT and DN or between VO_TS and VO_wS.

3.4.3 Results: Two-Tiered Classification

Based on the error analysis and motivated by the fact that the querying interest could be of a hierarchic nature, we propose a two-tiered classification system. We could then answer queries like ‘give me all segments with direct instructions’ that could then be interpreted as giving the location indices of the ON and FuLK sections. Thus, in the second experiment, we construct a tiered-classification to study the discrimination performance at different levels of resolution. At the top level, the decision tree (D-TREE) classifies a shot into four main categories: on-screen, voice-over, linkage, and supportive narrations. Next, we train an appropriate D-TREE within each class (cf. Figure-(3-1)). Again a 10-fold cross validation is used to evaluate the performance of each classification process on unseen data, and C4.5 is used as the learning algorithm. Table-(3.3) summarises the performance across the two levels.

		WITHIN CLASS CLASSIFICATION RESULTS				ACCURACY		
						Learning	10-fold	
TOP LEVEL		ON	VO	LK	SN	91.73%	85.1%	
	On-screen Narration	ON	139	44	4			0
	Voice-over Narration	VO	32	1167	43			1
	Linkage Narration	LK	3	132	202			1
	Supportive Narration	SN	2	4	0			4
SECOND LEVEL								
On-screen		DN	ANwT	ANwS		96.1%	73.3%	
	Direct Narration	DN	106	2	18			
	Ass. Narr. w. Texts	ANwT	3	3	4			
	Ass. Narr. w. Scenes	ANwS	21	2	28			
Voice-over		VOwT	VOwS	VOTs		94.8%	88.3%	
	VO w. Texts	VOwT	33	10	7			
	VO w. Scenes	VOwS	13	946	29			
	VO w. Text/Scenes	VOTs	83	3	119			
Linkage Narr.		ExLK	FuLK		97.0%	89.7%		
	Expressive Linkage	ExLK	243	15				
	Functional Linkage	FuLK	20	60				

Table 3.3: Classification results at each level of the hierarchy.

The results from Table-(3.3) clearly show an improvement in the performance using this ‘divide-and-conquer’ strategy. Classification results at the top level are adequate with an accuracy of 85.1% in a 10-fold evaluation. The classification results at the lower levels are also good at an average of 83.8%, in which the results for class LK is best with an

accuracy of 89.7%. Even though the classification within the class DN is lowest, we have seen a slight improvement for the sub-class AN_wS from the previous one-layer experiment. Results for other classes, such as SN or AN_wS, have also improved slightly.

3.5 Closing Remarks

Identifying and recognising potentially meaningful and useful structural units are essential to video content management systems. In this chapter, our work has developed techniques for the identification of a hierarchy of narrative structures in educational videos. We propose a set of audiovisual features to differentiate and classify shots into nine distinct structures. Important structures in these videos have been recognised with an acceptable accuracy. We have also represented a tiered classification system to recognise structures at different levels of granularity.

The main contributions from this chapter are the identification and construction of a hierarchy of narrative structural units and the classification system that goes with it. In the next chapter, we further exploit Film Grammar to extract *expressive* elements and use them for the segmentation of an educational video into a two-level hierarchy of topical content.

Chapter 4

Expressive Functions and Segmentation of Topical Content

In the previous chapter, we have presented an exposition into the identification of a meaningful hierarchy of structural units for the domain of educational videos. In this chapter, we delve deeper into Film Grammar, seeking higher-order semantics and investigating their use for segmentation of topical content at two conceptual levels: (main) topic and subtopic. We discuss commonly observed rules and patterns present when topics and subtopics are presented in these videos. Based on the observation that subtopic boundaries coincide with the ebb and flow of the *density* of content shown, we formulate a *content density* function. We examine the behaviour of the function to determine how well it relates to the subtopic boundaries. We then propose two methods for determining subtopic boundaries. One is based on heuristics and the other is based on a probabilistic measure. Our experiments on several instructional videos are presented to demonstrate the effectiveness and robustness of these subtopic detection schemes.

Next, we continue the explorative journey with two high-level expressive constructs, namely the *thematic* and *dramatic* functions of the educational content. Again, drawing on the extensive body of Film Grammar, media production rules and conventions used by filmmakers, we hypothesise key aesthetic elements that influence the *perception* of these expressive constructs in educational videos. Computational models for these functions are then formulated, and an evaluation of the performance of these functions is presented on several industrial safety training videos.

Given the set of these three key expressive functions, our last piece of work presented in this chapter aims at developing a framework to combine production grammar with these functions for a hierarchic detection of topics at two levels, namely: main topics and subtopics. Observing that a drop or rise in the mediation process is a deliberate decision in the ed-

educational video genre to emphasise key topics, we hypothesise the relationship between the thematic function and the nature of main topic boundaries and develop an edge-based detection algorithm to detect main topic indices. The experimental results demonstrate the effectiveness of the scheme when applied to a set of ten industrial instructional videos.

We would like to emphasise at the outset that even though the motivation for constructing expressive functions is primarily for the problem of segmentation, they themselves are a set of self-contained high-level semantic units that imply a range of potentially useful applications such as for query or navigation. For instance, a query such as “find me segments which provoke feelings” can be translated as the location of extreme points in the dramatic function.

The layout of the remaining text is organised as follows. The nature of topical content in educational videos is presented in Section 4.1, where we formulate two key hypotheses for topic and subtopic detection. The development of the content density function and a framework for subtopic detection is provided in Section 4.2 along with the experimental results. Next, the thematic and dramatic functions are formally studied and computed in Section 4.3. Partitioning the videos into main topics is presented in Section 4.4 and the results are provided in Section 4.4.1. Finally, we present a summary of the chapter with some concluding remarks in Section 4.5.

4.1 The Nature of Topics and Subtopics

We have emphasised previously that educational videos are a highly structured medium aiming to have the maximum impact on the audience. Consequently, the organisation of content is one of the most deliberate choices made by the filmmaker. The most popular style of organisation is the expository narration as we have studied in Section 3.1. Filmed material is generally organised into a sequence of topics with embedded subtopics. How then does a filmmaker distinguish these categories? There can be no firm answer and the boundary may be unclear. However, an answer to this question may be found in the literature on educational film making which tells us that the engagement of a subjective point of view plays a key role in the development of the content (Davis, 1969; Foss, 1992). That is, shots in which the narrator engages with the audience by directly speaking to the viewer make *critical* sections where main topics begin, or conclude. They reflect portions of the videos where the filmmaker decides to “step in” and interfere with subject being shown. It reflects directly the higher level of mediation by the filmmaker. This leads us to the following hypothesis.

Hypothesis 4.1 *In educational videos, (main) topic boundaries are likely to coincide with large changes observed in the mediation levels by the filmmaker.*

Mediation level is re-defined in our context as the involvement of the filmmaker via some direct channels such as speech (voice-over), displayed caption texts or, in the extreme cases, the narrator chooses to appear on the screen and speak directly to the audience. Hypothesis 4.1 essentially implies that topic boundaries are centered around the time when more “messages” are delivered by the filmmakers. That is, if there is a function that can encode the level of mediation, then topic boundaries are more likely to be found at points of significant change in the mediation function.

Subtopics, on the other hand, reflect the ebb and flow of the content that is being shown. Since the rate of delivery of content is deliberately manipulated to punctuate the content, sections where we observe a drop or increase in this rate is likely to mark important transition points, and thus generally coincident with smaller, coherent sections of material, such as subtopics. Based on this knowledge and empirical analysis of several videos in this genre, we propose the following hypothesis for subtopic boundaries.

Hypothesis 4.2 *In educational videos, subtopic boundaries are likely to be coincident with a rapid drop, then rise (a valley) in the content density presented by the filmmaker.*

In the following sections, we formulate a set of *expressive* functions to capture the mediation level and the content density. Based on these functions we develop a framework to segment a video into topics and subtopics.

4.2 Partitioning Videos into Subtopics

In this section, we formally construct the content density function and facilitate two algorithms to segment a video into subtopics.

4.2.1 The Content Density Function

Related work in the study of viewing perception in motion pictures dates back to as early as 1971 (eg: Penn (1971); Isenhour (1975)). However, it is only lately that with the aid of computer power, computational models have been developed. For example, the idea of

cutting-rate-effect in (Penn, 1971) was computationally formulated as a movie *tempo* function in the work of Adams (2003a). The problem of extracting *expressive* functions in motion picture has been re-visited and studied lately along with the introduction of the Computational Media Aesthetics (CMA) framework proposed by Dorai and Venkatesh (Dorai and Venkatesh, 2002). Visual tempo (Adams, 2003a) and sound tempo (Moncrieff, 2004), as reviewed Section 2.2.4, are the two known expressive functions in the literature. In this section, we lengthen this list with the *content density function*.

So what, then, should be defined as the content density of a video? Conceptually, it implies a rather high level of expressiveness. In fact, colour, lighting, editing, motion, sound, changes in fields of views, camera angles, all contribute to the ‘energy’ of a scene, which in turn affect the perception of the density. In his book on ‘Applied Media Aesthetics’, Zettl (Zettl, 1999) defines *density* as the “degree of detail occurring within a period of time”. Accordingly, a high density event is one in which many details are delivered within a relatively short duration. This is generally captured in scenes in movies by the use of extreme changes in the field of view, extreme close-ups, and fast editing. Music is sometimes employed to augment these scenes during which narration is stopped. High density events, therefore, make the viewing experience brief (Zettl, 1999). Low density events, on the other hand, impact the viewers with a perception of ‘time-stretching’ and generally draw their attention out. Low density scenes generally have long takes, little editing, and low sound energy.

Using these commonly employed video production practices as the underlying principles, we propose that the key contributing factors to manipulating the content density are editing (captured by shot length), motion within the scene, and sound energy. In the educational context, fast editing means many messages are shown in a short period of time, such as quick safety techniques or short explanations in the lecture. High motion and sound energy imply that there is a lot of activity in the scene. Analogous to the work of Adams et al. (Adams *et al.*, 2000) on the computation of movie tempo, we propose a new construct called the *content density function* $\mathbf{Dn}(n)$ defined as:

$$\mathbf{Dn}(n) = \alpha \frac{S^*(n) - \mu_{S^*}}{\sigma_{S^*}} + \beta \frac{M(n) - \mu_M}{\sigma_M} + \gamma \frac{A(n) - \mu_A}{\sigma_A}. \quad (4.1)$$

where n is the shot number in a video, $S^*(n)$ is the inverse of the shot length, $M(n)$ is the average shot motion and $A(n)$ is the average sound energy; $(\mu_{S^*}, \sigma_{S^*})$, (μ_M, σ_M) , (μ_A, σ_A) are the means and standard deviations of S^* , M and A respectively; α , β , and γ are the respective weighting factors for these contributors. When no further knowledge is available, we assume uniform weights for these factors.

As an illustrative example, Figure-(4-1) plots the content density functions computed for the

educational video ‘Against All Odds’ and the training video ‘Eye-Safety’. These functions are then smoothed with a Gaussian filter. To highlight Hypothesis 4.2, in Figure-(4-1b), the content of the video is annotated with the density function curve, and the connection between the ebb and flow of this function with topical transitions is evident.

4.2.2 Subtopic Boundary and the Curvature of the $Dn(\cdot)$ Function

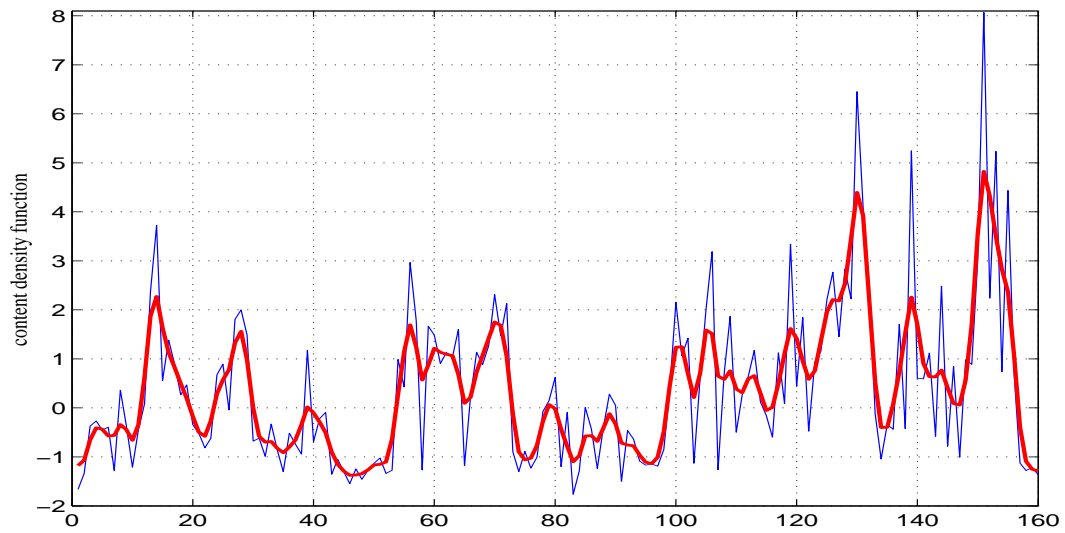
As stated in Hypothesis 4.2, we found that the flow of the density function over time reveals salient information about the structure of the content presented. In instructional videos, for example, a drop from high density to low density for a few shots is deliberate and used to draw attention to the material presented, and to let learners absorb it. For example, safety training videos will show an accident, and then drop the density at the end of this segment to emphasise the gravity of the accident. We hypothesise that *sharp* and *abrupt* changes in content density are indicative of subtopic switches or changes. Essentially, it means that a subtopic boundary corresponds to *a rapid drop followed by a rise* in the density function. Figure-(4-3) shows the smoothed content density function for the video ‘House-Keeping’, where vertical solid lines indicate subtopic boundaries. More examples are shown in Figure-(4-1). We further note the following patterns used in educational videos.

Remark 4.1 *In educational videos, a subtopic introduction coincides with the start of a new shot.*

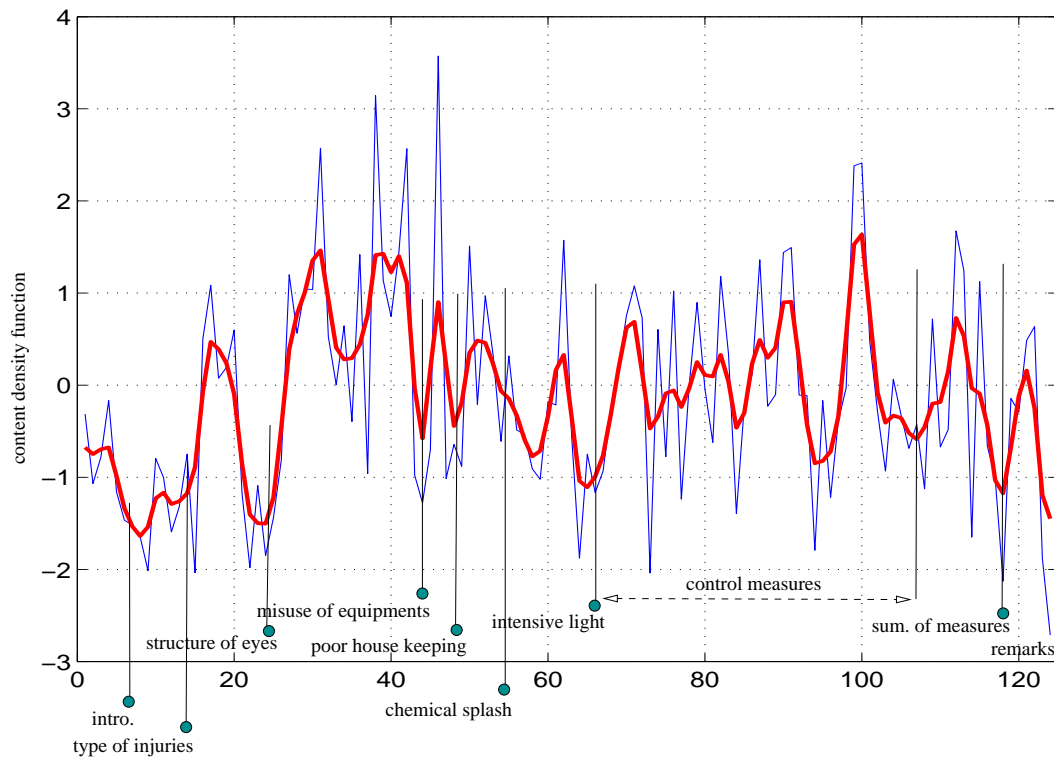
Remark 4.2 *Subtopics often start with either a shot of on-screen narration (narrator speaking directly to viewers), or with a shot containing superimposed text.*

The first remark implies that, to start a new subtopic, the director will choose to do so in a new shot. This eliminates the case when a subtopic is introduced during the running of a camera shot, which is very rare in our video domain. The second remark further strengthens the context in which a subtopic is started. Typically, a narrator will appear to introduce the subject or the subject headline will be displayed.

Thus, to segment educational videos into subtopics, we first identify the minima of the density function (Hypothesis 4.2). Since not all minima correspond to subtopic changes, we incorporate the domain knowledge that a minimum in the vicinity of an ON or FOLK shot is likely to correspond to a subtopic boundary (Remarks 4.1-4.2). To exploit this hypothesis we propose two methods. The first is an algorithm based on simple heuristics and the second is formulated in a probabilistic framework.



(a) 'Against All Odds'



(b) 'Eye Safety'

Figure 4-1: Content density functions (original and smoothed versions) plotted for educational video 'Against All Odds - Part 4' and training video 'Eye-Safety'.

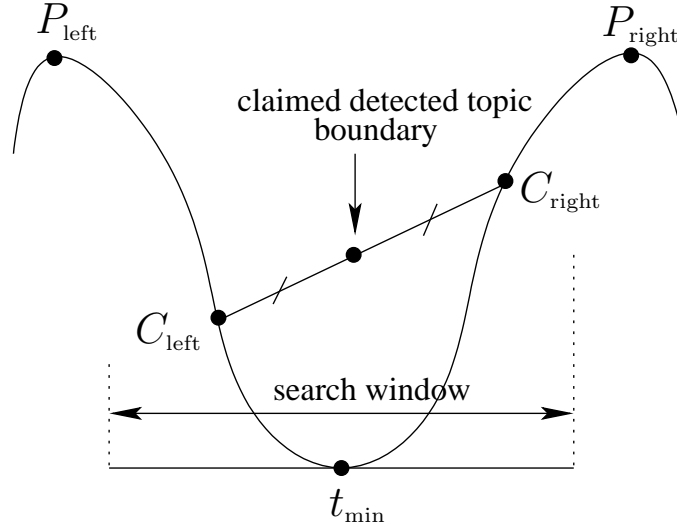


Figure 4-2: In the heuristic approach, the mid-point between C_{left} and C_{right} is labeled as the temporal index of a subtopic boundary.

Algorithm 4.1 Heuristic approach to subtopic detection

Input: a video \mathbf{V} along with extracted primitive shot-based features including: shot indices (shot length), motion, audio energy; and the list of candidate shots.

E1 [*Content Density Function*]. Compute the shot-based content density $\mathbf{Dn}(\cdot)$ as in Eqn. (4.1).

E2 [*local minima/maxima*]. Smooth $\mathbf{Dn}(\cdot)$ with a Gaussian filter and subsequently detect all local minima and maxima.

E3 [*Detection Loop*]. Perform steps *E4* to *E6* for each local minimum:

E4 [*time info*]. Let t_{min} be the time index at a local minimum, and let P_{left} and P_{right} be the left and right peaks in the content density function around t_{min} respectively.

E5 [*search window*]. A search window of size w centered around t_{min} is constructed. Candidate shots are searched for within the window. If the window encompasses P_{left} or P_{right} , the search is truncated at these points.

E6 [*detection*]. If no candidate shot is found, then no subtopic boundary is detected and the local minimum is labeled as a miss and discarded. Otherwise, let C_{left} and C_{right} be the leftmost and rightmost candidate shots found. A subtopic boundary is then deemed to be detected at temporal location t_{tp} given as the mid-point between C_{left} and C_{right} (Figure-4-2).

Output: List of detected subtopic indices.

4.2.3 Heuristic Approach to Subtopic Detection

The initial processing of a video includes detection of shot transition indices, computation of shot motion, and sound energy. Next, shots are analysed to determine all on-screen (ON) narration and superimposed text (FvLK) segments¹. We term ON and FvLK shots as *candidate shots* (CS). The detection algorithm proceeds as follows. First the content density function is computed for a video \mathbf{V} and smoothed with a Gaussian filter. All local minima and maxima are located based on the first derivative. We then search for the subtopic index within each local minimum valley within a window of size w , and the search is truncated should it exceed the boundaries of the enveloping maxima. A subtopic boundary is deemed to be found as the midpoint of the leftmost and rightmost candidate shots within this search area as shown in Figure-(4-2). The complete algorithm is outlined in Algorithm 4.1.

4.2.4 Probabilistic Approach to Subtopic Detection

In this section we explore a probabilistic approach to subtopic segmentation. Let t_{\min} be a local minimum of the density function encompassed by two surrounding peaks P_{left} and P_{right} . Within the temporal segment from P_{left} to P_{right} , let t_{tp} be the location of the subtopic boundary and $\{\Phi_{c(1)}, \dots, \Phi_{c(N)}\}$ be an observed sequence of candidate shots. Each $\Phi_{c(i)}$ is associated with a temporal index $\Phi_{c(i)}^{\text{time}}$ and has a density magnitude $\Phi_{c(i)}^{\text{mag}}$. We assume that only one of these candidate shots contributes to the location of the subtopic boundary and we call this t_c . Then, the relation between t_c and Φ_c is formulated as:

$$\Pr(\Phi_c | t_c) = \begin{cases} 0 & \text{if } t_c \neq \Phi_{c(i)}^{\text{time}}, \forall i = 1, \dots, N \\ \propto \Pr(\Phi_{c(i)}^{\text{mag}}) & \text{if } t_c = \Phi_{c(i)}^{\text{time}} \end{cases} \quad (4.2)$$

Our objective is to compute $\mathcal{L} \triangleq \Pr(\Phi_c, t_{\min} | t_{\text{tp}})$: the likelihood of the observed information Φ_c and t_{\min} given the current hypothesis for t_{tp} . This is given as,

$$\begin{aligned} \Pr(\Phi_c, t_{\min} | t_{\text{tp}}) &= \sum_{t_c} \Pr(\Phi_c, t_{\min}, t_c | t_{\text{tp}}) \\ &= \sum_{t_c} \Pr(\Phi_c | t_{\min}, t_c, t_{\text{tp}}) \Pr(t_{\min}, t_c | t_{\text{tp}}). \end{aligned} \quad (4.3)$$

¹Both of which are discussed and computed in our narrative structural analysis in the previous chapter

In essence, the sequence Φ_c is the evidence of t_c and we assume that it depends only on t_c . Thus, the RHS of Equation-(4.3) can be simplified to:

$$= \sum_{t_c} \Pr(\Phi_c | t_c) \Pr(t_{\min}, t_c | t_{\text{tp}}). \quad (4.4)$$

We further assume that t_{\min} and t_c are independent. Thus Equation-(4.4) reduces to:

$$= \sum_{t_c} \Pr(\Phi_c | t_c) \Pr(t_c | t_{\text{tp}}) \Pr(t_{\min} | t_{\text{tp}}). \quad (4.5)$$

From Equations (4.2), (4.3), (4.4) and (4.5), we finally achieve:

$$\mathcal{L} \propto \sum_{i=1}^N \Pr(\Phi_{c(i)}^{\text{mag}}) \Pr(t_c = \Phi_{c(i)}^{\text{time}} | t_{\text{tp}}) \Pr(t_{\min} | t_{\text{tp}}). \quad (4.6)$$

In summary, given t_{\min} and an observed sequence Φ_c from the time index P_{left} to P_{right} , if Φ_c is empty, in other words there is no evidence for t_c , then no subtopic boundary is detected. Otherwise the subtopic boundary is determined as the location of the candidate shot that gives the maximum value for \mathcal{L} computed as in Equation-(4.6).

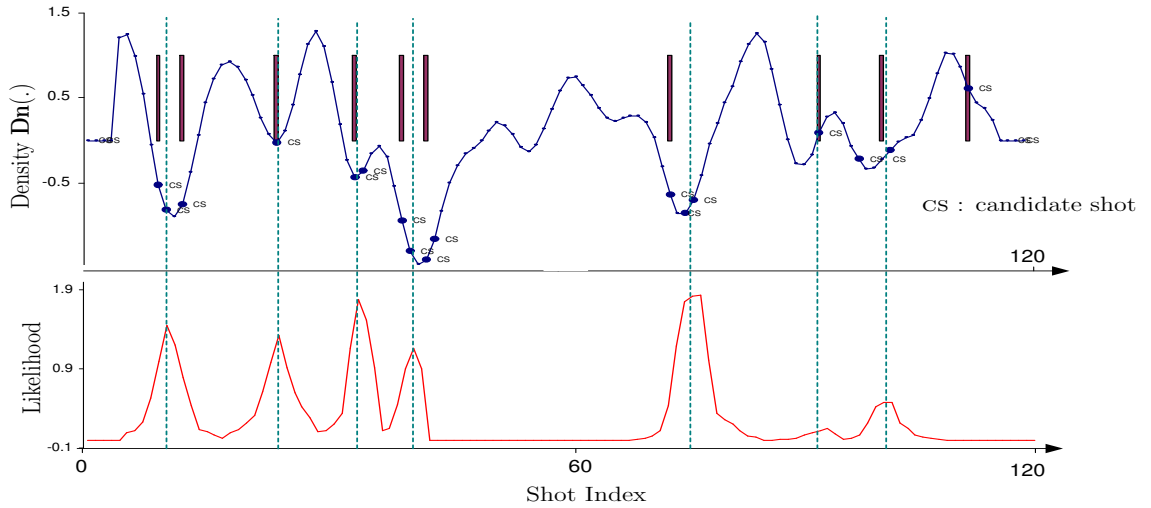


Figure 4-3: The density function and likelihood for the video, ‘House-Keeping’. Vertical solid lines are actual subtopic boundaries and vertical dashed lines are detected boundaries.

4.2.5 Experimental Results

Ten safety and teaching films are used in our experiments. Shot indices are first detected with the WebFlix software (Mediaware-Company, 1999) and all shot detection errors are corrected manually. Raw motion per shot is extracted as average values of camera pan and tilt using software implementing the qualitative motion estimation algorithm of (Srinivasan *et al.*, 1997). The audio-track is re-sampled at 44.1kHz with mono channel, and the sound energy is estimated as the root mean square (RMS) value. Subtopic boundaries for each video are manually determined to form the groundtruth.

4.2.5.1 Results from the Heuristic Approach

We use two well-known quantities namely *recall* and *precision* to measure the performance of our detectors. The averages obtained across all videos for different values of window size, w is plotted in Figure-4-4.

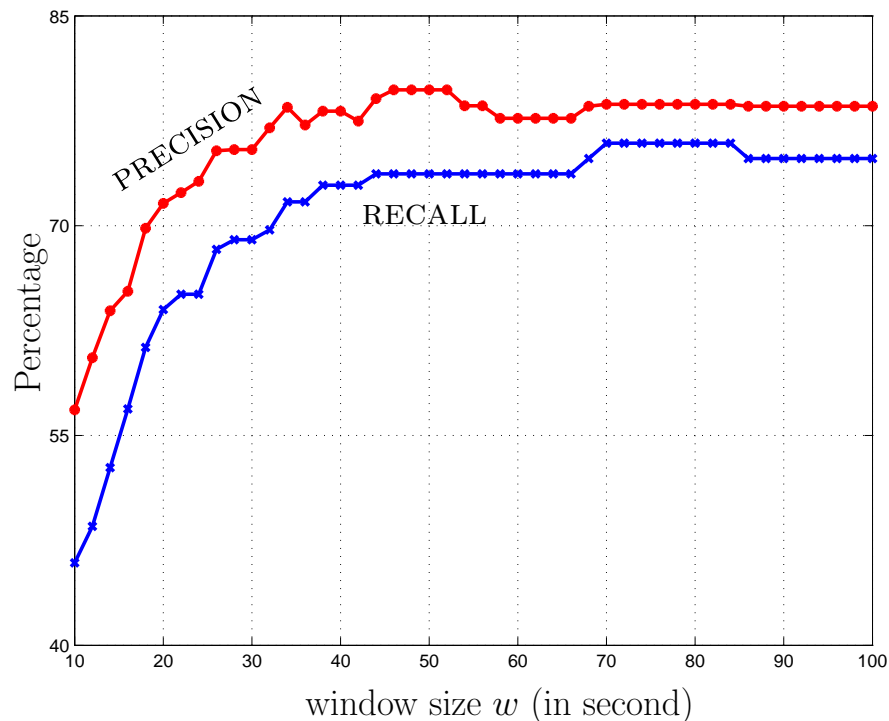


Figure 4-4: Average precision and recall values for various window sizes computed from a set of 10 videos.

The best result achieved for recall is 75.9% when w ranges from 70–84s; and for precision, it is 79.72% when w ranges from 44–52s. On an average, the detector performs best with

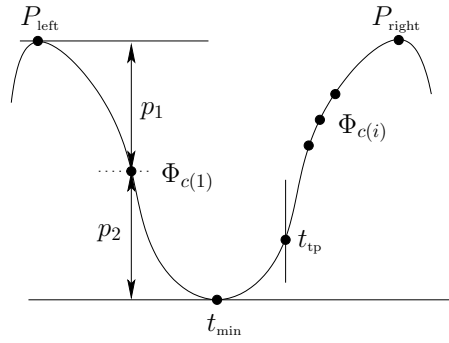


Figure 4-5: Magnitude of a candidate shot $\Phi_{c(i)}$ is computed as $\frac{p_1}{p_1+p_2}$.

$w = 70s$, and the average recall and precision are 75.9% and 78.7% respectively. The result for each individual video is presented in Table-(4.1) (see \mathcal{E}^*). The average distance error is computed as the temporal distance between the detected boundary and the real boundary. In this case, this error is 12.45s.

4.2.5.2 Results from the Probabilistic Approach

In the training stage, we collect the following data:

1. Let p_1 and p_2 represent the difference in magnitude of a candidate shot to the nearest maximum and minimum respectively. The effective magnitude for that shot is then given as: $p_1/(p_1 + p_2)$ (Figure-4-5). This is the source to model $\Pr(\Phi_c^{\text{mag}})$.
2. The distances in frames between a candidate shot to its nearest subtopic boundary. This is the source data to model $\Pr(t_c|t_{tp})$.
3. The distances in frames between a local minimum and a subtopic boundary. This is the source data to model $\Pr(t_{\min}|t_{tp})$.

MATLAB is used to find the best fitting distribution model for each of these data collections. We found that:

- Distribution of $1 - \Phi_{c(i)}^{\text{mag}}$ is well modeled with an *exponential* distribution ($\bar{\mu} = 0.3049$).
- $\Pr(t_{\min} | t_{tp})$ is modeled with a *Gaussian* ($\mu = 0, \sigma = 22.35s$).
- $\Pr(t_c | t_{tp})$ is modeled with an *exponential* distribution ($\bar{\mu} = 22.09s$).

Using our likelihood-based subtopic detection algorithm proposed in Section 4.2.3, the results are reported in Table-(4.1) (marked with \mathcal{P}^*). As can be seen, we have obtained an average recall and precision of 78.3%% and 83.4% respectively. The average distance error in this case is 9.72s.

Video	Recall (%)		Precision (%)		Avg. Distance Err (s)	
	\mathcal{P}^*	\mathcal{E}^*	\mathcal{P}^*	\mathcal{E}^*	\mathcal{P}^*	\mathcal{E}^*
1	73	64	100	100	7.9	9.7
2	67	56	67.0	62.9	10.14	12.9
3	78	89	78.0	89.0	7.81	9.8
4	80	80	76.2	50.0	6.43	13.7
5	64	73	78.0	89.0	1.18	11.4
6	89	89	85.6	66.9	12.41	12.4
7	86	71	92.5	91.3	11.41	7.4
8	85	79	92.9	100	13.65	18.7
9	79	75	84.0	66.4	16.9	19.8
10	82	83	80.4	71.6	8.3	8.5
Avg.	78.3	75.9	83.4	78.7	9.72	12.45

Table 4.1: Average results for subtopic detection with probabilistic (\mathcal{P}^*) and heuristic (\mathcal{E}^*) approaches using a window size = 70s.

Discussion. The performances of both algorithms are relatively comparable, with a slightly better performance recorded for the probabilistic approach. A close analysis of the results shows that the sources of false positives and misses in subtopic detection during our experiments are the following:

- There are more than two subtopic boundaries within a local minimum t_{\min} encompassed by P_{left} and P_{right} . This case generally results in false negatives. We also discover that subtopics in this case are usually composed of a relatively small number of shots. They typically deliver information in the same manner, for example, using only voice-overs with similar visual content. Often, no demonstration scenes are observed in these cases. The content density of such a sequence of shots therefore stays flat, resulting in a failure of our method.
- A false positive (or a miss) in detecting candidate shots may lead to a false positive (or a miss) in our subtopic detection scheme.

In comparing the two schemes, while the heuristic method may be slightly faster in computation, the probabilistic framework performs better, and more importantly, offers us a robust scheme with a better underlying foundation. One obvious advantage lies in the ease in incorporating new knowledge when needed into the detection scheme.

4.3 Thematic and Dramatic Functions of Video Content

As stated earlier, the mediation level, or the engagement of the filmmaker, is an important form of expressiveness for the extraction of meaningful events such as, in our case, the beginning of a subtopic or topic. In this section, we define the mediation functions for educational videos and study the computational forms.

So how should mediation functions be derived for the video media? Again, we seek guidance from Film Grammar to distil out the constituent elements. We start with the perspective of the ‘what’ and ‘how’ in films. We re-emphasise that educational films are “motion pictures designed to teach” (Herman, 1965), and a well-crafted segment that motivates learners to actions or enables them to retain a message that can last in their memory cannot be a trivial shooting. Numerous aesthetic choices must be made by the director in video production and the issue is not only *what* is to be shown but also *how* it is to be presented to achieve the maximum impact on learners.

Expression Planes	General Films	<i>Educational Films</i>
WHAT		
we see and hear	physical appearance, acting, costume and make-up, setting, props, time, weather, physical relations, movement, real colours, natural light, real sound, real music, dialogue	appearance of narrators, teachers, or presenters, motion in the scenes, use of music, speech, voice-over and the use of super-imposed texts
HOW		
we see and hear	format, shot composition, focal distance and definition size of shot, camera angle, camera movement, colour and monochrome, artificial light, type of film and exposure special photographic effects, editing, sound effect, texts, subtitles	use of colour (variations in hue, saturation, lightness values) camera pan, tilt, zoom, sound effects (expressive silence)

Table 4.2: Aesthetics elements for expression in general films suggested by Foss (1992), and our suggested mappings to educational films.

Addressing film techniques in general, Foss (Foss, 1992) approaches the *what* and *how* in films from two distinct views of what he calls, *Plane of Events* and *Plane of Discourse*. Elements that “can or could be perceived by the characters in the film or program” belong to the *Plane of Events*, whilst the *Plane of Discourse* contains factors that “are imperceptible to the characters in the film”. Hence, in the context of educational videos, contents and materials presented are captured in Foss’s *Plane of Events* and different techniques are employed to bring them to the viewers.

What media elements are then available for the filmmakers to manipulate these *planes*? Table (4.2) shows a partial list of suggested factors for general films (Foss, 1992), and our suggested mappings to the educational domain. Obviously, this list comes from the viewpoint of a filmmaker, and not all of them are applicable or amenable to our computing framework. The element ‘weather’, for example, is still far from being automatically extractable by computers. Limiting our analysis to those computable elements and the educational domain, we propose a list of elements for further study as shown in Table-(4.3).

	THEMATIC	DRAMATIC
Colour	● simple background is used to avoid distraction from displayed texts and narrators.	○ usually diversity in color.
Motion	○ less movement and activities is observed.	● lots of motion due to action in the scenes.
Music	○ mainly narration voice or silence (eg: with scrolling texts).	● often music is added to create the mood.
Voice-over	● voice of authority to narrate/explain the contents.	○ narration is usually stopped.
Sup. Texts	● introduce subjects, remarks, emphasise keypoints, reminder messages, etc.	○ scene texts may be observed but not superimposed texts (<i>see</i> Section 3.3.1.2 for the difference).
Narrators	● appears to speak directly to the viewers, to ‘mediate’ the subject.	○ usually there is no narrator on the screen.
Impact Codes	● : have a <i>high</i> impact on the expressive function. ○ : have a <i>low</i> impact on the expressive function. ○ : <i>rarely</i> impact on the expressive function.	

Table 4.3: Media elements and their utility in conveying thematic and dramatic functions of content in educational video.

Given the set of available aesthetic choices to express the *what* and *how* in motion pictures, Foss (Foss, 1992) defines six distinct functions of narrative expression, including: realistic, dramatic, thematic, lyrical, comic, and extraneous. In this study, we are mainly interested in the thematic and dramatic functions.

4.3.1 The Thematic Function

A narrative function is said to be thematic when it acts “as a *comment* on or *interpretation* of what happens on the Plane of Events” (Foss, 1992). In the case of educational videos, we interpret the thematic function as being *instructional* and *informative*. It reflects por-

tions of the video where the filmmaker decides to ‘step in’ and interfere in the subject being shown. Voice of authority over raw footage in documentary videos, for example, would help the video makers to clarify the visual content, and perhaps to make an appeal for his/her subjective point of view. Superimposed text in training videos would draw trainees to the specifics and emphasise key messages. Extraction of such a function that can disclose the degree of the video-makers’ involvement, or mediation level in a sense, would be useful for content management, especially in the educational domain. This will allow us to segment sections with high-level informative/instructional contents, and facilitate queries such as “finding me segments with specific instructions”.

4.3.2 The Dramatic Function

If the thematic function is designed to capture the degree of the video-makers’ mediation, the dramatic function reflects the ‘interesting’ or the dramatic nature of presentation in the mediating process. Foss (Foss, 1992) sees the dramatic function as that which “influences human relations as well as people’s wishes, opinions, and choices”, and the lyrical function as that which “create[s] a particular atmosphere or feeling”. In the domain of educational videos, we shall combine these two and call them the *dramatic function*. So what do we expect this function to tell us? Consider the example about the dramatization of ladder safety in the training video, “Maintenance”. Here, the filmmaker chooses to not only talk about safety rules, but also decides to *dramatise* what is said by showing a sequence of images. Narration is stopped, ambient music is played in the background and the falling actions shown in the scenes. The dramatic function should reach a climax for this sequence.

Based on these definitions and the examination of several educational videos, we hypothesise that the key elements that are manipulated to influence thematic and dramatic functions of the content are: Use of colour, degree of motion shown, use of superimposed text, soundtrack (e.g., augmented with music, voice-over narration) and the appearance of the narrator(s). Table-(4.3) shows these elements and the impact on these two categories of functions.

4.3.3 Relating Media Elements to Thematic and Dramatic Functions

Given these media elements and the hypothesised impact in Table-(4.3), how do we justify which element should be accounted for in determining the functions of the content? For

example, should music be counted as a variable in computing a measure of thematic function for that shot? While we do not have the groundtruth from the literature to justify what should be candidates to highlight ‘good’ thematic/dramatic functions, we resort to what we call an *extreme-case* methodology.

Given an educational video, we watch and manually label segments which, in our opinion and according to the definitions, are deemed to be *extreme* thematic or dramatic sections. A section with scrolling text and voice-overs, for example, is a form of extreme thematic. By extreme, we mean in a way “undoubtedly”. A fast editing segment augmented by music, contains no narration, as observed in the beginning of some educational films is an example of extreme dramatic. We collect a set S of ten videos for this investigation. Shot indices for each video are first generated. Extreme segments are manually identified at the shot level, i.e., each segment consists of a sequence of shots. We denote $T(S)$ and $D(S)$ be the corresponding sets for thematic and dramatic segments respectively. Next, key media elements proposed in Table-(4.3) are formulated into a set of features as shown in Table-(4.4).

Media elements	Associated features
Colour	Hue, Lightness, Saturation, Colour Warmth/Cold (this feature set is detailed in (Truong <i>et al.</i> , 2002b)).
Motion	Average shot motion estimated as the average of camera pan and tilt (Srinivasan <i>et al.</i> , 1997).
Superimposed Texts	Text-Content-Ratio, measured as the ratio of number of frames containing text to the total number of frames in the shot (discussed in Section 3.3).
Narrators	Face-Content-Ratio, measured as the ratio of number of frames with face detected to the total number of frames in the shot. (discussed in Section 3.3).
Audio Track	Music-Ratio, Speech-Ratio, Silence-Ratio, and NL-Ratio. These features are discussed in Section 3.3.

Table 4.4: Associated features for media elements in Table-(4.3).

Figure-(4-6) illustrates the proposed media elements and their impact on the thematic and dramatic functions. The bar-graphs are plotted with average values of these features computed over the data set $T(S)$ and $D(S)$. The line-graphs show some examples of how these features vary across dramatic and thematic segments.

With respect to the sound track (Figure-4-6a), it is evident that the music-ratio feature (MU) dominates dramatic segments, while the speech-ratio feature (SP) dominates thematic segments. Although there are significant differences in the silence- and NL-ratio features between thematic and dramatic segments, we do not consider them as key influencing factors as their values all fall under 30%. In Figure-(4-6b), we observe the dominance of

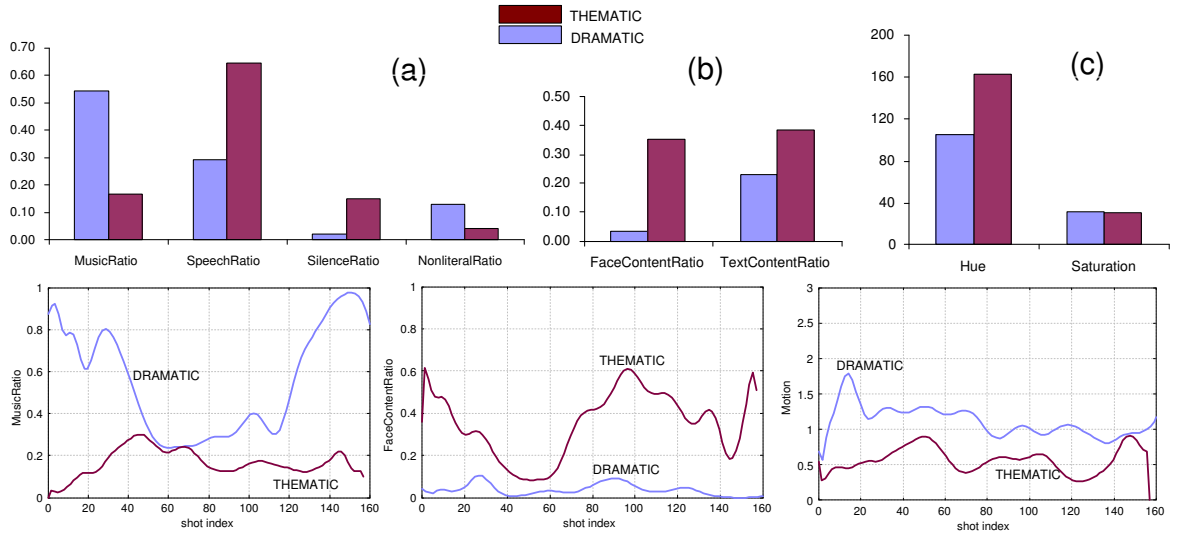


Figure 4-6: Establishing relationships between proposed media elements and their impact on thematic and dramatic nature of the educational content.

both face-content-ratio (FCR) and text-content-ratio (TCR) features for thematic segments. In studying the contribution of colour, Figure-(4-6c) shows no significant difference of the Hue and Saturation values between thematic and dramatic sections (note that the vertical axis here denotes the average values of Hue and Saturation, not the ratios), and thus does not strongly support our hypothesised impact of colour as indicated in Table-(4.3). We, therefore, decide to omit them from the list of primary contributing factors. We also observe the dominance of motion in dramatic sections, but not in thematic sections as shown inFigure-(4-6).

Our results, from analyzing the complete data set, shows that thematic segments are influenced primarily by the following features: speech-ratio (SP), text-content-ratio (TCR) and face-content-ratio (FCR). Dramatic segments are influenced primarily by: music-ratio (MU) and shot motion (MO). Based on these results, we formulate a thematic function as a weighted linear combination of its key contributing factors:

$$\mathbf{Th}[n] = \alpha \mathbb{N}\{SP[n]\} + \beta \mathbb{N}\{TCR[n]\} + \gamma \mathbb{N}\{FCR[n]\} \quad (4.7)$$

and likewise for the dramatic function:

$$\mathbf{Dr}[n] = \kappa \mathbb{N}\{MU[n]\} + \lambda \mathbb{N}\{MO[n]\} \quad (4.8)$$

where n is the shot index; $\mathbb{N}\{\cdot\}$ is the normalization operator defined as: $\mathbb{N}\{x\} = \frac{x - \mu_x}{\sigma_x}$; and α, β, γ and κ, λ are the weighting factors, which are assigned to 1 when no further knowledge is available (ie: each element contributes equally to the expressive functions). These functions are then smoothed with a Gaussian filter.

4.3.4 Experimental Results

In this experiment, we aim to study whether hand-labeled *extreme* segments do correspond to the *extrema* points (ie. maximum) of the thematic and dramatic functions. For a video, \mathbf{V} , to evaluate extreme *thematic* segments, we first compute the thematic function, and then detect all the peaks. A peak p whose thematic value ($\mathbf{Th}[p]$) exceeds a threshold, \mathbf{TH} is deemed to be an extreme peak and denoted as p^* . Let μ_T be the mean of thematic values for the video \mathbf{V} computed from the thematic function. In this study, we set the threshold to be $\mathbf{TH} = \mu_T + 20\% \times |\mu_T|$ (ie: placing a threshold of 20% more than average).

\mathbf{V}	THEMATIC					DRAMATIC				
	*	**	TP	FP	Ms	*	**	TP	FP	Miss
1.	10	6	5	1	0	8	2	1	1	0
2.	6	5	5	0	0	8	2	1	1	0
3.	7	6	4	2	1	7	1	1	0	0
4.	8	7	5	2	0	6	2	2	0	0
5.	12	8	4	4	0	10	2	0	2	0
6.	13	8	7	1	0	12	4	4	0	1
7.	18	10	8	2	0	21	3	3	0	1
8.	7	5	4	1	0	7	5	4	1	0
9.	8	7	5	2	1	9	2	2	0	0
10.	14	11	8	3	0	12	2	2	0	2
all	103	73	55	18	2	100	25	20	5	4
* : Number of detected maxima										
** : Number of detected maxima exceeding \mathbf{TH}										

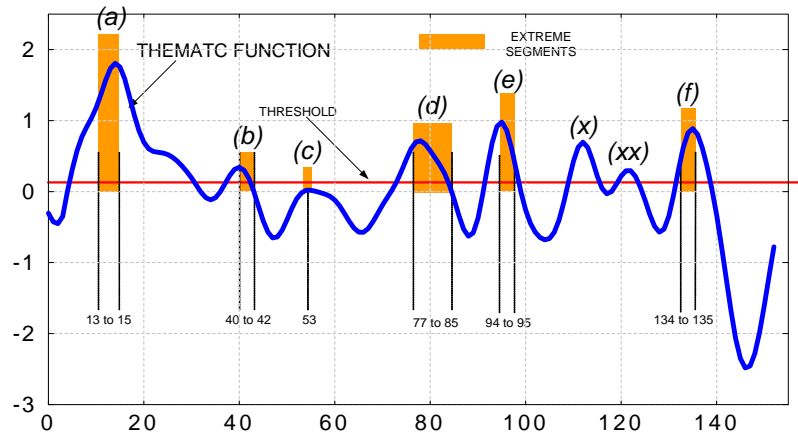
Table 4.5: Detection results of extrema points for dramatic and thematic functions.

A true positive is recorded if p^* belongs to one of groundtruth segments, and a false positive is counted otherwise. A miss is marked when a groundtruth segment does not correspond to any of the peaks that exceed \mathbf{TH} . We evaluate the *dramatic* content segments in a similar way. The results for this study are shown in the Table-(4.5). The recall and precision for dramatic segments are 96.5% and 83.3% respectively. For thematic segments these figures are 75.3% and 80%. An analysis of these results reveals that:

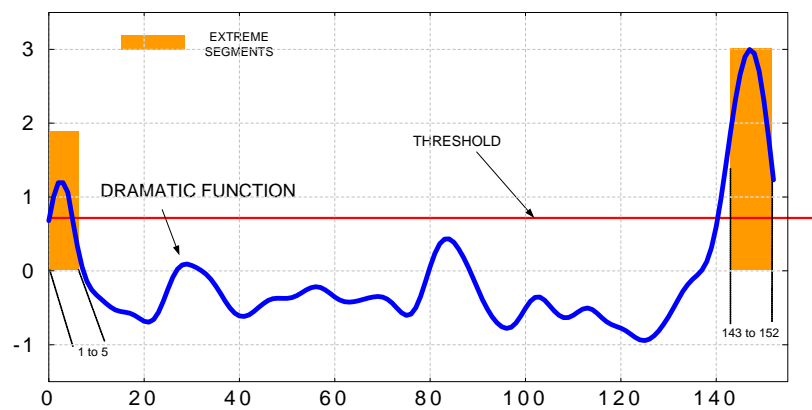
- False positives generally correspond to very weak peaks when compared to peaks identified as true positives.
- Misses are mostly attributed to errors in text/face detection results.

Figure-(4-7) shows an example of the dramatic and thematic functions computed for video

number 9 ('Electronics-Safety'). Segments labeled as 'extreme' are also shown. In Figure-(4-7(b)), we observe two strong peaks for the dramatic content at the beginning and the end of the video, corresponding to extreme dramatic sections as indicated in the groundtruth. The strongest peak at the end corresponds to the sections where the video makers summarise the video and 'dramatise' what was said. Music is played, narration is stopped, and a sequence of dissolves with lots of actions in the scenes are shown.



(a)



(b)

Figure 4-7: Analysis of thematic and dramatic functions for video 'Electronics-Safety'.

In Figure-(4-7(a)), we observe strong peaks at (a),(d),(e) and (f). These peaks all correspond to manually labeled extreme sections. The segment for shots 40 to 42 (peak b) also exceeds the threshold, though, it is a weak peak. However, there is one shot (53, peak c) that does not exceed the threshold and we also observe two cases of false positives (peaks

(x) and (xx)).

4.4 Hierarchically Partitioning Videos into Topics

Guided by Hypothesis A which indicates that the changes in mediation level are coincident with main topic changes, we now formulate an algorithm to detect these mediation change points. The level of mediation is encoded in the thematic and dramatic functions. Thus, detecting topics should reduce to detecting changes in the thematic function. However, instead of doing this directly, we first detect subtopics based on the content density function as outlined in Section 4.2. We then compute the changes in the thematic function and consider each subtopic transition point to be a candidate for a main topic transition point. Changes in the thematic function can be viewed as edges and can be detected using an edge detector such as Deriche’s recursive filtering algorithm using Gaussian kernels (Deriche, 1992). This is a multi-scale edge detection algorithm parameterized by Σ , which determines the slope of the target edges. Smaller slopes (more gradual) will be detected with larger Σ and vice versa. The edges are then selected based on a threshold τ . Larger τ will result in fewer but larger edges detected and vice versa.

In the pre-processing stage, the algorithm computes relevant primitive features followed by extraction of the content density and thematic functions. Detection of subtopic boundaries then follows as outlined in Algorithm 4.1. Let $\{s_1, s_2, \dots, s_m\}$ be the set of detected subtopic indices. At the next stage, each s_i is examined in relation to the thematic function to decide whether it can be refined as a main topic transition. Let \mathcal{E} be the set of edges detected from the thematic function $\mathbf{Th}(\cdot)$. A main topic boundary is deemed to exist at s_i if $\exists x, e_x \in \mathcal{E}$ such that $s_i \subset e_x$. The complete algorithm is formulated in Algorithm 4.2.

Note that a subtopic transition s_i is represented by a shot number being the beginning shot of the subtopic, while each edge e_x detected from \mathbf{Th} spans over a small number of consecutive shots. Therefore, in step **S4**, $s_i \subset e_x$ means s_i is one of the shots in e_x (see Figure-(4-9)).

4.4.1 Experimental Results

In this experiment, we test the performance of the detection scheme on a set of ten industrial safety films, which possess the hierarchical structure of main topics and subtopics. Each video is manually labeled to locate subtopic and main topic boundaries as the

Algorithm 4.2 Two-tiered hierarchical topical segmentation

Input: a video \mathbf{V} along with extracted primitive features, including: shot indices, motion, face detection results, text detection results, categorization of audio into speech, music, silence and non-literal (as shown in Section 3.3.2 in Chapter 3).

S1 [*expressive functions*]. Extract the content density \mathbf{Dn} and the thematic \mathbf{Th} functions (Equations 4.1 and 4.7), then smooth them with a Gaussian filter.

S2 [*subtopic level*]. Let $\mathcal{T} = \{s_i\}$: the set of topic indices detected at the subtopic level using a probabilistic approach (Section 4.2.4).

S3 [*edge detection*]. Let $\mathcal{E} = \{e_x\}$: the set of edges detected from \mathbf{Th} signal (Equation-(4.7)) using the edge detector detailed in (Deriche, 1992) with a pre-defined set of (Σ, τ) .

S4 [*main topic level*]. A subtopic index $s_i \in \mathcal{T}$ is deemed to be a main topic index if $\exists x, e_x \in \mathcal{E}$ such as $s_i \subset e_x$.

Output: A set of main topic and subtopic indices.

groundtruth. Most of these indices are derived from the manuals accompanying the films. Figure-(4-8) gives an example of the groundtruth for the video ‘Eye-Safety’. The solid bold line is the timeline of the film with the numbers in circles representing shot numbers corresponding to topic indices in the groundtruth. The top layer corresponds to main topic indices and the lower layer indicates subtopic indices.

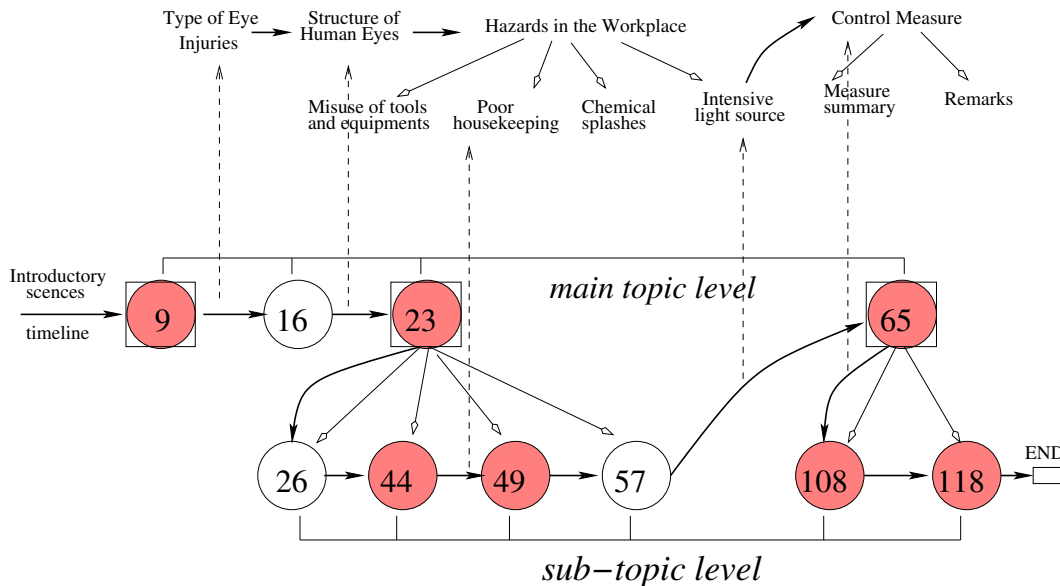


Figure 4-8: Groundtruth and detection results for the ‘Eye-Safety’ video.

First, we carry out the pre-processing stage to extract relevant primitive features across the data set. Then, the proposed algorithm is used to detect indices at subtopic and main

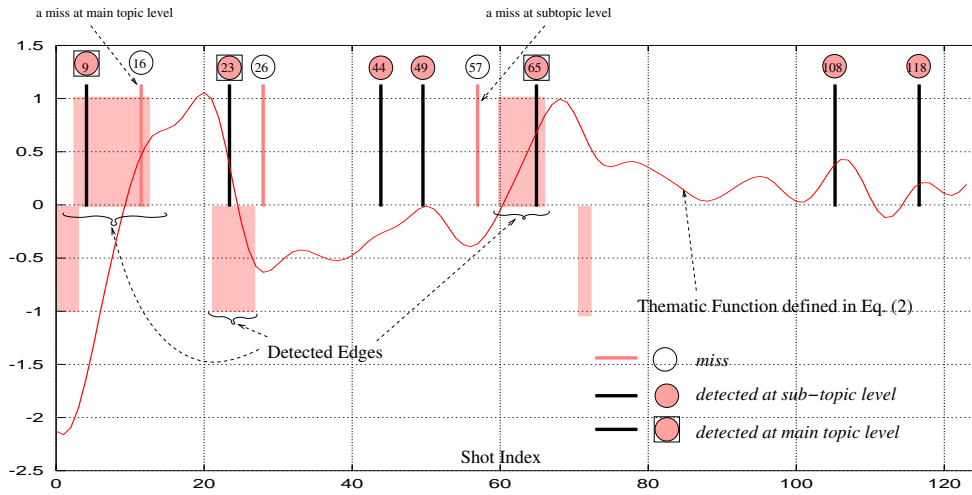


Figure 4-9: Plots of the thematic function, detected edges, topic and subtopic boundaries for instructional video ‘Eye-Safety’.

topic levels for each video. For example, in Figure-(4-8), a shaded circle indicates that a subtopic boundary is correctly detected by the algorithm. At the main topic level, a circle coupled with a square means that it is correctly detected by the algorithm. In this example, 7 out of 10 subtopic boundaries have been accurately detected at the subtopic level; and 3 out of 4 main topic boundaries are correctly detected. It is obvious from the diagram that the miss in the main topic boundary detection suffers from the miss at the subtopic level. Detection results for the whole data set are reported in Table-(4.6).

V	<i>subtopic level</i>				<i>main topic level</i>			
	GT	DT	<i>miss</i>	<i>false</i>	GT	DT	<i>miss</i>	<i>false</i>
1.	10	7	3	0	4	3	1	0
2.	9	6	3	1	5	3	2	0
3.	8	5	3	1	0	0	0	0
4.	5	5	0	2	3	2	1	1
5.	9	8	1	0	4	2	2	0
6.	13	10	3	1	4	3	1	0
7.	7	7	0	2	2	2	0	0
8.	6	6	0	1	3	3	0	1
9.	19	16	3	1	5	4	1	1
10.	12	11	1	1	3	3	0	1
All	98	81	17	10	33	25	8	4
GT : groundtruth, DT : detection <i>miss</i> : the algorithm misses groundtruth boundaries <i>false</i> : the algorithm wrongly claims boundaries								

Table 4.6: Detection results for a set of ten education and training videos.

The experimental results report an overall recall of 82.7% and a precision of 89.0% at the

subtopic level across the video set. At the main topic level, these figures are 75.8% and 86.2% respectively. Figure-(4-9) partially illustrates the analysis for video ‘Eye-Safety’, whose corresponding hierarchical topic structure is depicted in Figure-(4-8).

Nature of errors at the main topic level

Errors at the main topic level arise due to two reasons: (i) detection errors at the subtopic level, and (ii) errors resulting from our hypothesis, which is strongly tied to the assumption that main topic boundaries coincide with significant edges of the thematic function. Among the two, the former attributes for most of the *misses*, and thus accounts for the major mode of failure in the overall scheme (8 out of 33). Errors from the latter source are due to the ‘ambiguity’ in defining ‘how much is enough’ to claim noticeable changes in the mediation process, which is reflected in our thematic function. The fact is that not all main topic boundaries lie on strong edges of the thematic function. Therefore, the set of parameters (Σ, τ) for the edge detection algorithm has to be relaxed and the optimal parameters are achieved empirically (1 and 0.8 in this experiment).

4.5 Concluding Remarks

This chapter has addressed the problem of automatically segmenting instructional videos into high-level topical sections. From an analysis of a collection of training videos, we proposed a content density function and examined how subtopic changes relate to its behaviour over the duration of a video. We have presented two approaches to subtopic boundary detection. One is carried out in a deterministic manner based on heuristics and the other is based on probabilistic measures. The experimental results on several videos have shown the feasibility of our algorithms.

Next, we have defined and studied the thematic and dramatic functions of the content in educational videos. We first hypothesise key media elements, followed by experiments on ten videos to study primary contributing factors to these constructs. We then define two novel measures to indicate the thematic and dramatic nature of the content portrayed. Finally, we describe experiments to evaluate the performance of these functions. The results have shown the validity and the usefulness of these measures.

Lastly, we have proposed an algorithmic solution for segmenting an instructional video into hierarchical topical sections. Incorporating the knowledge of education-oriented film theory with our previous study of expressive functions, namely the content density and the thematic functions, we develop algorithms to effectively structuralise an instructional video into a two-tiered hierarchy of topical sections, at the main and subtopic levels. Our

experimental results on a set of ten industrial instructional videos have shown a 75.8% of recall and 86.2% of precision for detecting main topic boundaries. While the results are humble, it demonstrates the usefulness of devised media expressive functions in our framework for video segmentation.

Chapter 5

Hierarchical Hidden Markov Models with Shared Structures

Chapters 3 and 4 contain an exposition of our Film Grammar based approach to the problem of video analysis, in which the educational domain is used as the theme of investigation. At the core of this approach is the Computational Media Aesthetics framework which guides us through the identification of meaningful structural units, the extraction of expressive functions and the segmentation of the video into topics and subtopics. The domain-specific knowledge consisting of rules and conventions has served as the role of the knowledge-base in an expert-like system. In this and the next chapters, we present a different approach, set in a probabilistic framework, towards the problem of understanding video content. The benefit of probabilistic models has been evident for decades, in particular the successful spread of the Hidden Markov Model (HMM) in many areas. In the field of video content understanding, as reviewed in Chapter 2, the HMMS have been used extensively with varying degrees of success. An interesting direction is the use of HMMS to provide *semantic descriptions at multiple levels*. However, all previous works have modeled each semantic level separately and combined them in a rather heuristic manner, especially across boundaries at higher layers of semantics. The problems with these approaches are that the underlying framework does not capture the essential nature of interaction across semantic layers, and these interactions are not part of the learning process. Furthermore, the training data requires manual groundtruth at different levels of segmentation, which could be very time-consuming and subjective.

One natural framework that integrates the semantics at multiple resolutions is the *Hierarchical Hidden Markov Model* (HHMM) introduced lately in (Fine *et al.*, 1998). Although still at an early stage, this model has found its application in the area of video analysis (Xie *et al.*, 2002a; Xie and Chang, 2003). However, in all of these works the hierarchic information in the HHMM has not been kept explicitly, and thus the information about

the structure is not effectively exploited for segmentation or classification. In addition, despite the modeling attractiveness of the HHMM, its theoretical foundation still needs exploration. Most importantly, the strict tree-form structure limits its expressiveness and hinders its applications. In this work, we attempt to solve some of these problems both *theoretically* (in this chapter) and *practically* in (the next chapter).

From the theoretical aspect, we argue that being able to *represent and model the shared structures hierarchically* is the key to labeling and segmenting in the video domain. Our key observation is that, in all video domains, there is a common and natural hierarchical mapping to the semantic of the content: a film has episodes, story units, scenes then shots; a teaching video has intentional messages within subtopics which in turn are embedded in broader topics; a tracking video can be mapped into different resolutions of activities, and so on. But more importantly, *semantic structures are naturally shared and inherited* in the hierarchy. To this end, we introduce *a general Hierarchical Hidden Markov Model in which the state hierarchy can be a lattice allowing arbitrary sharing of substructures*. Our main contributions in this chapter are as follows:

- Based on the original HHMM proposed in (Fine *et al.*, 1998), we generalise this model to allow *shared substructures* in the topological specification, which results in a novel theoretical extension to the HHMM. The resulting model allows a HHMM to have the most generalised form of the state hierarchy. The direct modeling of shared structures results in practical savings in computation, and more accurate parameter learning with less training data. In addition, unproven technical details in the original work (Fine *et al.*, 1998) are also formally verified.
- A set of conditional independence properties formally exploited in the Dynamic Bayesian Network representation of the HHMM.
- A novel Asymmetric Inside-Outside algorithm to do inference in the presence of shared structures and a contribution of an EM parameter estimation procedure for the problem of learning in the HHMM.
- To deploy this model in real-world problems, we present a novel scaling algorithm for the HHMM to avoid numerical underflow, an issue that was ignored in the original paper (Fine *et al.*, 1998).
- To simplify the calculation as well as to enhance the understanding of computational details, we devise a set of diagrammatic tools to intuitively visualise the structure of complex computation during the inference process. Although formal derivation of several recursive algorithms appear very complicated, the tools simplify the deriving process enormously.

The rest of the chapter is organised as follows. We formulate the issue of shared structures and provide formal definitions for the HHMM in Section 5.1, pointing out the current limitations. Next, the Dynamic Bayesian Network representation for the HHMM is presented in Section 5.2 where a set of conditional independencies is formally exploited. We then compute the set of the sufficient statistics and a solution for Maximum Likelihood (ML) parameter estimation when the model is fully observed in Section 5.3. Extension to the presence of latent variables and the EM algorithm for the HHMM is addressed in Section 5.4, followed by a discussion of the inference algorithms in Section 5.5. Complexity analysis and some numerical results are then given in Section 5.7. In Section 5.8, we provide a novel scaling algorithm for the HHMM to avoid numerical underflows when dealing with long observation sequences. Finally, the conclusion is provided in Section 5.9. Supplementary information for this chapter, including several proofs for unproven equations and theorems, is further provided in the Appendix A, Appendix B, and Appendix C.

5.1 The Hierarchical HMM: Intuition and Definition

In this section, our goal is to revise the FST model (Fine *et al.*, 1998), pointing out its weaknesses and limitations. In addition, we provide a formal definition for the HHMM where the idea of shared states is introduced and formalised. Further, we familiarise readers with some important notations used in later developments. First, an intuition into the HHMM is provided in Section 5.1.1. Next, the HHMM is formally defined in Section 5.1.2, followed by the introduction of a HHMM with a general state hierarchy in Section 5.1.3.

5.1.1 From the Regular HMM to the Hierarchical HMM

Let us first provide some intuition. A regular discrete HMM is a finite state automata parameterised by $\theta_{\text{HMM}} \triangleq \{\pi, A, B\}$, where π is the initial probability vector, A is the transition matrix and B is the emission probability matrix. Figure 5-1(a) shows a simple finite state automata for a HMM with three states. To convey the idea of hierarchy, we introduce the notion of *topological structure*, ζ_{HMM} . In the case of a regular HMM, the topology is simply a set of states. For convenience, we assume that there is a dummy universal state 1 (root state) that gives birth to all other states. The topology for the HMM for this example is shown in Figure 5-1(b).

The hierarchical HMMs introduced in (Fine *et al.*, 1998) extends the traditional HMM in a hierarchic manner to include a hierarchy of hidden states. Each state in the normal

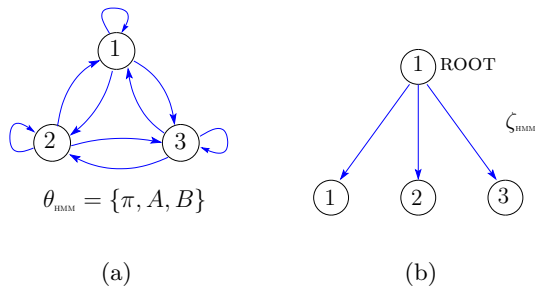


Figure 5-1: Representing a HMM with three states as a finite state automata (a), and its corresponding topological structure (b).

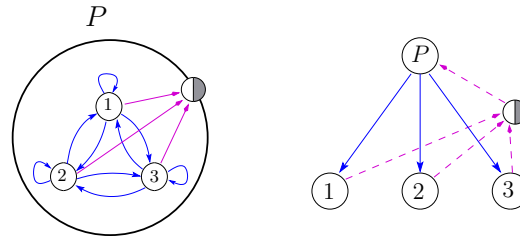


Figure 5-2: The HMM in Figure-5-1) can be considered as children of a higher abstract state P . The dashed lines and end-state in the topological diagram are usually not shown in the rest of the thesis for readability.

HMM is generalised *recursively* as another sub-HMM with special *end* states included to signal when the control of the activation is returned to the parent HMM. Further details of the work (Fine *et al.*, 1998) has been discussed in the background in Subsection 2.3.3.5 (page 44). Note that the end-state should be thought of as an indicator state rather than as a normal state in finite state automata. It cannot be initialised from the parent, does not transit to other states in the same level and when reached, does not emit any observations.

Example 5.1 Consider states 1, 2, 3 in Figure 5-1(b) to be the *children* of a higher abstract state P . To illustrate the idea, consider an example when the introduction of a subtopic is filmed (abstract state P), the video-maker decides to first take a shot of the narrator (child-state 1) speaking directly to the viewers, then shows some text captions of the main points (child-state 2) in a few shots, occasionally inserting a shot of the narrator explaining the points (child-state 3). The *parent* state P governs and encapsulates the Markovian process of its children $\text{ch}(P) = \{1, 2, 3\}$. Figure 5-2 shows the new form of the state transitions and the topology¹. \square

¹Here, the transition diagram is added with a special end-state. For the parent P state, the initial probability vector π and the transition matrix A for its children now has the form:

$$\pi^P = \begin{bmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \end{bmatrix}, \quad A^P = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{1,\text{end}} \\ a_{21} & a_{22} & a_{23} & a_{2,\text{end}} \\ a_{31} & a_{32} & a_{33} & a_{3,\text{end}} \end{bmatrix}$$

By recursively extending a state to be another HHMM, a hierarchy of Hidden Markov Models is defined.

Example 5.2 Following Example-(5.1), assume that there is a second abstract state Q , for example, to represent a film style corresponding to the body of a subtopic, in which text (child-state 4) and illustrative scenes (child-state 5) are shown alternatively. Markovian processes at the parent- and children- levels define a HHMM. Its state transition diagram and corresponding topology are shown in Figure-(5-3). \square

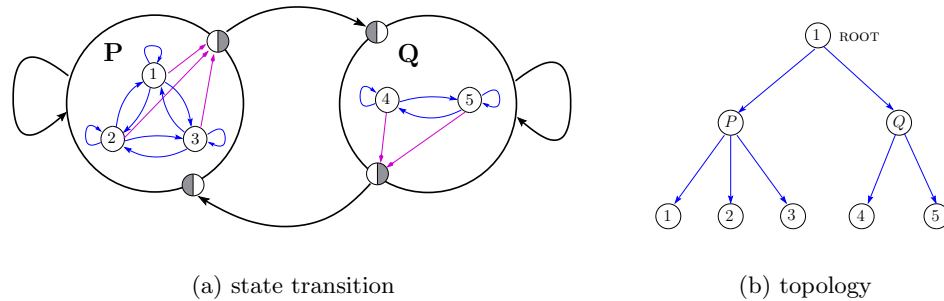


Figure 5-3: State transition and corresponding topology for the HHMM described in Example-(5.1) and Example-(5.2).

5.1.2 Hierarchical HMMs: Model Definition

A discrete hierarchical HMM is formally defined by a 3-tuple $\langle \zeta, \theta, \mathcal{Y} \rangle$: a topological structure ζ , a set of parameter θ , and an emission alphabet space \mathcal{Y} . This definition is general and applicable for the case when shared structures exist.

Topological Specification ζ

The topology ζ specifies the model depth D , and for $1 \leq d \leq D$, a set of states \mathcal{S}^d at level d is defined. For convenience, we will number the states sequentially and denote this set as:

$$\mathcal{S}^d = \{1 \dots |\mathcal{S}^d|\} \quad \text{where } |\mathcal{S}^d| \text{ is the number of states at level } d$$

Level 1 is the root level and consists of a single state, $\mathcal{S}^1 = \{1\}$, while level D is the bottom level and sometimes referred to as the *production* level – the only level permitted to emit observations and at this level the state does not have any children.

For each state $x^d \in \mathcal{S}^d, d < D$, the topological structure also specifies the set of its children, where π and A are now “attached” to P and an extra column (going to end) is added for A.

denoted as $\text{ch}(x^d) \subset \mathcal{S}^{d+1}$. From this, the set of parents of x^d can be derived and denoted as $\text{pa}(x^d)$.

In addition to the specification of the states and their parent-children relationship, we also specify an observation model. In the discrete case, we denote \mathcal{Y} to be the set of alphabets.

Example 5.3 By numbering sequentially, states P and Q in Example-(5.2) will be numbered as 1, 2. The topological structure ζ is then specified as:

$$\begin{aligned} \text{Model depth } D = 3, \quad \mathcal{S}^1 &= \{1\}, \quad \text{ch}(x^1 = 1) = \{1, 2\} \\ \mathcal{S}^2 &= \{1, 2\}, \quad \text{ch}(x^2 = 1) = \{1, 2, 3\}, \quad \text{ch}(x^2 = 2) = \{4, 5\} \\ \mathcal{S}^3 &= \{1, 2, 3, 4, 5\} \end{aligned}$$

This topology is clearly a tree. □

Parameter set θ and the observation space \mathcal{Y}

Given a topological structure ζ , the parameter set θ for the HHMM is specified as follows.

For each level $d \in \{1 \dots D - 1\}$, $p \in \mathcal{S}^d$, and $i, j \in \text{ch}(p)$:

$\pi_i^{d,p} \in [0, 1]$	is the initial probability of child i , activated given the its parent p .
$A_{i,j}^{d,p} \in [0, 1]$	is the transition probability from child i to j given both are the children of p .
$A_{i,\text{end}}^{d,p} \in [0, 1]$	is the transition probability of child i going to end-state given the parent is p .

Table 5.1: Parameters definition for a HHMM at level d for $1 \leq d \leq D - 1$.

Finally, for each state at the production level $i \in \mathcal{S}^D$, and the alphabet $v \in \mathcal{Y}$, we specify the emission probability $B_{v|i} \in [0, 1]$ to be the probability of observing v given that the current state at the production level is i . The whole parameter set is written as: $\theta = \{\pi, A, A_{[\text{end}]}, B\}$, where:

$$\begin{aligned} \pi &= \bigcup_{\substack{1 \leq d \leq D-1 \\ p \in \mathcal{S}^d}} \{\pi^{d,p} : 1 \times M\} & B &: |\mathcal{S}^D| \times |\mathcal{Y}| \\ A &= \bigcup_{\substack{1 \leq d \leq D-1 \\ p \in \mathcal{S}^d}} \{A^{d,p} : M \times M\} & A_{[\text{end}]} &= \bigcup_{\substack{1 \leq d \leq D-1 \\ p \in \mathcal{S}^d}} \{A_{[\text{end}]}^{d,p} : 1 \times M\} \end{aligned}$$

where in each case $M = |\text{ch}(p)|$ is number of children for p . Clearly, the number of parameters required for an abstract state p of M_p children is: $M_p + M_p(M_p + 1)$, and at production level $|\mathcal{S}^D| \times |\mathcal{Y}|$ parameters is required in the discrete case. Thus, the total

number of parameters for a HHMM is:

$$\text{NUMBER OF PARAMETERS} = \sum_{1 \leq d \leq D-1} \sum_{p \in \mathcal{S}^d} M_p(M_p + 2) + |\mathcal{S}^D| \times |\mathcal{Y}| \quad (5.1)$$

Clearly, when $D = 2$, the number of parameters is the same as in the flat HMM case.

Constraints

The condition of stochastic processes require the following constraints:

$$\sum_{i \in \text{ch}(p)} \pi_i^{d,p} = 1, \quad \forall d, p, i : \quad 1 \leq d \leq D-1, p \in \mathcal{S}^d, i \in \text{ch}(p) \quad (5.2a)$$

$$\sum_{v \in \mathcal{Y}} B_{v|i} = 1, \quad \forall v \in \mathcal{Y}, i \in \mathcal{S}^D \quad (5.2b)$$

$$\sum_{j \in \text{ch}(p)} A_{i,j}^{d,p} + A_{i,\text{end}}^{d,p} = 1, \quad \forall d, p, i, j : \quad 1 \leq d \leq D-1, p \in \mathcal{S}^d, \{i, j\} \in \text{ch}(p) \quad (5.2c)$$

5.1.3 Hierarchical HMMs with Shared Substructures

The original HHMM in (Fine *et al.*, 1998) requires that *each state has strictly only one parent* and, thus, $\text{pa}(x^d)$ always contains a *single* state. This is equivalent to the *disjoint* condition on the children set:

$$\boxed{\text{For } 1 \leq d \leq D, \text{ and } \forall p \neq q \in \mathcal{S}^d : \quad \text{ch}(p) \cap \text{ch}(q) = \{\emptyset\}} \quad (5.3)$$

The disjoint condition in Equation-(5.3) necessarily limits the topological structure in the original HHMM to *be strictly a tree*, and limits the expressiveness of the FST model by disallowing shared sub-states at the same level.

We lift that restriction here and *allow a state to be shared* by an arbitrary set of parent-states from the upper level. When this restriction is lifted, the topological structure ζ is, in general, a lattice which results in a HHMM with a general state hierarchy. This significantly reduces the model size, resulting in practical savings in computation, and requires much less training data to learn the shared structures.

Example 5.4 Continuing with Example-(5.1) and Example-(5.2), we see that child 2 of P and child 4 of Q carry the same semantic functionality, that is, the segment in the video where captioned texts are displayed. In the new setting, these two states can be modeled as a single one, denoted by $*$, which is shared by both P and Q . The topological structure

is now specified as:

$$\begin{aligned}
 \text{Model depth } D = 3, \quad \mathcal{S}^1 &= \{1\}, \quad \text{ch}(x^1 = 1) = \{1, 2\} \\
 \mathcal{S}^2 &= \{1, 2\}, \quad \text{ch}(x^2 = 1) = \{1, 3, *\}, \quad \text{ch}(x^2 = 2) = \{*, 5\} \\
 \mathcal{S}^3 &= \{1, 3, 5, *\}
 \end{aligned}$$

Figure-(5-4) shows the new transition diagram and the corresponding topological structure. ⊠

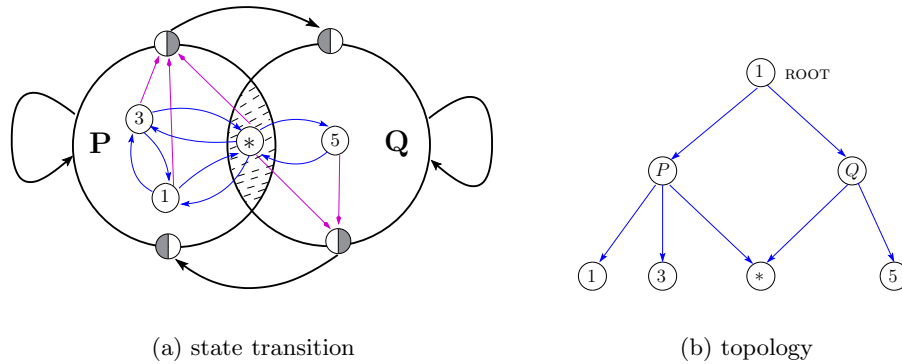


Figure 5-4: State transition and corresponding topology for the HHMM described in Example-(5.4) (inner self-transition arrows are not shown). State * is shared by both *P* and *Q*.

Figure-(5-5) shows how a lattice topology structure can greatly reduce the model size for a fully connected four-level HHMM when compared to its equivalent expanded HHMM when no shared structures are allowed.

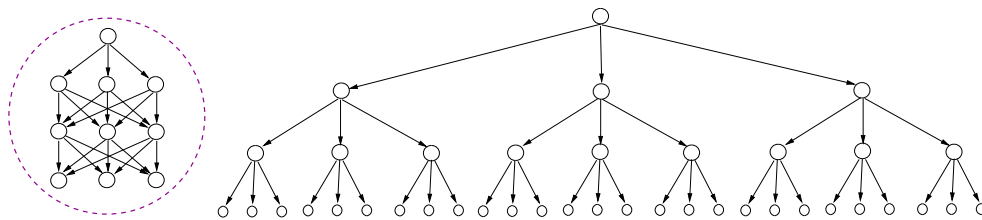


Figure 5-5: A fully shared four-level HHMM (lattice) topology, and its equivalent expanded tree topology (when shared structure are not modeled).

5.2 DBN Representation for the Hierarchical HMM

In this section, we describe the Dynamic Bayesian Network (DBN) representation for the HHMM, outline the assumptions, and the mappings of the parameter set θ to the DBN structure. The network serves two main purposes: (1) as the tool to derive the probabilistic independence of this stochastic model, and (2) as a computational framework for the inference and learning algorithms developed throughout this chapter. While the idea of the DBN representation is not new, our contribution in this section is the exploitation of the set of conditional independencies in Section 5.2.3. These properties provide key tools to develop the inference algorithm in Section 5.5.

5.2.1 Network construction

The idea that the hierarchical decomposition of a stochastic dynamic process can be modeled in a Dynamic Bayesian Network (DBN) first appeared in (Bui *et al.*, 2000; Murphy and Paskin, 2001). Murphy and Paskin (2001) convert a HHMM into a DBN and apply a general DBN inference to the model. Given the parameter set θ , the HHMM defines a joint distribution over a set of variables that represents the evolution of the stochastic process over time.

At time t , let x_t^d represent the current state at level d ($d = 1, \dots, D$), and e_t^d represent its ending status, ie: a boolean variable indicating whether its child-state x_t^{d+1} has reached the end-state at the current time. In addition, we denote the observation by y_t . For a compact representation, we use hierarchic superscript $d^* : d$ to represent a sequence of variables over layers, eg: $x_t^{d^*:d} = (x_t^{d^*}, x_t^{d^*+1}, \dots, x_t^d)$ (for $d^* \leq d$). The set of all variables at time t , therefore, includes $\{x_t^{1:D}, e_t^{1:D}, y_t\}$, and we denote this set as \mathcal{V}_t .

5.2.1.1 Assumptions in the DBN Structure

The way the dynamic Bayesian network is constructed imposes some restrictions. First, by the definition of the model, the root is always fixed to be a unique value 1 and does not terminate during the entire execution of the network, therefore:

$$x_t^1 = 1 \quad \forall t = 1, \dots, T \quad (5.4a)$$

$$\text{and } e_t^1 = 0 \quad \forall t = 1, \dots, T - 1 \quad (5.4b)$$

Note that in the original HHMM setting (Fine *et al.*, 1998), the root is assumed to end at T , which means it further requires $e_T^1 = 1$. Here we lift this restriction and assume that this knowledge is unknown. In addition, a state at the production level x_t^D is required to make an immediate transition when reached, therefore:

$$e_t^D = 1 \quad \forall t = 1, \dots, T - 1 \quad (5.5)$$

Finally, in the generative process of a HHMM, a state can end only when the state below it has ended. This leads to the following conditions:

$$\forall t, \quad e_t^d = 1 \implies e_t^{d^*} = 1 \quad \text{for } d^* > d \quad (5.6a)$$

$$\text{and } \forall t, \quad e_t^d = 0 \implies e_t^{d^*} = 0 \quad \text{for } d^* < d \quad (5.6b)$$

5.2.1.2 Construction of the first slice

The first slice of the DBN network is constructed as follows. First, a chain of states is activated in a top-down manner. At the root level, $x_1^1 = 1$ (root), then x_1^{d+1} is recursively generated solely from its parent x_1^d from the distribution π^{d, x_1^d} . Finally, when it reaches the bottom level, the observation y_1 is generated from the state at the production level x_1^D , and the emission probability matrix B . The dependency is shown in Figure-(5-6(a)).

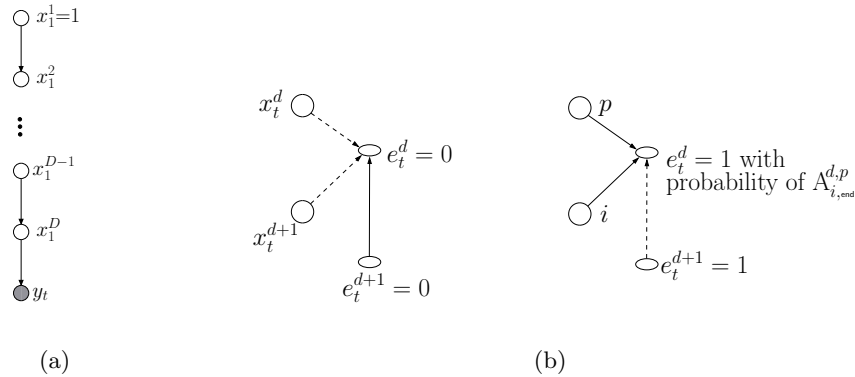


Figure 5-6: Network construction at $t = 1$: a) activation chain for the states, and b) sub-network structures for ending status e_t^d . Broken dependencies are shown in dashed arrows.

Next, a bottom-up chain is activated for the end-states in the following manner. A state at the lowest level always makes a transition, therefore: $e_1^D = 1$. Next, the value of

e_1^d is recursively determined based on e_1^{d+1} and $\{x_1^d, x_1^{d+1}\}$. This dependency is shown in Figure-(5-6(b)) (for $t = 1$) and can further broken down into two cases:

- (1) if $e_1^{d+1} = 0$, ie: the children of x_1^{d+1} have not reached the end-state, and thus it must still remain in execution, and therefore, $e_1^d = 0$.
- (2) otherwise, $e_1^{d+1} = 1$ implies that the children of x_1^{d+1} have reached end-state, and therefore, it terminates, ie: $e_t^d = 1$, with a probability of $A_{i,\text{end}}^{d,p}$ where $p = x_1^d$ and $i = x_1^{d+1}$.

Finally, when it reaches the top, e_1^1 is assigned to 0 by default.

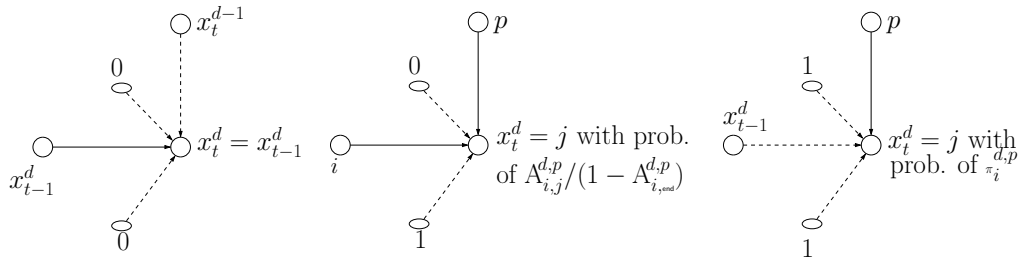


Figure 5-7: Subnetworks for the dependence structure during a transition. Broken dependencies are shown in dashed arrows.

5.2.1.3 Construction at time t

The structure of the network at time slice t is also constructed by a two-phase process: a top-down activation for x_t^d and a bottom-up activation for e_t^d . While the bottom-up phase for e_t^d behaves exactly in the same way as described above (Figure-(5-6(b))), the top-down activation for x_t^d is more complicated and we describe it now.

Initially, at the root level, x_t^1 is always assigned to 1. Next, the value of a state x_t^d is determined based on the set of its parental variables $\{x_t^{d-1}, x_{t-1}^d, e_{t-1}^{d-1}, e_{t-1}^d\}$. Their dependencies are shown in Figure-(5-7) and can be further simplified according to the values of the pair $(e_{t-1}^{d-1}, e_{t-1}^d)$:

- if $(e_{t-1}^{d-1}, e_{t-1}^d) = (0, 0)$, then x_{t-1}^d must still remain in execution, thus: $x_t^d = x_{t-1}^d$. The link from $x_{t-1}^d \rightarrow x_t^d$ can also be removed.
- if $(e_{t-1}^{d-1}, e_{t-1}^d) = (0, 1)$, then $x_t^d = j$ with a probability of $A_{i,j}^{d-1,p}/(1 - A_{i,\text{end}}^{d-1,p})$ (where $p = x_{t-1}^{d-1}$, and $i = x_{t-1}^d$)

- if $(e_{t-1}^{d-1}, e_{t-1}^d) = (1, 1)$, then $x_t^d = j$ with a probability of $\pi_j^{d,p}$. The link from $x_{t-1}^d \rightarrow x_t^d$ can also be removed.

and we note that $(e_{t-1}^{d-1}, e_{t-1}^d)$ cannot take on the values of $(1, 0)$. The whole network construction at time t can be summarised as follows. A state x_t^d can end only when the state below it x_t^{d+1} has ended, and it does so with a probability given by the termination parameter:

$$\Pr(e_t^d = 1 \mid x_t^{d+1} = i, x_t^d = p, e_t^{d+1}) = \begin{cases} A_{i,\text{end}}^{d,p} & \text{if } e_t^{d+1} = 1 \\ 0 & \text{if } e_t^{d+1} = 0 \end{cases} \quad (5.7)$$

A state stays the same until the next time slice if it does not end. If it ends, and the parent stays the same, it transits to a new child-state of the same parent; otherwise, it is initialised by a new parent state, ie:

$$\Pr(x_t^d = j \mid x_{t-1}^d = i, x_{t-1}^{d-1} = p, e_{t-1}^{d-1:d}) = \begin{cases} \delta(i, j) & \text{if } e_{t-1}^{d-1:d} = 00 \\ A_{i,j}^{d-1,p} / (1 - A_{i,\text{end}}^{d-1,p}) & \text{if } e_{t-1}^{d-1:d} = 01 \\ \pi_j^{d-1,p} & \text{if } e_{t-1}^{d-1:d} = 11 \end{cases} \quad (5.8)$$

where we use shortened notation $e_{t-1}^{d-1:d} = 01$ to mean $e_{t-1}^{d-1} = 0$, and $e_{t-1}^d = 1$.

5.2.1.4 The full DBN structure

The full Dynamic Bayesian Network for the HHMM can be constructed at $t = 1$ and subsequently for $t > 1$ as described above. The resulting DBN structure when unrolled T times is shown in Figure-(5-8). The DBN has D state-layers and one observation layer. In general, all the states are hidden, and only the observation is observed (shaded nodes).

In addition to hierarchic superscript, we use the time subscript $l : r$ to represent a sequence of variables over time, eg : $y_{l:r} \triangleq y_l, y_{l+1}, \dots, y_{r-1}, y_r$ ($l \leq r$). The whole set of variables for DBN can then be represented compactly as:

$$\mathcal{V} = \{\mathcal{V}_t\}_{t=1}^T = \{x_{1:T}^{1:D}, e_{1:T-1}^{1:D}, y_{1:T}\} \quad (5.9)$$

The HHMM defines a joint probability distribution (JPD) over \mathcal{V} following the factori-

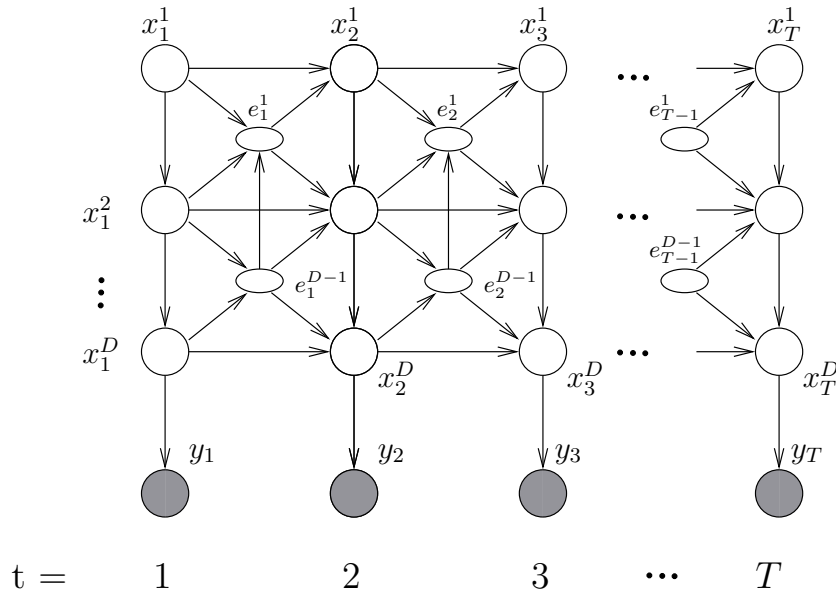


Figure 5-8: DBN representation for the HHMM, shaded nodes are observed variables. By definition, at the bottom level e_t^D is fixed to 1 and thus is removed from the network.

sation of the DBN²:

$$\Pr(\mathcal{V} \mid \theta) = \left(\prod_{t=1}^T \prod_{d=1}^{D-1} \psi_{(x_t^{d+1})} \right) \left(\prod_{t=1}^{T-1} \prod_{d=1}^{D-1} \psi_{(e_t^d)} \right) \left(\prod_{t=1}^T \psi_{(y_t)} \right) \quad (5.10)$$

where we use the notation $\psi_{(z)}$ to represent the conditional probability table (CPT) defined over z and its parents π_z , ie:

$$\psi_{(x_1^{d+1})} \triangleq \Pr(x_1^{d+1} \mid x_1^d) \quad \text{for } t = 1 \quad (5.11a)$$

$$\psi_{(x_t^{d+1})} \triangleq \Pr(x_t^{d+1} \mid x_t^d, x_{t-1}^{d+1}, e_{t-1}^d, e_{t-1}^{d+1}) \quad \text{for } t > 1 \quad (5.11b)$$

$$\psi_{(e_t^d)} \triangleq \Pr(e_t^d \mid e_t^{d+1}, x_{t-1}^d, x_{t-1}^{d+1}) \quad (5.11c)$$

$$\psi_{(y_t)} \triangleq \Pr(y_t \mid x_t^D) \quad (5.11d)$$

There are four types of ‘family’ $\{z, \pi_z\}$ are identified as shown in Figure-(5-9).

Alternatively, this JPD can be compactly written as:

$$\Pr(x_{1:T}^{1:D}, e_{1:T-1}^{1:D}, y_{1:T}) \triangleq \prod_{z \in \mathcal{V}} \Pr(z \mid \pi_z) \quad (5.12)$$

²We note that since the top-level state does not end during the HHMM process, the set of well-defined events is restricted to those instantiations of \mathcal{V} such that $e_{1:T-1}^D$ are false. An event that does not satisfy this property is not associated with any probability mass. We therefore ignore the top level, ie: $(x_{1:T}^D, e_{1:T}^D)$, in the JPD factorisation. In addition, since e_t^D is always fixed to 1, it is also ignored.

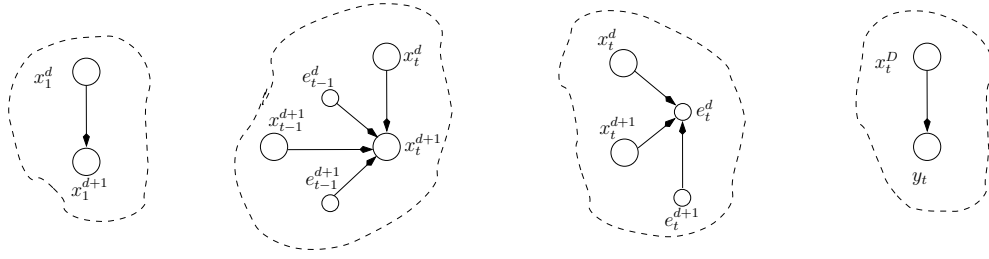


Figure 5-9: Four types of ‘family’ $\{z, \pi_z\}$ identified in the DBN of the HHMM.

where z represents a general node in the network structure being either x_t^d , e_t^d or y_t . The conditional probability $\Pr(z \mid \pi_z)$ is defined from the parameters of the HHMM (Section 5.2.2) and captures the evolution of the variables over time.

5.2.2 Mapping Parameters from the DBN Structure

Given that a HHMM can be represented as a DBN its parameter set θ , defined in Table-(5.1), is mapped into the following equivalent conditional probabilities:

$$\pi_i^{d,p} \triangleq \Pr(x_t^{d+1} = i \mid \cdot x_t^d = p) \quad (5.13a)$$

$$A_{i,j}^{d,p} \triangleq \Pr(x_{t+1}^{d+1} = j, e_t^d = 0 \mid x_t^{d+1} = i, x_t^d = p) \quad (5.13b)$$

$$A_{i,\text{end}}^{d,p} \triangleq \Pr(e_t^d = 1 \mid x_t^d = p, x_t^{d+1} = i) \quad (5.13c)$$

where we have utilised the notation $\{\cdot x_t^d = p\}$ to represent the event $\{x_t^d = p, e_{t-1}^d = 1\}$: the event that state p is activated at time t . Similarly $\{x_t^d = p\}$ represents the event $\{x_t^d = p, e_t^d = 1\}$ (Figure-(5-10)). Finally, the emission probability in the discrete case is defined as in the usual HMM setting:

$$B_{v|i} \triangleq \Pr(y_t = v \mid x_t^D = i) \quad (5.13d)$$

For the case in which the emission probability is continuous, we model the emission probability as a mixture of Gaussians and present the results in Chapter 6. Clearly, the set of probabilities for the parameters, as defined in Equation-(5.13a) and Equation-(5.13d), satisfy the stochastic constraints in Equation-(5.2a) and Equation-(5.2c).

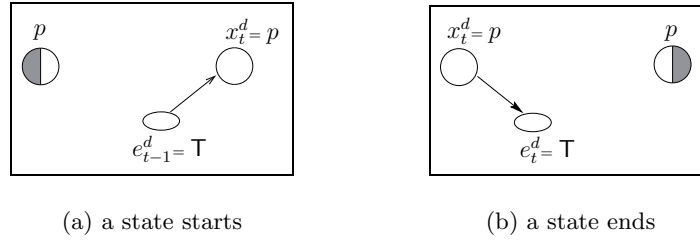


Figure 5-10: Sub-network configuration for the event ‘start’ and ‘end’ of a state $x_t^d = p$. A state starts is represented as a half-left-shaded node, and when it ends a half-right-shaded is used.

5.2.3 Conditional Independence in the DBN structure of the HHMM

In this subsection, we focus on a set of conditional independencies in the DBN structure of the HHMM. In particular, we provide two theorems and two lemmas for the conditional independence for the symmetric and asymmetric boundary conditions. This set of results will be used extensively in Section 5.5.2 and Appendix B to compute the set of auxiliary variables. For convenience, we introduce a pair of random variables $(\cdot\tau_t^d, \tau_t^d)$ to denote the starting and ending time indices of state x_t^d . This definition requires:

$$1 \leq \cdot\tau_t^d = l \leq t, \quad t \leq \tau_t^d = r \leq T \quad (5.14)$$

This condition is also equivalent to the event $\{e_{l-1}^d = 1, e_{l:r-1}^d = \mathbf{0}, e_r^d = 1\}$.

5.2.3.1 Symmetric independence theorem

We define the symmetric boundary condition as follows.

Definition 5.1 (Sym-Bound) *A symmetric boundary condition for a state p at time t and level d , started at time l and ended at time r , for $l \leq t \leq r$, is the set of events:*

$$\text{SB}_{l:r}^{d,p} \triangleq \left\{ x_t^d = p, \cdot\tau_t^d = l, \tau_t^d = r \right\}$$

Governed by the ending-condition in Equation-(5.6a), and observing $\cdot\tau_t^d = l$ leads to a chain of effects: first, the state x_{l-1}^d is terminated at time $l-1$ ($e_{l-1}^d = 1$), and, therefore, all its descendants must also terminate³, ie: $e_{l-1}^{d+1:D} = \mathbf{1}$. This also implies that the

³For brevity, we denote a vector whose elements are 1 everywhere by $\mathbf{1}$, and write $e_{l-1}^{d+1:D} = \mathbf{1}$ to mean $e_{l-1}^{d^*} = 1, \forall d^* = d+1, \dots, D$. Similar abbreviation is also applied for $\mathbf{0}$.

dependency (links) from $x_{l-1}^{d^*}$ to $x_l^{d^*}$ for all $d^* > d$ no longer hold and, therefore, can be removed from the network.

Similarly, $\tau_t^d = r$ implies $e_r^{d:D} = \mathbf{1}$, and all links from $x_r^{d^*}$ to $x_{r+1}^{d^*}$ for $d^* > d$ are broken. These facts together with the condition in Equation-(5.14) allow us to construct the network for this boundary event as in Figure-(5-11). Based on the construction of this

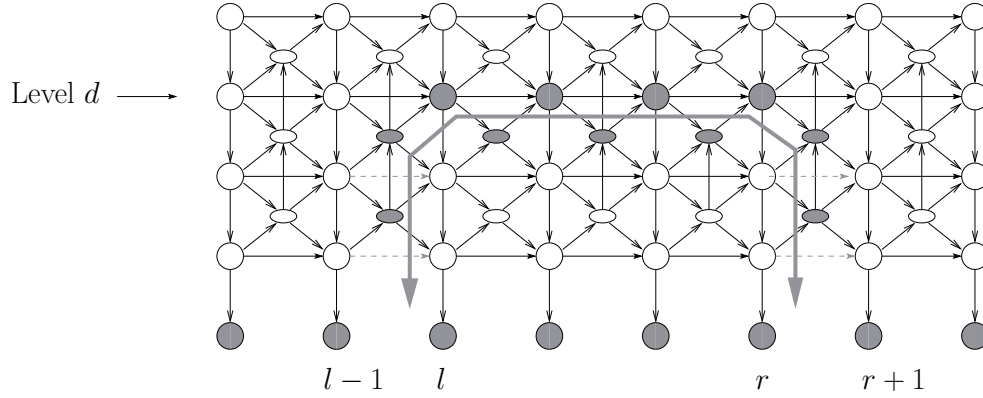


Figure 5-11: Symmetric boundary event $\text{SB}_{l:r}^{d,p} \triangleq \{x_t^d = p, \bullet\tau_t^d = l, \tau_t^d = r\}$. Nodes along the boundary are shaded to indicate that they are observed. Removed links are shown in dashed-lines.

boundary, we propose the following theorem.

Theorem 5.1 (Sym-Bound) Let $\text{SB}_{l:r}^{d,p}$ be a symmetric boundary condition. Further, let $\text{SI}_{l:r}^{d,p}$ and $\text{SO}_{l:r}^{d,p}$ be the set of all variables ‘inside’ and ‘outside’ this boundary respectively, ie:

$$\begin{aligned} \text{SI}_{l:r}^{d,p} &\triangleq \{x_{l:r}^{d+1:D}, e_{l:r-1}^{d+1:D}, y_{l:r}\} \\ \text{SO}_{l:r}^{d,p} &\triangleq \mathcal{V} \setminus \{\text{SB}_{l:r}^{d,p} \cup \text{SI}_{l:r}^{d,p}\} \end{aligned}$$

then the following conditional independence holds:

$$\text{SI}_{l:r}^{d,p} \perp\!\!\!\perp \text{SO}_{l:r}^{d,p} \mid \text{SB}_{l:r}^{d,p}$$

Proof. The proof is obtained by evaluating d-separation on the DBN network. Applying the Baye’s Ball algorithm (cf. Section 2.3.2, Chapter 2) on the network in Figure-(5-11) for the set of ‘inside’ variables and ‘outside’ variables and conditioning on the boundary event immediately yields the results for Theorem 5.1. ■

5.2.3.2 Asymmetric independence theorem

Definition 5.2 (Asym-Bound) An asymmetric boundary condition at level d for a state p starts at time l , and its children i at the lower level ends at time r for $l \leq r$ is the set of events⁴:

$$AB_{l:r}^{d,p}(i) \triangleq \left\{ \cdot x_l^d = p, \tau_l^d \geq r, x_r^{d+1} = i \right\}$$

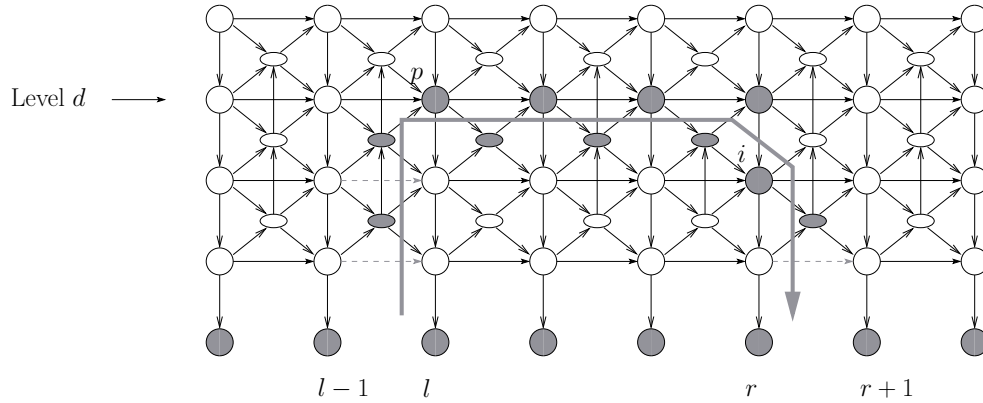


Figure 5-12: Illustration of the asymmetric boundary event: $AB_{l:r}^{d,p}(i) \triangleq \left\{ \cdot x_l^d = p, \tau_l^d \geq r, x_r^{d+1} = i \right\}$.

Figure-(5-12) depicts the asymmetric boundary condition. Similar to the previous case, we formulate the following theorem for the asymmetric case:

Theorem 5.2 (Asym-Bound) Let $AB_{l:r}^{d,p}(i)$ be an asymmetric boundary condition. Further, let $AI_{l:r}^{d,p}(i)$ and $AO_{l:r}^{d,p}(i)$ be the set of all variables ‘inside’ and ‘outside’ this boundary respectively, ie:

$$AI_{l:r}^{d,p}(i) \triangleq \left\{ x_{l:r-1}^{d+1:D}, x_r^{d+2:D}, e_{l:r-1}^{d+1:D}, y_{l:r} \right\}$$

$$AO_{l:r}^{d,p}(i) \triangleq \mathcal{V} \setminus \left\{ AB_{l:r}^{d,p}(i) \cup AI_{l:r}^{d,p}(i) \right\}$$

then the following conditional independence holds:

$$AI_{l:r}^{d,p}(i) \perp\!\!\!\perp AO_{l:r}^{d,p}(i) \mid AB_{l:r}^{d,p}(i)$$

Proof. Again, evaluating d-separation in the network in Figure-(5-12) for set of variables in $AI_{l:r}^{d,p}(i)$ and $AO_{l:r}^{d,p}(i)$ and conditioning on $AB_{l:r}^{d,p}(i)$ gives the above results. \blacksquare

⁴Note that the event $\tau_l^d \geq r$ is also equivalent with the event $\{e_{l:r-1}^d = \mathbf{0}\}$. We, therefore, occasionally switch between these two events during mathematical manipulation because $\cdot \tau_t^d$ or τ_t^d are random variables and thus allow us to properly perform the sum operator.

5.2.3.3 The Start-to-End (STE) and Started-Idp (SI) lemmas

The final form of conditional independence in the DBN we exploit is the independence of the starting and ending times of a state.

Lemma 5.1 (STE) *Assume that at level d , a state p has started at time t , and ends at time r . Let Z_t^{OUT} denote the set of all variables prior to t and above level d :*

$$Z_t^{\text{OUT}} \triangleq \mathcal{V}_{1:t-1} \cup \left\{ x_{t:r}^{1:d-1}, e_{t:r-1}^{1:d-1} \right\}$$

then the following conditional independence holds for its ending time τ_t^d and the observations during its execution:

$$Z_t^{\text{OUT}} \perp\!\!\!\perp \left\{ \tau_t^d = r, y_{t:r} \right\} \mid \left\{ \cdot x_t^d = p \right\}$$

Proof. A graphical version for this theorem is depicted in Figure-(5-13). As it appears,

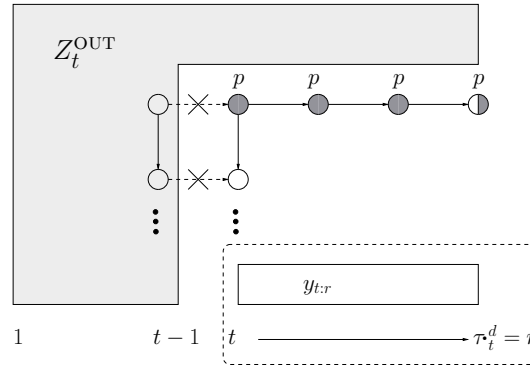


Figure 5-13: Graphical representation for Lemma 5.1.

this is a special case of the symmetric-boundary theorem when the right part from r to T in the network is removed, and hence, the proof is obtained accordingly. ■

Lemma 5.2 (SI) *Assume that at level $d+1$ state i has ended at time t , and its current parent is p . Let Z_t^{OUT} denote the set of all variables prior to $t+1$ and above d , but excluding $\{i, p\}$. Further, let j denote the state activated at time $t+1$ (at the same level) from i , then the following conditional independence holds:*

$$Z_t^{\text{OUT}} \perp\!\!\!\perp \left\{ \cdot x_{t+1}^{d+1} = j, e_t^d = 0 \right\} \mid \left\{ x_t^{d+1} = i, x_t^d = p \right\}$$

Proof. The graphical representation for this theorem is shown in Figure-(5-14). The proof

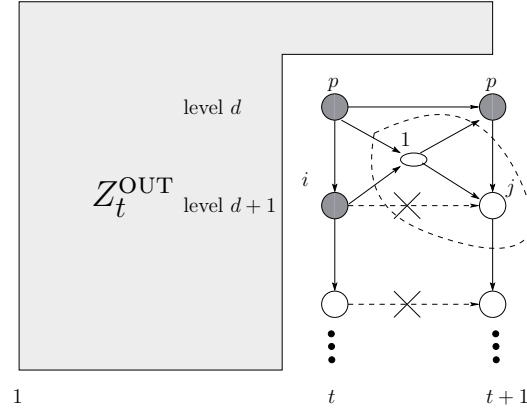


Figure 5-14: Graphical representation for Lemma 5.2.

is obtained through the following properties in conditional independence (Pearl, 1998):

$$\begin{aligned} \text{If } A \perp\!\!\!\perp D \mid \{B, C\} \text{ and } B \perp\!\!\!\perp D \mid C \\ \text{then: } \{A, B\} \perp\!\!\!\perp D \mid C. \end{aligned}$$

From Figure-(5-14) and using d -separation, we clearly have:

$$\begin{aligned} \cdot x_{t+1}^{d+1} = j \perp\!\!\!\perp Z_t^{\text{OUT}} \mid \{x_t^d = p, x_t^{d+1} = i, e_t^d = 0\} \\ \text{and } e_t^d \perp\!\!\!\perp Z_t^{\text{OUT}} \mid \{x_t^d = p, x_t^{d+1} = i\} \end{aligned}$$

and, therefore Lemma 5.2 is attained as: $Z_t^{\text{OUT}} \perp\!\!\!\perp \{ \cdot x_{t+1}^{d+1} = j, e_t^d = 0 \} \mid \{ x_t^{d+1} = i, x_t^d = p \}$. We note, however, that the above conditional independence is *not* held if the condition $\{e_t^d = 0\}$ is replaced by $\{e_t^d = 1\}$ (because the first conditional independence is violated). ■

5.3 Sufficient Statistics and Maximum-Likelihood for the HHMM

In this section, we provide a solution for a Maximum-Likelihood (ML) estimation of the parameters θ when the model is fully observed. We derive the form of the sufficient statistics (SS) for θ directly from the expression of the complete log-likelihood function, $\ell^c(\mathcal{D} \mid \theta)$. Alternatively, the sufficient statistics can be obtained indirectly by viewing the dynamic Bayesian Network presented in Section 5.2 as the result of a sequence of *parameter tying transformations*. This is an interesting and intuitive way to obtain the

SS and can be applied to more general models. However, for the sake of readability, we do not present it here and leave it to Appendix C.

The roadmap for this section is as follows. Since the key to our derivation is based on the well-known fact that BNs belong to a class of the *exponential family* (eg: see (Dan, 1998)), we briefly define this family of distributions in Section 5.3.1. Next, in Section 5.3.2, we show how the dynamic Bayesian Network representation for the HHMM can be viewed as an exponential family, from which the sufficient statistics are derived. The final forms of the ML-solution are given in Section 5.3.3.

5.3.1 The Exponential Family

A probability density is said to belong to the family of exponential distributions (Jordan, 2004, ch.8) if it takes the form:

$$\Pr(x | \eta) = h(x) \exp \{ \eta^T T \langle \eta \rangle - A(\eta) \} \quad (5.15)$$

where η is the parameter vector; $h(x)$ is a function that reflects the underlying measure with respect to which $\Pr(x | \eta)$ is a proper density; $A(\cdot)$ is the log-partition function, calculated as:

$$A(\eta) = \log \int h(x) \exp \{ \eta^T T \langle \eta \rangle \} dx$$

Of most importance to us, by Fisher-Neyman-Factorization theorem (Nowak and Scoot, 2003), $T \langle \eta \rangle$ is the *sufficient statistics* for the parameter η . We note that $T \langle \eta \rangle$ is a *function of the data* $x: \mathcal{X} \rightarrow \mathbb{R}^M$, where \mathcal{X} is the domain of x and M is the dimension of the parameter space. The data x remains *constant* in all of our settings, hence we do not include it explicitly in the expression of the sufficient statistics $T \langle \eta \rangle$.

The important feature of the exponential family to us is that the sufficient statistics can be obtained by direct inspection once the distribution is expressed in the canonical form of Equation-(5.15).

5.3.2 DBN as an Exponential Density and the Sufficient Statistics

Assume that the data observed consists of K iid. sequences, $\mathcal{D} = \{ \mathcal{O}^{(1)}, \dots, \mathcal{O}^{(K)} \}$. In the fully-observed case, each sequence $\mathcal{O}^{(k)}$ is a realisation of all variables in $\mathcal{V} =$

$\{x_{1:T}^{1:D}, e_{1:T-1}^{1:D}, y_{1:T}\}$. The likelihood function is given as:

$$\mathcal{L}(\mathcal{D} | \theta) = \prod_{k=1}^K \Pr(\mathcal{O}^{(k)} | \theta) = \prod_{k=1}^K \Pr(\mathcal{V}^{(k)} | \theta) \quad (5.16)$$

Follow the JPD factorisation form for the DBN in Equation-(5.10), the complete log-likelihood is, hence, given as:

$$\begin{aligned} \ell^c(\mathcal{D} | \theta) &= \log \mathcal{L}(\mathcal{D} | \theta) = \sum_{k=1}^K \log \Pr(\mathcal{V}^{(k)} | \theta) \\ &= \sum_{k=1}^K \left(\sum_{t=1}^T \sum_{d=1}^{D-1} \log \psi_{(x_t^{d+1})}^{(k)} + \sum_{t=1}^{T-1} \sum_{d=1}^{D-1} \log \psi_{(e_t^{d+1})}^{(k)} + \sum_{t=1}^T \log \psi_{(y_t)}^{(k)} \right) \end{aligned} \quad (5.17)$$

To show that the likelihood function $\mathcal{L}(\mathcal{D} | \theta)$ is in the exponential family, we use the ‘identity trick’ and write its corresponding log-form, $\ell^c(\mathcal{D} | \theta)$, in terms of θ and its sufficient statistics $\mathbb{T} \langle \theta \rangle$. That is, we show that Equation-(5.17) can be written as:

$$\ell^c(\mathcal{D} | \theta) = \mathbb{T} \langle \theta \rangle \log \theta \quad (5.18)$$

Again, we note that $\mathbb{T} \langle \theta \rangle$ is a function of \mathcal{D} , and this expression is having its most general form. The parameter in θ is actually decoupled into its local parameterisation forms. As an example, consider the term $\log \psi_{(y_t)}$. For clarity, we assume there is only one observation sequence for now and thus, drop the script k :

$$\begin{aligned} \log \psi_{(y_t)} &= \log \Pr(y_t | x_t^D) \stackrel{(a)}{=} \sum_{v \in \mathcal{Y}} \sum_{i \in \mathcal{S}^D} \log \Pr(y_t = v | x_t^D = i) \delta_{y_t}^{(v)} \delta_{x_t^D}^{(i)} \\ &= \sum_{v \in \mathcal{Y}} \sum_{i \in \mathcal{S}^D} \left(\delta_{y_t}^{(v)} \delta_{x_t^D}^{(i)} \right) \log B_{v|i} \end{aligned}$$

where in step (a), we sum over the domain of y_t and x_t^D , and raise the term $\Pr(y_t = v | x_t^D = i)$ to the identity functions $\delta_{y_t}^{(v)} \delta_{x_t^D}^{(i)}$. Taking the sum over t both sides of this equation, we have:

$$\sum_{t=1}^T \log \psi_{(y_t)} = \sum_{v \in \mathcal{Y}} \sum_{i \in \mathcal{S}^D} \left(\sum_{t=1}^T \delta_{y_t}^{(v)} \delta_{x_t^D}^{(i)} \right) \log B_{v|i} = \sum_{v \in \mathcal{Y}} \sum_{i \in \mathcal{S}^D} \mathbb{T} \langle B_{v|i} \rangle \log B_{v|i} \quad (5.19)$$

$$\text{where: } \mathbb{T} \langle B_{v|i} \rangle \triangleq \sum_{t=1}^T \left(\delta_{y_t}^{(v)} \delta_{x_t^D}^{(i)} \right) \quad (5.20)$$

Equation-(5.19) can also be rearranged into the product of two vectors: $\mathbb{T} \langle B \rangle \log B$, which identifies $\mathbb{T} \langle B_{v|i} \rangle$ as the sufficient statistics for the parameter $B_{v|i}$. Equation-(5.20) clearly shows that the sufficient statistics is a function of the data \mathcal{D} .

Similarly, with respect to the terms $\log \psi_{(x_t^{d+1})}$ and $\log \psi_{(e_t^{d+1})}$, we have:

$$\begin{aligned} \sum_{t=1}^T \log \psi_{(x_t^{d+1})} + \sum_{t=1}^{T-1} \log \psi_{(e_t^{d+1})} &= \sum_{p \in \mathcal{S}^d} \sum_{i,j \in \text{ch}(p)} \mathbb{T} \langle A_{i,j}^{d,p} \rangle \log A_{i,j}^{d,p} + \sum_{p \in \mathcal{S}^d} \sum_{i \in \text{ch}(p)} \mathbb{T} \langle \pi_i^{d,p} \rangle \log \pi_i^{d,p} \\ &\quad + \sum_{p \in \mathcal{S}^d} \sum_{i \in \text{ch}(p)} \mathbb{T} \langle A_{i,\text{end}}^{d,p} \rangle \log A_{i,\text{end}}^{d,p} \end{aligned} \quad (5.21)$$

where the sufficient statistics are given as:

$$\mathbb{T} \langle A_{i,j}^{d,p} \rangle \triangleq \sum_{t=2}^T \delta_{x_t^d}^{(p)} \delta_{x_{t-1:t}^{d+1}}^{(i,j)} \delta_{e_{t-1}^{d,d+1}}^{(0,1)} \quad \mathbb{T} \langle A_{i,\text{end}}^{d,p} \rangle \triangleq \sum_{t=1}^{T-1} \delta_{e_t^d}^{(1)} \delta_{x_t^{d,d+1}}^{(p,i)} \quad (5.22a)$$

$$\mathbb{T} \langle \pi_i^{d,p} \rangle \triangleq \sum_{t=2}^T \delta_{x_t^d}^{(p)} \delta_{x_t^{d+1}}^{(i)} \delta_{e_{t-1}^{d,d+1}}^{(1,1)} + \delta_{x_1^d}^{(p)} \delta_{x_1^{d+1}}^{(i)} \quad (5.22b)$$

Substituting Equation-(5.19) and Equation-(5.21) into Equation-(5.17) (assume there is still only one observation sequence), the log-likelihood function can now be written as:

$$\begin{aligned} \ell^c(\mathcal{D} \mid \theta) &= \sum_{d=1}^{D-1} \left(\sum_{p,i,j} \mathbb{T} \langle A_{i,j}^{d,p} \rangle \log A_{i,j}^{d,p} + \sum_{p,i} \mathbb{T} \langle A_{i,\text{end}}^{d,p} \rangle \log A_{i,\text{end}}^{d,p} + \sum_{p,i} \mathbb{T} \langle \pi_i^{d,p} \rangle \log \pi_i^{d,p} \right) \\ &\quad + \sum_{v,i} \mathbb{T} \langle B_{v|i} \rangle \log B_{v|i} \end{aligned} \quad (5.23)$$

Equation-(5.23) clearly shows that the log-likelihood function $\ell^c(\mathcal{D} \mid \theta)$ belongs to an exponential family as required. The sufficient statistics for θ are, thus, readily available from the expression of $\ell^c(\mathcal{D} \mid \theta)$ and given in Equation-(5.20), and Equation-(5.22)).

When the observed data \mathcal{D} contains K iid. sequences, it can be shown that (eg: Jordan (2004)) the sufficient statistics identified in the previous section remains the same with an extra summation over k as shown in Equation-(5.24) and Equation-(5.25) where in the calculation of $\mathbb{T} \langle \pi_i^{d,p} \rangle$ we assume $\delta_{e_0^{d,d+1}}^{(1,1)}$ (when $t = 1$) is 1 for convenience.

$$\mathbb{T} \langle A_{i,j}^{d,p} \rangle = \sum_{k=1}^K \sum_{t=2}^T \delta_{x_t^d}^{(p)} \delta_{x_{t-1:t}^{d+1}}^{(i,j)} \delta_{e_{t-1}^{d,d+1}}^{(0,1)} \quad \mathbb{T} \langle \pi_i^{d,p} \rangle = \sum_{k=1}^K \sum_{t=1}^{T-1} \delta_{x_t^d}^{(p)} \delta_{x_t^{d+1}}^{(i)} \delta_{e_{t-1}^{d,d+1}}^{(1,1)} \quad (5.24)$$

$$\mathbb{T} \langle A_{i,\text{end}}^{d,p} \rangle = \sum_{k=1}^K \sum_{t=1}^T \delta_{e_t^d}^{(1)} \delta_{x_t^{d,d+1}}^{(p,i)} \quad \mathbb{T} \langle B_{v|i} \rangle = \sum_{k=1}^K \sum_{t=1}^T \left(\delta_{y_t}^{(v)} \delta_{x_t^D}^{(i)} \right) \quad (5.25)$$

5.3.3 Maximum-Likelihood Estimation

In a ML parameter estimation framework, the objective is to find an optimum θ_{ML} that maximises the log-likelihood function with respect to the constraints of the model in Equation-(5.2):

$$\theta_{ML} = \operatorname{argmax}_{\theta} \ell^c(\mathcal{D} \mid \theta)$$

The general technique when doing the optimisation is to decouple the likelihood function and group variables with the same constraints together and optimise them separately. Since all variables are discrete, we use the technique of Lagrangian multipliers to do the optimisation. The key to this optimising process is the following result⁵.

Theorem 5.3 *Let \vec{c} and \vec{z} be two M -dimension vectors, whose elements are non-negative: $\vec{c} = \{c_i \mid c_i \geq 0\}_{i=1}^M$, $\vec{z} = \{z_i \mid z_i \geq 0\}_{i=1}^M$. Let the objective function be:*

$$f(\vec{z}) = \vec{c} \cdot \log \vec{z} = \sum_{i=1}^M c_i \log z_i$$

with the constraint: $\sum_{i=1}^M z_i = 1$. Then $f(\vec{z})$ is maximised for: $\hat{z}_i = c_i / \sum_{i=1}^M c_i$.

As an example, let us consider the case for re-estimating $A_{i,j}^{d,p}$. The constraint for this parameter from Equation-(5.2) is:

$$\sum_{j \in \text{ch}(p)} A_{i,j}^{d,p} + A_{i,\text{end}}^{d,p} = 1 \quad (5.26)$$

Clearly the row vector $A_{(i)}^{d,p}$ is a multinomial distribution over the children j (of p) plus the extra ending status. The relevant terms for this constraint from complete log-likelihood function, therefore, are:

$$\ell^c(\mathcal{D} \mid A_{(i)}^{d,p}) = \sum_{j \in \text{ch}(p)} \text{T} \langle A_{i,j}^{d,p} \rangle \log A_{i,j}^{d,p} + \text{T} \langle A_{i,\text{end}}^{d,p} \rangle \log A_{i,\text{end}}^{d,p} \quad (5.27)$$

Optimising $\ell^c(\mathcal{D} \mid A_{(i)}^{d,p})$ in Equation-(5.27) with the constraint in Equation-(5.26) is clearly in the form of the result in Theorem 5.3. The ML-solution is thus given as:

$$\begin{aligned} \hat{A}_{i,j}^{d,p} &= \frac{\text{T} \langle A_{i,j}^{d,p} \rangle}{A^{(\Sigma)}}, & \hat{A}_{i,\text{end}}^{d,p} &= \frac{\text{T} \langle A_{i,\text{end}}^{d,p} \rangle}{A^{(\Sigma)}} \\ \text{where } A^{(\Sigma)} &= \sum_{j \in \text{ch}(p)} \text{T} \langle A_{i,j}^{d,p} \rangle + \text{T} \langle A_{i,\text{end}}^{d,p} \rangle \end{aligned} \quad (5.28)$$

⁵Proof is provided in Appendix A.

For brevity, we write:

$$\hat{A}_{(i)}^{d,p} = \text{norm} \left\{ \text{T} \left\langle A_{(i)}^{d,p} \right\rangle, \text{T} \left\langle A_{i,\text{end}}^{d,p} \right\rangle \right\} \quad (5.29)$$

Doing similar optimisation routines with other constraints we obtain the ML-solution for the rest of the parameter set as:

$$\hat{\pi}^{d,p} = \text{norm} \left\{ \text{T} \left\langle \pi^{d,p} \right\rangle \right\}, \quad \hat{B}_{(i)} = \text{norm} \left\{ \text{T} \left\langle \mathbf{B}_{(i)} \right\rangle \right\} \quad (5.30)$$

Algorithm 5.1 summarises the ML-estimation for the HHMM in the fully observed data case.

Algorithm 5.1 ML-estimation for the HHMM in the fully observed data case

Input: A discrete HHMM specified by $\langle \zeta, \theta, \mathcal{V} \rangle$, and the fully observed data set \mathcal{D} .

- Calculate the sufficient statistics for θ as in Equation-(5.24) and Equation-(5.25).
- Obtain the ML-estimated solution for $\hat{\theta}$ as in Equation-(5.29) and Equation-(5.30).

Output: ML-estimated parameter $\hat{\theta}$.

5.4 Expected Sufficient Statistics and EM Estimation

We have shown that in the case where the model is fully observed, ML-estimation from the complete data set $\mathcal{D} = \{\mathcal{V}^{(k)}\}$ is reduced to computing the sufficient statistics for θ as the counts of different types of configurations. In many cases, we, however, observe only a subset \mathcal{O} of the variables \mathcal{V} and the remaining of variables $\mathcal{H} = \mathcal{V} \setminus \mathcal{O}$ are hidden (latent variables). More commonly, what we observe is the sequence $y_{1:T}$, together with the implicit assumption that the HHMM does not end prior to T . Thus, our observation set is $\mathcal{O} = \{y_{1:T}, e_{1:T-1}^1 = \mathbf{0}\}$, and we write the set of K iid. observed data as $\mathcal{D} = \{\mathcal{O}^{(1)}, \dots, \mathcal{O}^{(K)}\}$. In the rest of this chapter, we will explicitly work with this observation set⁶. However, the method presented here can generally be applied to any arbitrary set of observations.

In the presence of latent variables, we use the Expectation-Maximisation (EM) algorithm (Dempster *et al.*, 1977) to estimate θ . This is a well-known algorithm to do ML parameter estimation by alternating between the E-step and the M-step. In our case, it

⁶We note that Fine *et al.* assume that the observation also includes $e_1^T = 1$. This limits training data to the set of complete sequences of observations from the start to the end of the HHMM generation process. Removing this terminating condition allows us to train the HHMM with observed data that does not have to last until the end of the process.

reduces to first calculating the expected sufficient statistics (E-step), followed by a normalisation step to re-estimate the parameter (M-step). For the sake of completeness, we briefly explain the EM algorithm here and refer readers to various sources for details (eg: Dempster *et al.* (1977); Prescher (2003); Jordan (2004); Bilmes (1998)). Our goal is to identify the form of the expected sufficient statistics (ESS) and we introduce a set of auxiliary variables to calculate the ESS.

5.4.1 Expectation-Maximisation Algorithm

In a Maximum-Likelihood estimation framework, our goal is to maximise the log-likelihood with respect to θ :

$$\hat{\theta} = \operatorname{argmax}_{\theta} \log \Pr(\mathcal{D} \mid \theta) \quad (5.31)$$

Assume that the data set \mathcal{D} contains only a single observation sequence \mathcal{O} , the log-likelihood is then written as:

$$\begin{aligned} \ell(\mathcal{D} \mid \theta) &= \log \Pr(\mathcal{O} \mid \theta) = \log \sum_{\mathcal{H}} \Pr(\mathcal{H}, \mathcal{O} \mid \theta) \\ &= \log \sum_{\mathcal{H}} \left(\frac{\Pr(\mathcal{V} \mid \theta)}{Q(\mathcal{H})} Q(\mathcal{H}) \right) = \log \left\langle \frac{\Pr(\mathcal{V} \mid \theta)}{Q(\mathcal{H})} \right\rangle_Q \\ &\stackrel{(a)}{\geq} \left\langle \log \frac{\Pr(\mathcal{V} \mid \theta)}{Q(\mathcal{H})} \right\rangle_Q = \langle \log \Pr(\mathcal{V} \mid \theta) \rangle_Q - \langle \log Q \rangle_Q \\ &= \langle \ell^c(\mathcal{V} \mid \theta) \rangle_Q - \mathbb{H}(Q) \triangleq \Lambda(Q; \theta) \end{aligned} \quad (5.32)$$

where the entropy for Q is denoted as $\mathbb{H}(Q)$, and $\langle \cdot \rangle_Q$ is the expectation operator with respect to the distribution Q . The inequality in step (a) is a direct application of Jensen's inequality with the knowledge of the concavity in the $\log(\cdot)$ function. Starting with some random parameter $\theta^{(0)}$, the EM algorithm iteratively optimises $\ell(\mathcal{D} \mid \theta)$ by alternating between the following E-step and M-step:

$$\text{E-step:} \quad Q^{(t+1)} = \operatorname{argmax}_Q \Lambda(Q; \theta^{(t)}) \quad (5.33a)$$

$$\text{M-step:} \quad \theta^{(t+1)} = \operatorname{argmax}_{\theta} \Lambda(Q^{(t+1)}, \theta) \quad (5.33b)$$

It can be shown that the E-step in Equation-(5.33a) is achieved when the Q distribution is chosen to be:

$$Q^{(t+1)}(\mathcal{H}) = \Pr(\mathcal{H} \mid \mathcal{O}, \theta^{(t)}) \quad (5.34)$$

and the expectation is now taken over the complete log-likelihood $\ell^c(\mathcal{V} \mid \theta^{(t)})$, which, loosely speaking, ‘fills’ in the values for the hidden variables by their expectation⁷. In the M-step, since the $\mathbb{H}(Q)$ is independent of θ , the estimated $\theta^{(t+1)}$ at the $(t+1)$ -th iteration is determined as:

$$\hat{\theta}^{(t+1)} = \operatorname{argmax}_{\theta} \left\langle \ell^c(\mathcal{V} \mid \theta^{(t)}) \right\rangle \quad (5.35)$$

The EM algorithm is essentially a coordinate-ascent optimisation process by first (i) fixing θ and maximising Q (E-step), and then (ii) fixing Q and maximising θ . This is a two-step hill climbing process and is guaranteed to converge⁸. The maximisation step is similar to what we have done in the fully observed data case once the expected sufficient statistics has been computed.

5.4.2 The Complete Log-likelihood Function and the M-step

Taking the expectation for the complete log-likelihood in Equation-(5.23) over $Q(\mathcal{H})$ yields:

$$\begin{aligned} \langle \ell^c(\mathcal{D} \mid \theta) \rangle_Q &= \sum_{d=1}^{D-1} \left(\sum_{p,i,j} \langle A_{i,j}^{d,p} \rangle_Q \log A_{i,j}^{d,p} + \sum_{p,i} \langle A_{i,\text{end}}^{d,p} \rangle_Q \log A_{i,\text{end}}^{d,p} + \sum_{p,i} \langle \pi_i^{d,p} \rangle_Q \log \pi_i^{d,p} \right) \\ &\quad + \sum_{v,i} \langle B_{v|i} \rangle_Q \log B_{v|i} \end{aligned} \quad (5.36)$$

where we use the notation $\langle f \rangle_Q$ to denote the expected value of f with respect to the distribution Q , and shorten the notation of the expectation over the sufficient statistics, eg: write $\langle \mathbb{T} \langle A_{i,j}^{d,p} \rangle \rangle$ as $\langle A_{i,j}^{d,p} \rangle$. The previous section has outlined that, in the E-step, the Q distribution is chosen to be $\Pr(\mathcal{H} \mid \mathcal{O}, \theta)$. Therefore, unless otherwise stated, throughout this chapter, the expectation is always taken with respect to this distribution.

Before going into details of the expected sufficient statistics, we note that in the M-step, maximising $\langle \ell^c(\mathcal{D} \mid \theta) \rangle$ in Equation-(5.36) is carried out in the same way as maximising $\ell^c(\mathcal{D} \mid \theta)$ in Equation-(5.23) presented in Section 5.3.3. Given that the expected sufficient statistics are computed, the ML-solution during the M-step is:

$$\hat{A}_{(i)}^{d,p} = \operatorname{norm} \left\{ \left\langle A_{(i)}^{d,p} \right\rangle, \left\langle A_{i,\text{end}}^{d,p} \right\rangle \right\} \quad \hat{\pi}^{d,p} = \operatorname{norm} \left\{ \left\langle \pi^{d,p} \right\rangle \right\} \quad (5.37a)$$

$$\hat{B}_{(i)} = \operatorname{norm} \left\{ \left\langle \mathbf{B}_{(i)} \right\rangle \right\} \quad (5.37b)$$

⁷Follows the interpretation mentioned in Jordan (2004).

⁸It is not too hard to prove the convergence. Essentially, by choosing $Q(\mathcal{H}) = \Pr(\mathcal{H} \mid \mathcal{O})$, the gap between the lower bound and the log-likelihood is closed, ie: $\ell(\mathcal{O} \mid \theta) = \Lambda(Q, \theta)$, then M-step guarantees the increase of the log-likelihood. Various sources provide formal proofs of convergence, eg: see Jordan (2004).

Assuming for now that the expected sufficient statistics can be computed, the pseudocode for the EM estimation is given in Algorithm 5.2. In the following subsection, we discuss

Algorithm 5.2 EM estimation for the HHMM

Input: A discrete HHMM specified by $\langle \zeta, \theta, \mathcal{Y} \rangle$, and the observed data set \mathcal{O} , and an initial parameter θ .

Set $\theta^* \leftarrow \theta$.

Loop until converge

E-step: using θ^* calculate the expected sufficient statistics $\langle A \rangle, \langle \pi \rangle, \langle B \rangle$.

M-step: obtain a ML-estimated $\hat{\theta} = \{ \hat{A}, \hat{\pi}, \hat{B} \}$ from Equation-(5.37).

Set $\theta^* \leftarrow \hat{\theta}$.

EndLoop

Output: ML-estimated parameter $\hat{\theta}$.

how to obtain the expected sufficient statistics during the E-step.

5.4.3 Expected Sufficient Statistics and Auxiliary Variables

The expected sufficient statistics for the model parameter θ is obtained by taking the expectation over the sufficient statistics identified in Equation-(5.24) and Equation-(5.25). For example, $\langle A_{i,j}^{d,p} \rangle$ is obtained as:

$$\begin{aligned} \langle A_{i,j}^{d,p} \rangle &= \sum_{k=1}^K \sum_{t=2}^T \left\langle \delta_{x_t^d}^{(p)} \delta_{x_{t-1:t}^{d+1}}^{(i,j)} \delta_{e_{t-1}^{d:d+1}}^{(0,1)} \right\rangle_{\Pr(\mathcal{H} | \mathcal{O}^{(k)})} \\ &\stackrel{(a)}{=} \sum_{k=1}^K \sum_{t=2}^T \Pr(x_t^d = p, x_{t-1}^{d+1} = i, x_t^{d+1} = j, e_{t-1}^{d:d+1} = 01 | \mathcal{O}^{(k)}) \end{aligned} \quad (5.38)$$

where in step (a) we have used the fact that the expectation of an indicator function is its corresponding form of probability. That is, for example, we would write:

$$\begin{aligned} \left\langle \delta_x^{(i)} \right\rangle_{\Pr(\mathcal{H} | \mathcal{O})} &= \sum_{\mathcal{H}} \delta_x^{(i)} \Pr(\mathcal{H} | \mathcal{O}) = \sum_x \delta_x^{(i)} \Pr(x | \mathcal{O}) \overbrace{\sum_{\mathcal{H} \setminus \{x\}} \Pr(\mathcal{H} \setminus \{x\} | \mathcal{O})}^{=1} \\ &= \Pr(x = i | \mathcal{O}) \end{aligned}$$

Performing similar analysis for the other parameters in θ , we have:

$$\langle A_{i,\text{end}}^{d,p} \rangle = \sum_{k=1}^K \sum_{t=1}^{T-1} \langle \delta_{e_t^d}^{(1)} \delta_{x_t^{d:d+1}}^{(p,i)} \rangle = \sum_{k=1}^K \sum_{t=2}^T \Pr(x_t^d = p, x_t^{d+1} = i, e_t^d = 1 \mid \mathcal{O}^{(k)}) \quad (5.39a)$$

$$\langle \pi_i^{d,p} \rangle = \sum_{k=1}^K \sum_{t=1}^{T-1} \langle \delta_{x_t^d}^{(p)} \delta_{x_t^{d+1}}^{(i)} \delta_{e_{t-1}^{d:d+1}}^{(1,1)} \rangle = \sum_{k=1}^K \sum_{t=1}^{T-1} \Pr(x_t^d = p, x_t^{d+1} = i, e_{t-1}^{d:d+1} = 11 \mid \mathcal{O}^{(k)}) \quad (5.39b)$$

$$\begin{aligned} \langle B_{i|v} \rangle &= \sum_{k=1}^K \sum_{t=1}^T \langle \delta_{y_t}^{(v)} \delta_{x_t^D}^{(i)} \rangle = \sum_{k=1}^K \sum_{t=1}^T \Pr(x_t^D = i, y_t = v \mid \mathcal{O}^{(k)}) \\ &= \sum_{k=1}^K \sum_{t=1}^T \Pr(y_t = v \mid x_t^D = i, \mathcal{O}^{(k)}) \Pr(x_t^D = i \mid \mathcal{O}^{(k)}) \end{aligned} \quad (5.39c)$$

The set of Equation-(5.38) and Equation-(5.39)) identify the probabilistic quantities needed to derive the ESS for θ . For convenience, we introduce a set of auxiliary variables for these quantities (disregarding the proportionality constant $1/\Pr(\mathcal{O})$):

- The *horizontal transition* probability variables, $\xi_t^{d,p}(i, j)$ and $\xi_t^{d,p}(i, \text{end})$, to compute $\langle A_{i,j}^{d,p} \rangle$ and $\langle A_{i,\text{end}}^{d,p} \rangle$ in Eqs-(5.38,5.39a):

$$\xi_t^{d,p}(i, j) \triangleq \Pr(x_t^d = p, x_t^{d+1} = j, x_{t-1}^{d+1} = i, e_{t-1}^{d:d+1} = 01, \mathcal{O}) \quad (5.40a)$$

$$\xi_t^{d,p}(i, \text{end}) \triangleq \Pr(x_t^d = p, x_t^{d+1} = i, e_t^{d:d+1} = 11, \mathcal{O}) \quad (5.40b)$$

- The *vertical transition* probability variable, $\chi_t^{d,p}(i)$, to compute $\langle \pi_i^{d,p} \rangle$ in Equation-(5.39b):

$$\chi_t^{d,p}(i) \triangleq \Pr(x_t^d = p, x_t^{d+1} = i, e_{t-1}^{d:d+1} = 11, \mathcal{O}) \quad (5.41)$$

- The *emission* probability variable, $\Gamma_t^D(i)$, to compute $\langle B_{v|i} \rangle$ in Equation-(5.39c):

$$\Gamma_t^D(i) \triangleq \Pr(x_t^D = i, \mathcal{O}) \quad (5.42)$$

These auxiliary variables are appealing because of their natural meanings. For example, the variable $\xi_t^{d,p}(i, j)$ represents the expected counting number of ‘seeing’ a transition from i to j at time t , and when summed over t , it represents the expected counts of seeing this configuration. The set of expected sufficient statistics for the parameter set θ can now be

translated into these variables via the following relations:

$$\langle A_{i,j}^{d,p} \rangle = \sum_{k=1}^K \left[\frac{\sum_{t=2}^T \xi_t^{d,p}(i,j)}{\Pr(\mathcal{O})} \right] \quad \langle \pi_i^{d,p} \rangle = \sum_{k=1}^K \left[\frac{\sum_{t=1}^T \chi_t^{d,p}(i)}{\Pr(\mathcal{O})} \right] \quad (5.43a)$$

$$\langle A_{i,\text{end}}^{d,p} \rangle = \sum_{k=1}^K \left[\frac{\sum_{t=1}^T \xi_t^{d,p}(i,\text{end})}{\Pr(\mathcal{O})} \right] \quad \langle B_{v|i} \rangle = \sum_{k=1}^K \left[\frac{\sum_{t=1}^T \Pr(y_t = v \mid x_t^D = i, \mathcal{O}) \Gamma_t^D(i)}{\Pr(\mathcal{O})} \right] \quad (5.43b)$$

where we implicitly understand that the observation sequence \mathcal{O} inside the bracket is referring to the k th sequence.

It is also noteworthy to mention the probability $\Pr(y_t \mid x_t^D, \mathcal{O})$ in the calculation of $\langle B_{i|v} \rangle$. When all y_t is observed, this probability becomes deterministic, ie: $\Pr(y_t = v \mid x_t^D = i, \mathcal{O}) = 1$, if y_t is observed as v in \mathcal{O} and $= 0$ otherwise. This allows us to easily extend the model to the missing observations case. That is, if y_t is not observed in \mathcal{O} , then this quantity is simply replaced by $B_{i|v}$.

5.5 Inference in the Hierarchical HMM

We have shown that computing the ESS reduces to evaluating a set of auxiliary variables and the likelihood $\Pr(\mathcal{O} \mid \theta)$. In this section, we present methods to compute these variables, first with the FST method, when the topology ζ is a tree, and then with our method, when ζ has a general lattice shared structure.

5.5.1 The Fine-Singer-Tishby (FST) Method

When the topology is strictly a tree, Fine *et al.* (Fine *et al.*, 1998) apply the inside-outside algorithm during the inference process⁹. Let us first describe the main intuition behind the FST method. Suppose we know that the state $x_t^d = p$ starts at time $l \leq t$ and stops at time $m \geq t$ (when these time indices are not know, we can simply sum over all possible values of l and m). Fine *et al.* observe that the auxiliary variable $\xi_t^{d,p}(i,j)$ can then be *factorised into the product of four different parts*:

- the ‘in’ part (η_{in}) consisting of the observations prior to l ,

⁹Readers who are interested in this algorithm are referred to (Lari and Young, 1990; Murphy, 2001; Fine *et al.*, 1998) for more details. Murphy in (Murphy, 2001) also briefly mentions how the inside-outside algorithm is used to derive the inference algorithm for the HHMM.

- the ‘forward’ part (α) consisting of the observations from l to t ,
- the ‘backward’ part (β) consisting of the observations from $t + 1$ to m ,
- and the ‘out’ part (η_{out}) consisting of observations after m .

Figure-(5-15) provides a diagrammatic illustration for this partitioning.

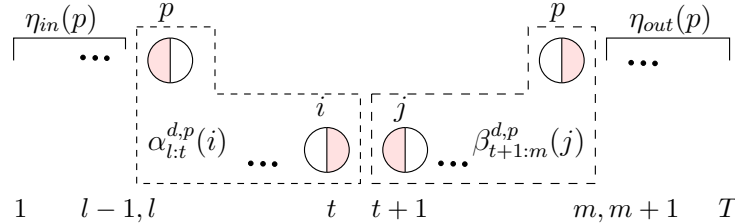


Figure 5-15: Decomposition of the auxiliary variable $\xi_t^{d,p}(i, j)$ in the FST method.

This form of factorisation is valid *only when p is assumed to have a unique set of ancestor-states* (ie: knowing $x_t^d = p$ also gives the observable of $x_t^{d'}$ for $d' < d$). This is equivalent to the tree condition in the topological structure ζ in Equation-(5.3). When this assumption holds, the ‘in’ and ‘out’ parts are conditionally independent given the state $x_t^d = p$ and its starting/ending times, and thus the factorisation in Figure-(5-15) is valid. Given this form of factorisation, four main types of auxiliary variables including ‘forward’, ‘backward’, ‘in’ and ‘out’ are defined and computed in the FST method. We discuss here a particular case of the ‘forward’ variable and show how it can be defined in the context of our DBN setting, and graphically show its recursion form. We shall, however, defer the proofs to the next section.

For a segment of observation from l to r , Fine *et al.* defines the forward variable as $\Pr(y_{l:r}, x_r^{d+1} = i \text{ finished at } r \mid x_l^d = p \text{ started at } l)$. This definition involves descriptive and informal words such as ‘started’, ‘finished’ and this thus makes it hard, if not impossible, to formally verify its recursion form. We re-define this variable as:

$$\alpha_{l:r}^{d,p}(i) \triangleq \Pr(y_{l:r}, x_r^{d+1} = i, e_{l:r-1}^d = \mathbf{0} \mid \cdot x_l^d = p) \quad (5.44)$$

Its diagrammatic structure can be seen from the left dashed-box in Figure-(5-15). The strategy to calculating this variable is to recursively work from left-to-right and bottom-up, consisting of an initialisation followed by the recursion loop. The initialisation phase occurs at the bottom when $d = D - 1$. By direct substitution into the definition, it is not

too hard¹⁰ to show that this initialisation step is:

$$\bullet \quad l = r = t : \quad \alpha_{t:t}^{D-1,p}(i) = \pi_i^{D-1,p} B_{y_t|i} \quad (5.45a)$$

$$\bullet \quad l < r : \quad \alpha_{l:r}^{D-1,p}(i) = \left[\sum_{j \in \text{ch}(p)} \alpha_{l:r-1}^{D-1,p}(j) A_{i,j}^{D-1,p} \right] B_{y_t|i} \quad (5.45b)$$

When $d < D - 1$, the recursion is (Figure-(5-16)):

$$\bullet \quad \alpha_{t:t}^{d,p}(i) = \pi_i^{d,p} \left[\sum_{s \in \text{ch}(i)} \alpha_{t:t}^{d+1,i}(s) A_{s,\text{end}}^{d+1,i} \right] \quad (5.46a)$$

$$\bullet \quad \alpha_{l:r}^{d,p}(i) = \sum_{s \in \text{ch}(i)} \left[\sum_{\substack{j \in \text{pa}(p) \\ l < t \leq r}} \alpha_{l:t-1}^{d,p}(j) A_{j,i}^{d,p} \alpha_{t:r}^{d+1,i}(s) + \overbrace{\pi_i^{d,p} \alpha_{l:r}^{d+1,i}(s)}^{\text{when } t=l} \right] A_{s,\text{end}}^{d+1,i} \quad (5.46b)$$

The set of Equation-(5.45) and Equation-(5.46) recover the same calculations presented in the original paper (Fine *et al.*, 1998).

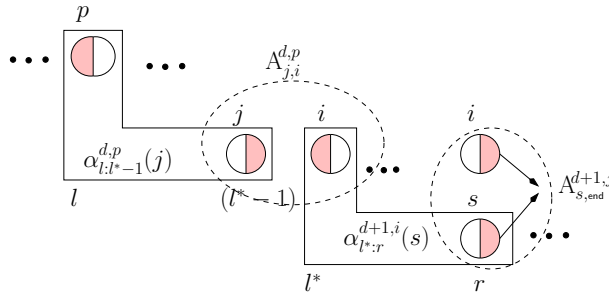


Figure 5-16: Diagrammatic forms of recursive partitions for $\alpha_{l:r}^{d,p}(i)$ in Equation-(5.46b) when r strictly greater than l .

5.5.2 The Asymmetric Inside-Outside Algorithm

When the topological structure ζ is generalised to be a lattice, the condition on the unique path for a state's ancestor as in the FST method no longer holds. When referring to Figure-(5-15), *the unknown ancestor-states of x_t^d destroys the conditional independence between the 'in' and 'out' parts, and thus the factorisation in the FST method no longer holds*. This generalisation in the topological structure calls for better forms of factorisation.

Our method requires two modifications to the original method. First, rather than summing over the stopping time m of the state x_t^d , we sum over the stopping time r of the child

¹⁰The correctness of this step is followed in our discussion on the AIO-algorithm.

state x_{t+1}^{d+1} . Secondly, we turn to a similar technique used by the inside-outside algorithm in the PCFG, and group all the observations outside the interval $[l, r]$ into one and call it the ‘*asymmetric outside*’ part (λ). Thus, the boundary between the inside and out parts is not symmetrical, hence we introduce an *asymmetric inside-outside* factorisation. The asymmetric conditional independence in Theorem 5.2 guarantees the factorisation of the ‘inside’ and ‘outside’ when conditioning on the boundary. The inside part is further factorised into an ‘*asymmetric forward*’ part (same as FST’s forward α), and a ‘*symmetric inside*’ part (Δ) as shown in Figure-(5-17).

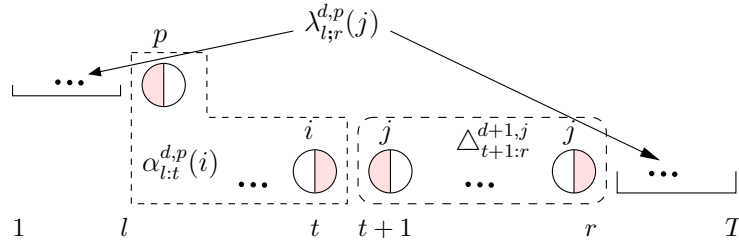


Figure 5-17: Decomposition of the auxiliary variable $\xi_t^{d,p}(i, j)$ in our AIO (Asymmetric Inside-Outside) method.

5.5.2.1 The set of inside/outside auxiliary variables

The newly identified auxiliary variables in Figure-(5-17) are formally defined as:

$$\alpha_{l:r}^{d,p}(i) \triangleq \Pr(\mathcal{O}_{l:r}^{\text{in}}, x_r^{d+1} = i, e_{l:r-1}^d = \mathbf{0} \mid \cdot x_l^d = p) \quad (5.47a)$$

$$\Delta_{l:r}^{d,i} \triangleq \Pr(\mathcal{O}_{l:r}^{\text{in}}, e_{l:r-1}^d = \mathbf{0}, e_r^d = 1 \mid \cdot x_l^d = i) \quad (5.47b)$$

$$\lambda_{l;r}^{d,p}(i) \triangleq \Pr(\mathcal{O}_{l;r}^{\text{out}} \mid \cdot x_l^d = p, e_{l:r-1}^d = \mathbf{0}, x_r^{d+1} = i) \Pr(\cdot x_l^d = p) \quad (5.47c)$$

where $\mathcal{O}_{l:r}^{\text{in}} = \{y_{l:r}\}$ is the set of observations ‘inside’, and $\mathcal{O}_{l:r}^{\text{out}} = \{y_{l:r-1}, y_{r+1:T}, e_{1:T-1}^1 = \mathbf{0}\}$ is the set of observations ‘outside’. Intuitively, the asym-inside variable $\alpha_{l:r}^{d,p}(i)$ contains all the observations between when a parent state p starts and its child state i ends; and its counterpart, the asym-outside variable $\lambda_{l;r}^{d,p}(i)$, contains all the remaining observations. The symmetric inside $\Delta_{l:r}^{d,i}$ contains all the observations in between the execution of a state i from its start to its end. For computational convenience, we also define a *started- α* , denoted as $\alpha_{l:r+1}^{d,p}(i)$, as a companion for $\alpha_{l:r}^{d,p}(j)$, and define it as:

$$\alpha_{l:r+1}^{d,p}(i) \triangleq \Pr(\mathcal{O}_{l:r}^{\text{in}}, \cdot x_{r+1}^{d+1} = i, e_{l:r}^d = \mathbf{0} \mid \cdot x_l^d = p) \quad (5.48)$$

We also find it useful to introduce the ‘*symmetric outside*’ variable $\Lambda_{l;r}^{d,p}$ as a companion for the asymmetric case. This variable is defined as:

$$\Lambda_{l;r}^{d,p} \triangleq \Pr(\mathcal{O}_{l;r}^{\text{out}} \mid \cdot x_l^d = p, e_{l:r-1}^d = \mathbf{0}, e_r^d = 1) \Pr(\cdot x_l^d = p) \quad (5.49)$$

Since the calculation involves complex mathematical manipulation, we devise a set of diagrammatic visualisation tools by which the recursive relations between these variables can be easily identified and computed. The diagrammatic forms for the inside/outside variables defined in Equations (5.47), (5.48) and (5.49) are provided in Figure-(5-18).

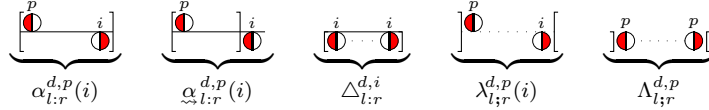


Figure 5-18: Diagrams for the inside and outside variables. Brackets denote the range of observations. Half-shaded circles denote starting/ending states.

5.5.2.2 Calculating the horizontal probability variables

Now let us return to the discussion of the AIO-algorithm and formally show the derivation of the horizontal transition probability variable $\xi_t^{d,p}(i, j)$ based on our newly defined inside/outside auxiliary variables. To serve as an example for other variables, we attempt to treat the derivation in rather substantial detail.

We start by writing down the definition of $\xi_t^{d,p}(i, j)$ in Equation-(5.40a), and then expand it according to the form suggested in Figure-(5-17):

$$\begin{aligned} \xi_t^{d,p}(i, j) &\triangleq \Pr(x_t^d = p, x_t^{d+1} = i, x_{t+1}^{d+1} = j, e_t^{d:d+1} = 01, \mathcal{O}) \\ &\stackrel{(a)}{=} \sum_{l,r} \Pr(\underbrace{\cdot \tau_t^d = l, \tau_{t+1}^{d+1} = r, x_t^d = p, x_t^{d+1} = i, x_{t+1}^{d+1} = j}_{A}, \underbrace{e_t^{d:d+1} = 01}_{B}, \mathcal{O}) \end{aligned}$$

where in step (a) we simply sum over starting time $\cdot \tau_t^d$ of p , and the ending time τ_{t+1}^{d+1} of j . Next, we group these starting/ending conditions A with $B \triangleq \{e_t^{d:d+1} = 01\}$ and write them in terms of events, that is, we write $\{A \cup B\}$ as:

$$\left\{ e_{l-1}^d = 1, e_{l:r-1}^d = \mathbf{0}, e_t^{d+1} = 1, e_{t+1:r-1}^{d+1} = \mathbf{0}, e_r^{d+1} = 1 \right\} \triangleq C$$

Furthermore, since $e_{t+1:r-1}^{d+1} = \mathbf{0}$ appears in C , we can now write $x_{t+1}^{d+1} = j$ as $x_r^{d+1} = j$ (ie: j continues from $t+1 \rightarrow r$). Let us continue from step (a):

$$\begin{aligned} &\stackrel{(b)}{=} \sum_{l,r} \Pr(\underbrace{e_{l-1}^d = 1, x_l^d = p, e_{l:r-1}^d = \mathbf{0}}_{\cdot x_l^d = p}, \underbrace{x_t^{d+1} = i, e_t^{d+1} = 1}_{x_t^{d+1} = i}, \underbrace{x_r^{d+1} = j, e_r^{d+1} = 1}_{x_r^{d+1} = j}, e_{t+1:r-1}^{d+1} = \mathbf{0}, \mathcal{O}) \\ &= \sum_{l,r} \Pr(\cdot x_l^d = p, e_{l:r-1}^d = \mathbf{0}, x_t^{d+1} = i, x_r^{d+1} = j, e_{t+1:r-1}^{d+1} = \mathbf{0}, \mathcal{O}) \end{aligned}$$

here in step (b) we have ‘absorbed’ the ‘ending’ condition into the node, eg: $\{e_{l-1}^d = 1, x_l^d = p\}$ is written as $\cdot x_l^d = p$. Next, we establish the asymmetric boundary event at time l and r and partition the observation set \mathcal{O} into the ‘inside’ $\mathcal{O}_{l:r}^{\text{in}}$ and ‘outside’ $\mathcal{O}_{l:r}^{\text{out}}$ parts

$$\stackrel{(c)}{=} \sum_{l,r} \Pr(\underbrace{\cdot x_l^d = p, e_{l:r-1}^d = \mathbf{0}, x_r^{d+1} = j}_{\text{AB}_{l:r}^{d,p}(j)}, \underbrace{e_{t+1:r-1}^{d+1} = \mathbf{0}, x_t^{d+1} = i, \cdot x_{t+1}^{d+1} = j, \mathcal{O}_{l:r}^{\text{in}}, \mathcal{O}_{l:r}^{\text{out}}}_{\text{INSIDE}})$$

Note that in step (c) there is an extra term $x_{t+1}^{d+1} = j$ compared with results in step (b). This is because $e_{t+1:r-1}^{d+1} = \mathbf{0}$ together with $x_r^{d+1} = j$ implies $x_{t+1}^{d+1} = j$ (state j stays the same from $t+1$ to r), then combined with $e_t^{d+1} = 1$ implies $\cdot x_{t+1}^{d+1} = j$.

The set of all variables is then grouped into an asymmetric boundary $\text{AB}_{l:r}^{d,p}(j)$ (cf. Definition 5.2), a set of observations outside this boundary $\mathcal{O}_{l:r}^{\text{out}}$ and the set of remaining inside variables INSIDE. As a corollary to Theorem 5.2 we have:

$$\mathcal{O}_{l:r}^{\text{out}} \perp\!\!\!\perp \text{INSIDE} \mid \text{AB}_{l:r}^{d,p}(j)$$

we can, therefore, further expand $\xi_t^{d,p}(i, j)$ into:

$$\begin{aligned} \xi_t^{d,p}(i, j) &= \sum_{l,r} \Pr\left(\mathcal{O}_{l:r}^{\text{out}} \mid \text{AB}_{l:r}^{d,p}(j), \text{INSIDE}\right) \Pr\left(\text{AB}_{l:r}^{d,p}(j), \text{INSIDE}\right) \\ &= \sum_{l,r} \frac{\lambda_{l:r}^{d,p}(j)}{\Pr(\cdot x_l^d = p)} \Pr\left(\text{AB}_{l:r}^{d,p}(j), \text{INSIDE}\right) \end{aligned} \quad (5.50)$$

where by definition in Equation-(5.47c), the term $\frac{\lambda_{l:r}^{d,p}(j)}{\Pr(\cdot x_l^d = p)}$ equates to the probability $\Pr(\mathcal{O}_{l:r}^{\text{out}} \mid \cdot x_l^d = p, e_{l:r-1}^d = \mathbf{0}, x_r^{d+1} = j)$, whose conditioned term on the RHS is exactly the same as the definition of the asymmetric boundary $\text{AB}_{l:r}^{d,p}(j)$.

As can be seen, the asymmetric outside part $\lambda_{l:r}^{d,p}(j)$ has emerged in Equation-(5.50). To continue from there, we expand the term $\left\{\text{AB}_{l:r}^{d,p}(j), \text{INSIDE}\right\}$ and rearrange the variables into useful groups to exploit the conditional independencies among them. This is a joint probability of the asymmetric boundary $\text{AB}_{l:r}^{d,p}(j)$ and all the variables inside this boundary. If we keep following the form of factorisation in Figure-(5-17), we hope to factorise this term into a symmetric Δ and an asymmetric forward part α .

Expanding and arranging $\left\{\text{AB}_{l:r}^{d,p}(j), \text{INSIDE}\right\}$ as follows:

$$\begin{aligned} &\underbrace{\{\cdot x_l^d = p, e_{l:r-1}^d = \mathbf{0}, x_r^{d+1} = j\}}_{\text{AB}_{l:r}^{d,p}(j)} \underbrace{\{e_{t+1:r-1}^{d+1} = \mathbf{0}, x_t^{d+1} = i, \cdot x_{t+1}^{d+1} = j, \mathcal{O}_{l:r}^{\text{in}}\}}_{\text{INSIDE}} \\ &\stackrel{(d)}{=} \{\cdot x_l^d = p, e_{l:t}^d = \mathbf{0}, e_{t+1:r-1}^{d+1} = \mathbf{0}, e_r^{d+1} = 1, x_t^{d+1} = i, \cdot x_{t+1}^{d+1} = j, \mathcal{O}_{l:t}^{\text{in}}, \mathcal{O}_{t+1:r}^{\text{in}}\} \end{aligned}$$

where in step (d), $e_{t+1:r-1}^{d+1} = \mathbf{0}$ implies $e_{t+1:r-1}^d = \mathbf{0}$, and together with $x_{t+1}^{d+1} = j$ implies that j stays the same from $t+1$ to r ; thus, we simplify $\{e_{l:r-1}^d = \mathbf{0}, e_{t+1:r-1}^{d+1} = \mathbf{0}, x_r^{d+1} = j\}$ into $\{e_{l:t}^d = \mathbf{0}, e_{t+1:r-1}^{d+1} = \mathbf{0}, e_r^{d+1} = 1\}$. In addition, we split the observation $\mathcal{O}_{l:r}^{\text{in}}$ into two parts: $\mathcal{O}_{l:t}^{\text{in}}$ and $\mathcal{O}_{t+1:r}^{\text{in}}$. Next, grouping all terms from time l to t together, ie: $U \triangleq \{\cdot x_l^d = p, e_{l:t}^d = \mathbf{0}, x_t^{d+1} = i, \mathcal{O}_{l:t}^{\text{in}}\}$, and continuing from the previous expansion, we have:

$$\stackrel{(e)}{=} \underbrace{\{\mathcal{O}_{t+1:r}^{\text{in}}, e_{t+1:r-1}^{d+1} = \mathbf{0}, e_r^{d+1} = 1, \cdot x_{t+1}^{d+1} = j, U\}}_V \quad (5.51)$$

here at step (e), V is equivalent to $\{y_{t+1:r}, \tau_{t+1}^{d+1} = r\}$, hence, by Lemma 5.1, the following conditional independence holds:

$$V \perp\!\!\!\perp U \mid \{\cdot x_{t+1}^{d+1} = j\}$$

Next, when factorising the joint probability in Equation-(5.51), the term U will join $\{\cdot x_{t+1}^{d+1} = j\}$ in the conditioned term, so we arrange them in advance as follows:

$$\begin{aligned} \{\cdot x_{t+1}^{d+1} = j, U\} &= \{\cdot x_{t+1}^{d+1} = j, \cdot x_l^d = p, e_{l:t}^d = \mathbf{0}, x_t^{d+1} = i, \mathcal{O}_{l:t}^{\text{in}}\} \\ &\stackrel{(f)}{=} \{\cdot x_{t+1}^{d+1} = j, e_t^d = 0, x_t^{d+1} = i, x_t^d = p, \underbrace{e_{l:t-1}^d = \mathbf{0}, \cdot x_l^d = p, \mathcal{O}_{l:t}^{\text{in}}}_{W}\} \end{aligned} \quad (5.52)$$

where $W \triangleq \{e_{l:t-1}^d = \mathbf{0}, \cdot x_l^d = p, \mathcal{O}_{l:t}^{\text{in}}\}$, and in step (f) we split $e_{l:t}^d = \mathbf{0}$ into $\{e_{l:t-1}^d = \mathbf{0}, e_t^d = 0\}$ and add extra information $x_t^d = p$ (since $e_{l:t-1}^d = \mathbf{0}$, p stays the same from l to t). Note that W involves only variables prior to $t+1$ and thus by Lemma 5.2, in step (f), we obtain the following conditional independence:

$$\{\cdot x_{t+1}^{d+1} = j, e_t^d = 0\} \perp\!\!\!\perp W \mid \{x_t^{d+1} = i, x_t^d = p\}$$

We are now ready to factorise $\Pr(\text{AB}_{l:r}^{d,p}(j), \text{INSIDE})$. When a conditional independence $A \perp\!\!\!\perp C \mid B$ holds, we would write $\Pr(A, B, C) = \Pr(A \mid B, \Leftarrow) \Pr(B, C)$. Combining this form of factorisation with the above groupings, starting with Equation-(5.51), we write $\Pr(\text{AB}_{l:r}^{d,p}(j), \text{INSIDE})$ as:

$$\begin{aligned} &= \Pr(\underbrace{\{\mathcal{O}_{t+1:r}^{\text{in}}, e_{t+1:r-1}^{d+1} = \mathbf{0}, e_r^{d+1} = 1 \mid \cdot x_{t+1}^{d+1} = j, U\}}_{\Delta_{t+1:r}^{d+1,j}}) \Pr(\cdot x_{t+1}^{d+1} = j, U) \\ &= \Delta_{t+1:r}^{d+1,j} \Pr(\underbrace{\{\cdot x_{t+1}^{d+1} = j, e_t^d = 0 \mid x_t^{d+1} = i, x_t^d = p, W\}}_{A_{i,j}^{d,p}}) \Pr(x_t^{d+1} = i, x_t^d = p, W) \\ &= \Delta_{t+1:r}^{d+1,j} A_{i,j}^{d,p} \Pr(\underbrace{\{\mathcal{O}_{l:t}^{\text{in}}, x_t^{d+1} = i, e_{l:t-1}^d = \mathbf{0} \mid \cdot x_l^d = p\}}_{\alpha_{l:t}^{d,p}(i)}) \Pr(\cdot x_l^d = p) \\ &= \Delta_{t+1:r}^{d+1,j} A_{i,j}^{d,p} \alpha_{l:t}^{d,p}(i) \Pr(\cdot x_l^d = p) \end{aligned} \quad (5.53)$$

Substituting Equation-(5.53) into Equation-(5.50) and noting that the term $\Pr(\cdot x_t^d = p)$ is cancelled out, we have:

$$\xi_t^{d,p}(i, j) = \sum_{l;r} \lambda_{l;r}^{d,p}(j) \left[\alpha_{l:t}^{d,p}(i) A_{i,j}^{d,p} \Delta_{t+1:r}^{d+1,j} \right] \quad (5.54)$$

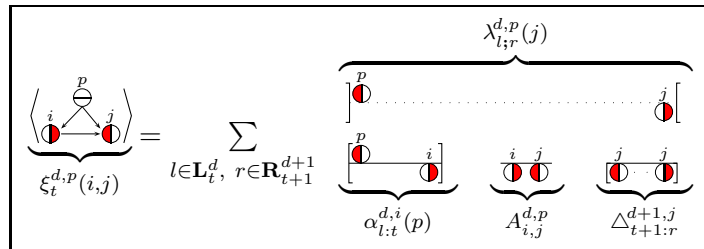
We have, so far, been treating the starting and ending time indices l and r ‘implicitly’ to avoid any complications during the explanation. Their ranges of summation is different for different hierarchic indices d . Let \mathbf{L}_t^d be the range of the values the starting time $\cdot \tau_t^d$ can take, and similarly, let \mathbf{R}_t^d be the range the ending time τ_t^d can take. Then, by the model assumption, we can easily establish:

$$\mathbf{L}_t^d = \begin{cases} 1 & \text{for } d = 1 \\ [1, t] & \text{for } 1 < d < D - 1 \\ t & \text{for } d = D - 1 \end{cases} \quad \mathbf{R}_t^d = \begin{cases} \text{undef} & \text{for } d = 1 \\ [t, T] & \text{for } 1 < d < D - 1 \\ t & \text{for } d = D - 1 \end{cases} \quad (5.55)$$

The final form of computation for $\xi_t^{d,p}(i, j)$ from Equation-(5.54), therefore, is given as:

$$\xi_t^{d,p}(i, j) = \sum_{l \in \mathbf{L}_t^d} \sum_{r \in \mathbf{R}_{t+1}^{d+1}} \lambda_{l;r}^{d,p}(j) \left[\alpha_{l:t}^{d,p}(i) A_{i,j}^{d,p} \Delta_{t+1:r}^{d+1,j} \right] \quad (5.56)$$

The process to derive Equation-(5.56) seems complicated at first, but indeed once the consistency in the definition of the auxiliary inside/outside variables is formed (Equations-(5.47a-5.47c)), the derivation is simple since the factorisation is guaranteed by the conditional independency properties in the DBN network. For example, using our diagrammatic tools (shown in Figure-(5-18)), Equation-(5.56) can be informally visualised and obtained directly from the diagrams as:



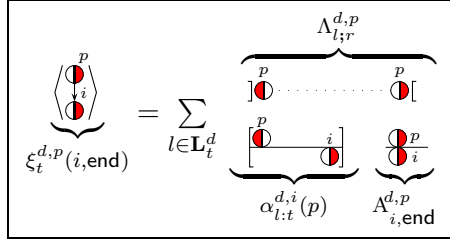
in which we sum over the starting time l of the parent-state p , and the ending time r of child-state j . Next the forms of inside/outside variables are easily identified from there to yield the formula in Equation-(5.56). We will, henceforth, discuss computational forms for the rest of the auxiliary variables in this diagrammatic ‘language’ and leave the formal proofs to Appendix B.

The horizontal probability of ‘going to end’ $\xi_t^{d,p}(i, \text{end})$ defined in Equation-(5.40b) is com-

puted as:

$$\xi_t^{d,p}(i, \text{end}) = \sum_{l \in \mathbf{L}_t^d} \left[\alpha_{l;t}^{d,p}(i) A_{i, \text{end}}^{d,p} \right] \Lambda_{l;t}^{d,p} \quad (5.57)$$

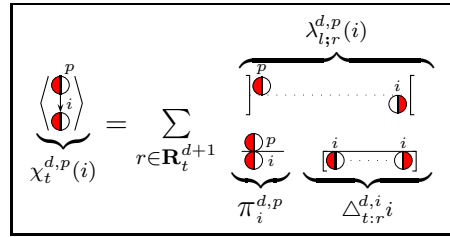
whose diagrammatic form is shown below, and in this diagram, we simply sum over the starting time l of the parent-state p .



5.5.2.3 Calculating the vertical-transition/emission probability variables

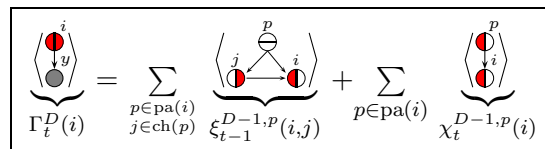
The vertical transition probability variable $\chi_t^{d,p}(i)$ in Equation-(5.41) is computed by summing over the ending time r of the child state i :

$$\chi_t^{d,p}(i) = \pi_i^{d,p} \left[\sum_{r \in \mathbf{R}_t^{d+1}} \lambda_{t;r}^{d,p}(i) \Delta_{t;r}^{d+1,i} \right] \quad (5.58)$$



The emission probability variable $\Gamma_t^D(i)$ is computed by summing over the parent p of i and considering the value of the end variable e_{t-1}^{D-1} . When $e_{t-1}^{D-1} = 1$, it reduces to the form of the vertical variable $\chi_t^{D-1,p}(i)$, when $e_{t-1}^{D-1} = 0$, we sum over the state j prior to i and this results in a form of horizontal variable $\xi_{t-1}^{D-1,p}(j, i)$:

$$\Gamma_t^D(i) = \sum_{p \in \text{pa}(i)} \left(\sum_{j \in \text{ch}(p)} \xi_{t-1}^{D-1,p}(j, i) + \chi_t^{D-1,p}(i) \right) \quad (5.59)$$



5.5.3 Computation of Other Auxiliary Variables

In this subsection, we show the computation of the inside/outside auxiliary variables defined in Section 5.5.2.1. In particular, we show the diagrammatic intuition behind each formula. Their formal proofs, however, are provided the Appendix B.

5.5.3.1 Calculating the asymmetric forward inside variable

The diagrammatic visualisation of the inside and outside variables in Figure-(5-18) gives us a useful tool for summarising the recursive relations between these variables. Let us first consider the computation of $\alpha_{l:r}^{d,p}(i)$. Let t be the starting time of the child i that ends at r . We can then break the diagram for $\alpha_{l:r}^{d,p}(i)$ into two sub-diagrams: one corresponding to $\alpha_{l:t}^{d,p}(i)$ and another corresponding to $\Delta_{t:r}^{d+1,i}$:

$$\underbrace{\begin{array}{|c|} \hline \text{[} \bullet^p \text{ --- } \bullet^i \text{]} \\ \hline \end{array}}_{\alpha_{l:r}^{d,p}(i)} = \sum_{l \leq t \leq r} \underbrace{\begin{array}{|c|} \hline \text{[} \bullet^p \text{] } \bullet^i \\ \hline \end{array}}_{\alpha_{l:t}^{d,p}(i)} \times \underbrace{\begin{array}{|c|} \hline \bullet^i \text{ --- } \bullet^i \\ \hline \end{array}}_{\Delta_{t:r}^{d+1,i}}$$

The conditional independence property in the HHMM (5.1) then allows us to simply take the product of the two parts. Summing over the unknown time t then gives us the precise recursion formula for $\alpha_{l:r}^{d,p}(i)$ as:

$$\alpha_{l:r}^{d,p}(i) = \sum_{t=l}^r \alpha_{l:t}^{d,p}(i) \Delta_{t:r}^{d+1,i} \quad (5.60)$$

5.5.3.2 Calculating the started-forward variable

Recall the definition from Equation-(5.48):

$$\alpha_{l:r}^{d,p}(i) \triangleq \Pr(y_{l:r-1}, \cdot x_r^{d+1} = i, e_{l:r-1}^d = \mathbf{0} \mid \cdot x_l^d = p)$$

When the starting time of the parent state p is the same as of its child i , ie: $l = r = t$, the terms $y_{l:r-1}$ and $e_{l:r-1}^d$ disappear, therefore:

$$\alpha_{t:t}^{d,p}(i) \triangleq \Pr(\cdot x_t^{d+1} = i \mid \cdot x_t^d = p) = \pi_i^{d,p} \quad (5.61a)$$

Algorithm 5.3 Pseudo codes to compute the set of inside variables

Input: A discrete HHMM specified by $\langle \zeta, \theta, \mathcal{Y} \rangle$.

Output: Calculating $\alpha_{l:r}^{d,p}(i)$, $\Delta_{l:r}^{d,i}$, and $\underline{\alpha}_{l:r}^{d,p}(i)$ for $d = 1, \dots, D$, $p \in \mathcal{S}^d$, $i \in \text{ch}(p)$, and $1 \leq l \leq r \leq T$.

Initialization. (at the bottom level $d = D$)

For l from 1 to T , and $i \in \mathcal{S}^D$ Do:

Compute $\Delta_{l:i}^{D,i}$ from Equation-(5.62a)

EndFor

Recursion.

For $d = D - 1$ to 1 (bottom-up) Do

For $l = 1$ to T (left-to-right) Do

For $r = l$ to T , and $p \in \mathcal{S}^d$, $i \in \text{ch}(p)$ Do

For $t = l$ to r Do

Calculate $\underline{\alpha}_{l:t}^{d,p}(i)$ from Equation-(5.61)

Calculate $\Delta_{t:r}^{d+1,i}$ from Equation-(5.62b)

EndFor

Calculate $\alpha_{l:r}^{d,p}(i)$ from Equation-(5.60)

EndFor

EndFor

EndFor

For r strictly greater than l , summing over the previous child j , we obtain the relation:

$$\underline{\alpha}_{l:r}^{d,p}(i) = \sum_{j \in \text{ch}(d,p)} \alpha_{l:r-1}^{d,p}(j) A_{j,i}^{d,p} \quad (5.61b)$$

$$\boxed{\underbrace{\left[\begin{array}{c} p \\ \text{---} \\ i \end{array} \right]}_{\underline{\alpha}_{l:r}^{d,p}(i)}} = \sum_{j \in \text{ch}(d,p)} \underbrace{\left[\begin{array}{c} p \\ \text{---} \\ j \end{array} \right]}_{\alpha_{l:r-1}^{d,p}(j)} \times \underbrace{\left[\begin{array}{c} j \\ \text{---} \\ i \end{array} \right]}_{A_{j,i}^{d,p}}$$

5.5.3.3 Calculating the symmetric inside variable

If $d = D$ then r must equal l to be consistent with the definition of the model. In this case, we have:

$$\Delta_{r:r}^{D,i} \triangleq \Pr(y_r \mid x_r^D = i) = B_{y_r|i} \quad (5.62a)$$

For $d > D$, we simply sum over the child s of state i at time r :

$$\Delta_{l:r}^{d,i} = \sum_{s \in \text{ch}(i)} \alpha_{l:r}^{d,i}(s) A_{s,\text{end}}^{d,i} \quad (5.62b)$$

$$\underbrace{\left[\begin{array}{c} i \\ \vdots \\ i \end{array} \right]}_{\Delta_{l:r}^{d,i}} = \sum_{s \in \text{ch}(d,i)} \underbrace{\left[\begin{array}{c} i \\ \vdots \\ s \end{array} \right]}_{\alpha_{l:r}^{d,i}(s)} \times \underbrace{\left[\begin{array}{c} i \\ \vdots \\ s \end{array} \right]}_{A_{s,\text{end}}^{d,i}}$$

The algorithm used to calculate the set of inside variables is outlined in Algorithm 5.3.

5.5.3.4 Calculating the symmetric/asymmetric outside variables

Recall the definition of these two outside variables from Equation-(5.47c) and Equation-(5.49) as:

$$\begin{aligned} \lambda_{l:r}^{d,p}(i) &\triangleq \Pr(\mathcal{O}_{l:r}^{\text{out}} \mid \cdot x_l^d = p, e_{l:r-1}^d = \mathbf{0}, x_r^{d+1} = i) \Pr(\cdot x_l^d = p) \\ \Lambda_{l:r}^{d,p} &\triangleq \Pr(\mathcal{O}_{l:r}^{\text{out}} \mid \cdot x_l^d = p, e_{l:r-1}^d = \mathbf{0}, e_r^d = 1) \Pr(\cdot x_l^d = p) \end{aligned}$$

Initialisation at the root $d = 1$. From direct definition of these variables (Equation-(5.49), Equation-(5.47c)), we have:

$$\Lambda_{1:T}^{1,1} \triangleq \Pr(e_T^1 = 1 \mid \cdot x_1^1 = 1, e_{1:T-1}^1 = \mathbf{0}, e_T^1 = 1) \Pr(\cdot x_1^1 = 1) = 1 \quad (5.63a)$$

$$\lambda_{1:T}^{1,1}(i) \triangleq \Pr(e_T^1 = 1 \mid \cdot x_1^1 = 1, e_{1:T-1}^1 = \mathbf{0}, x_T^2 = i, e_T^1 = 1) \Pr(\cdot x_1^1 = 1) = A_{i,\text{end}}^{1,1} \quad (5.63b)$$

$$\lambda_{1:r}^{1,1}(i) = \sum_{t \in \mathbf{R}_r^2} \sum_{j \in \mathcal{S}^2} \lambda_{1;t}^{1,1}(j) \Delta_{r+1:t}^{2,j} A_{i,j}^{1,1} \quad \text{for } r < T \quad (5.63c)$$

where in Equation-(5.63c), the summation is over the state j that follows i , and the ending time t of j . This gives the form of factorisation as follows:

$$\underbrace{\left[\begin{array}{c} 1 \\ \vdots \\ i \end{array} \right]}_{\lambda_{1:r}^{1,1}(i)} = \sum_{\substack{t \in \mathbf{R}_r^2 \\ j \in \mathcal{S}^2}} \underbrace{\left[\begin{array}{c} 1 \\ \vdots \\ i \\ \vdots \\ j \end{array} \right]}_{\lambda_{1;t}^{1,1}(j)} \underbrace{\left[\begin{array}{c} i \\ \vdots \\ j \end{array} \right]}_{A_{i,j}^{1,1}} \underbrace{\left[\begin{array}{c} j \\ \vdots \\ j \end{array} \right]}_{\Delta_{r+1:t}^{2,j}}$$

Induction step when $d < D$. The symmetric outside variable $\Lambda_{l:r}^{d,p}$ is computed by simply summing over the parent q of p and its starting time t .

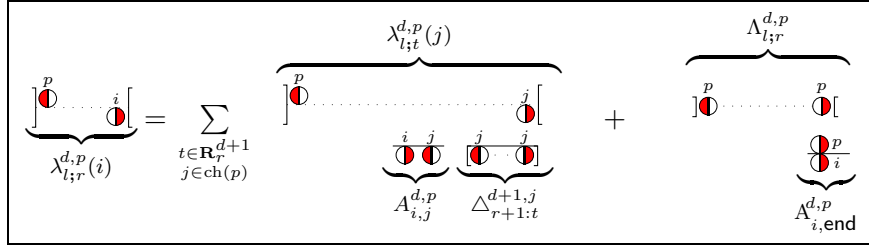
$$\Lambda_{l:r}^{d,p} = \sum_{q \in \text{pa}(p)} \sum_{t \in \mathbf{L}_l^{d-1}} \alpha_{t:l}^{d-1,q}(p) \lambda_{t;r}^{d-1,q}(p) \quad (5.64)$$

$$\underbrace{\left[\begin{array}{c} p \\ \vdots \\ p \end{array} \right]}_{\Lambda_{l:r}^{d,1}} = \sum_{\substack{q \in \text{pa}(p) \\ t \in \mathbf{L}_l^{d-1}}} \underbrace{\left[\begin{array}{c} q \\ \vdots \\ p \end{array} \right]}_{\lambda_{t;r}^{d-1,q}(p)} \underbrace{\left[\begin{array}{c} q \\ \vdots \\ p \end{array} \right]}_{\alpha_{t:l}^{d,q}(p)}$$

The asymmetric outside variable $\lambda_{l;r}^{d,p}(i)$ is computed by breaking it down into two cases when considering e_r^d . When $e_r^d = 1$, the parent-state p also terminates at r , therefore, the symmetric form $\Lambda_{l;r}^{d,p}$ is recovered (RHS). When $e_r^d = 0$, we sum over the j state starting at time $r + 1$ and ending time at t to achieve the recursive asymmetric form $\lambda_{l;t}^{d,p}(j)$ (shown on LHS).

$$\lambda_{l;r}^{d,p}(i) = \sum_{t \in \mathbf{R}_r^{d+1}} \sum_{j \in \text{ch}(p)} \lambda_{l;t}^{d,p}(j) A_{i,j}^{d,p} \Delta_{r+1:t}^{d+1,j} + \Lambda_{l;r}^{d,p} A_{i,\text{end}}^{d,p} \quad (5.65)$$

The diagrammatic form for this calculation is given as:



The proofs for Equation-(5.63c), Equation-(5.64), and Equation-(5.65) are formally given in Appendix B. The pseudocode to compute the set of outside variables is provided in Algorithm 5.4.

Algorithm 5.4 Pseudo codes to compute the set of outside variables

Input: A discrete HHMM specified by $\langle \zeta, \theta, \mathcal{Y} \rangle$. Assuming Algorithm 5.3 has run to compute $\alpha_{l;r}^{d,p}(i)$, $\Delta_{l;r}^{d,i}$, and $\alpha_{l;r}^{d,p}(i)$.

Output: Calculating $\lambda_{l;r}^{d,p}(i)$ and $\Lambda_{l;r}^{d,p}$ for $d = 1, \dots, D$, $p \in \mathcal{S}^d$, $i \in \text{ch}(p)$, and $1 \leq l \leq r \leq T$.

Initialization. (at the root level $d = 1$)

For $1 \leq r \leq T$ and $i \in \mathcal{S}^2$ Do

 Calculate $\Lambda_{1;T}^{1,1}$, $\lambda_{1;T}^{1,1}(i)$, $\lambda_{1;r}^{1,1}(i)$ from Equation-(5.63).

EndFor

Recursion. ($d > 1$)

For $d = 2$ to $D - 1$ (top-down) Do

 For $l = 1$ to T Do

 For $r = T$ downto l , and $p \in \mathcal{S}^d$, $i \in \text{ch}(p)$ Do

 Calculate $\lambda_{l;r}^{d,p}(i)$, $\Lambda_{l;r}^{d,p}$ from Equation-(5.64), and Equation-(5.65).

 EndFor

 EndFor

EndFor

In summary, the Asymmetric Inside-Outside algorithm outlined in Algorithm 5.5 is thus consisting of Algorithm 5.3 to compute the set of inside variables and Algorithm 5.4 to compute the set of outside variables. The results from these two algorithms are then used to calculate the horizontal, vertical, and emission probability variables, which provide the expected sufficient statistics needed for the learning.

Algorithm 5.5 The Asymmetric Inside-Outside Algorithm for the HHMM

Input: A discrete HHMM specified by $\langle \zeta, \theta, \mathcal{Y} \rangle$.

- Run Algorithm 5.3 to calculate $\alpha_{l;r}^{d,p}(i)$, $\Delta_{l;r}^{d,i}$, and $\underline{\alpha}_{l;r}^{d,p}(i)$.
- Run Algorithm 5.4 to calculate $\lambda_{l;r}^{d,p}(i)$ and $\Lambda_{l;r}^{d,p}$.
- Compute $\xi_t^{d,p}(i, j)$ and $\xi_t^{d,p}(i, \text{end})$ from Equation-(5.56) and Equation-(5.57).
- Compute $\chi_t^{d,p}(i)$ from Equation-(5.58).
- Compute $\Gamma_t^D(i)$ from Equation-(5.59).

Output: All necessary values needed for the inference and learning in the HHMM, including the set of inside/outside variables, the horizontal, vertical, and emission probability variables.

5.6 Computing the Likelihood

We sum over the asymmetric inside variable at the top level to yield:

$$\begin{aligned} \sum_{i \in \mathcal{S}^2} \alpha_{1:T}^{1,1}(i) &\stackrel{(a)}{=} \sum_{i \in \mathcal{S}^2} \Pr(\mathcal{O}, x_T^2 = i, e_{1:T-1}^1 = \mathbf{0} \mid \cdot x_1^1 = 1) \\ &= \Pr(\mathcal{O}, e_T^2 = 1 \mid \cdot x_1^1 = 1) = \Pr(\mathcal{O}, e_T^2 = 1) \end{aligned}$$

where in step (a) we expand $\{x_T^2 = i\}$ into $\{x_T^2 = i, e_T^2 = 1\}$ and summing over i results in $e_T^2 = 1$. Unfortunately, this is not enough to give us the likelihood $\Pr(\mathcal{O})$. The missing term is $\Pr(\mathcal{O}, e_T^2 = 0)$, which cannot be computed with the defined set of inside and outside variables. To solve this problem, we appeal to a new form of the asymmetric inside variable called the ‘*continuing*’- α , denoted by α :

$$\alpha_{l;r}^{d,p}(i) \triangleq \Pr(\mathcal{O}_{l;r}^{\text{in}}, x_r^{\circ d+1} = i, e_{l;r-1}^d = \mathbf{0} \mid \cdot x_l^d = p) \quad (5.66)$$

where we write $x_r^{\circ d+1} = i$, in contrast to $x_r^{d+1} = i$, to represent the event $\{x_r^{d+1} = i, e_r^{d+1} = 0\}$, ie: state i does not end at r . With this newly defined variable, it then follows that:

$$\begin{aligned} \sum_{i \in \mathcal{S}^2} \left(\alpha_{1:T}^{1,1}(i) + \alpha_{1:T}^{\circ 1,1}(i) \right) &= \sum_{i \in \mathcal{S}^2} \Pr(\mathcal{O}, x_T^2 = i, e_T^2 = 1, e_{1:T-1}^1 = \mathbf{0} \mid \cdot x_1^1 = 1) \\ &\quad + \sum_{i \in \mathcal{S}^2} \Pr(\mathcal{O}, x_T^2 = i, e_T^2 = 0, e_{1:T-1}^1 = \mathbf{0} \mid \cdot x_1^1 = 1) \\ &= \Pr(\mathcal{O}, e_T^2 = 1) + \Pr(\mathcal{O}, e_T^2 = 0) = \Pr(\mathcal{O}) \end{aligned} \quad (5.67)$$

which is the required likelihood. The remaining issue is the computation of the $\alpha_{l;r}^{d,p}(i)$ variable. Turning to a similar decomposition form for $\alpha_{l;r}^{d,p}(i)$, we sum over the starting time l of the child state i and decompose $\alpha_{l;r}^{d,p}(i)$ as follows:

$$\underbrace{\left[\begin{array}{c} p \\ \bullet \cdots \ominus \\ i \end{array} \right]}_{\alpha_{l:r}^{d,p}(i)} = \sum_{l \leq t \leq r} \underbrace{\left[\begin{array}{c} p \\ \bullet \cdots \bullet \\ i \end{array} \right]}_{\alpha_{l:t}^{d,p}(i)} \times \underbrace{\left[\begin{array}{c} i \\ \bullet \cdots \ominus \\ i \end{array} \right]}_{\Delta_{\ominus t:r}^{d+1,i}}$$

which gives rise to the precise computation form for $\alpha_{l:r}^{d,p}(i)$ as:

$$\alpha_{l:r}^{d,p}(i) = \sum_{t=l}^r \alpha_{l:t}^{d,p}(i) \Delta_{\ominus t:r}^{d+1,i} \quad (5.68)$$

This computation, however, involves a new variant of Δ , which we call the symmetric ‘*continuing*’-inside variable Δ_{\ominus} and define it as:

$$\Delta_{\ominus l:r}^{d,i} \triangleq \Pr(y_{l:r}, e_{l:r}^d = \mathbf{0} \mid \cdot x_l^d = i) \quad (5.69)$$

Again, with the diagrammatic tools, this variable can be graphically decomposed and computed as:

$$\Delta_{\ominus l:r}^{d,i} = \begin{cases} 0 & \text{if } d = D \\ \sum_{s \in \text{ch}(i)} \alpha_{l:r}^{d,i}(s) (1 - A_{s,\text{end}}^{d,i}) + \alpha_{l:r}^{d,i}(i) & \text{if } d < D \end{cases} \quad (5.70)$$

$$\underbrace{\left[\begin{array}{c} i \\ \bullet \cdots \ominus \\ i \end{array} \right]}_{\Delta_{\ominus l:r}^{d,i}} = \sum_{s \in \text{ch}(d,i)} \underbrace{\left[\begin{array}{c} i \\ \bullet \cdots \bullet \\ s \end{array} \right]}_{\alpha_{l:r}^{d,i}(s)} \times \underbrace{\left[\begin{array}{c} \ominus \\ s \end{array} \right]}_{1 - A_{s,\text{end}}^{d,i}}$$

Proofs for Equation-(5.68) and Equation-(5.70) are formally given in Appendix B. The algorithm to compute $\alpha_{l:r}^{d,p}(i)$ and $\Delta_{\ominus l:r}^{d,i}$ is identical to Algorithm 5.3 except that $\{\alpha_{l:r}^{d,p}(i), \Delta_{l:r}^{d,i}\}$ are replaced by their ‘unfinished’ counterparts $\{\alpha_{l:r}^{d,p}(i), \Delta_{\ominus l:r}^{d,i}\}$ provided in Equation-(5.68) and Equation-(5.70).

5.7 Complexity Analysis and Some Numerical Results

In this section we provide a complexity analysis for the AIO-algorithm and provide some numerical results.

5.7.1 Complexity of the AIO-algorithm

Let S be the maximum number of states at a particular level in the hierarchy, ie:

$$S = \left| \mathcal{S}^{d^*} \right| \text{ where } d^* = \operatorname{argmax}_d \left| \mathcal{S}^d \right|,$$

and let b be the branching factor which is the maximum number of children for each state. The complexity for Algorithm 5.3 to compute the set of inside variables is $O(T^3 S b^2 D)$ due to the following facts:

- T^3 arises by three loops: (1) over the left-time index l , (2) over the right-time index r , and (3) over t from l to r .
- D is accounted for the loop over d .
- S is accounted for the loop over p .
- b^2 is accounted for the loop over i and j (both children of p).

Similar analysis shows that Algorithm 5.4 used to calculate the set of outside variables has the same complexity, and thus the overall complexity for the AIO-algorithm is $O(T^3 S b^2 D)$, which is the same as the complexity of the FST algorithm provided in the original paper (Fine *et al.*, 1998) for the HHMM with non-shared structures.

5.7.2 Some Numerical Results

As a proof of concept, we provide here some numerical results. All of the algorithms for the HHMM developed in this thesis were implemented in Matlab and run on the PC configured with 3.0Gz CPU with 1Gb of memory.

Experiment 1

In a report, Schlick (Schlick, 2000) provides a detailed implementation of the FST method, and we use the HHMM reported in (Schlick, 2000) in this experiment. The topology and initial parameters are shown in Figure-(5-19). This HHMM has a depth $D = 3$ and the size of the observation alphabets is three. The training data is the observation sequence:

$$\mathcal{O} = (1 \ 3 \ 2 \ 3 \ 2 \ 2 \ 3 \ 1 \ 1 \ 2), \quad T = 10$$

and with the assumption that the root state ends at time T . The log-likelihood curve after 20 iterations during the EM learning is shown in Figure-(5-20). The results of estimated

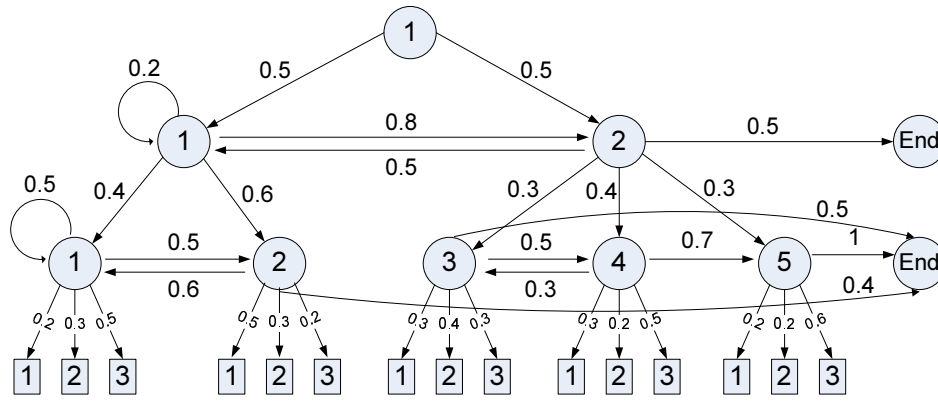


Figure 5-19: The HHMM used in the report of (Schlick, 2000) and in our experiment.

parameters after 20 iterations reported are identical to the results in the report of (Schlick, 2000) with some very small variances due to the difference in the Dirichlet’s priors¹¹ added during the training. As an example, Table-(5.2) shows the initial parameters at the root

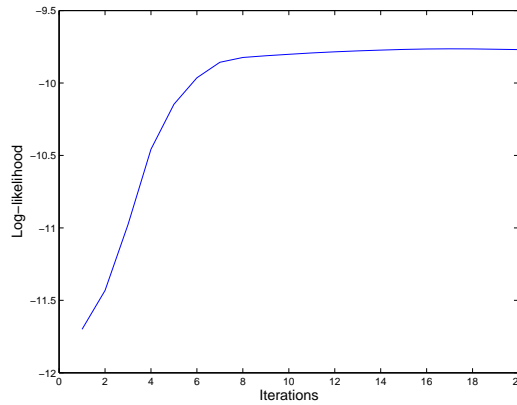


Figure 5-20: The training curve after 20 iterations.

level, estimated results from (Schlick, 2000) and from our experiment after 20 iterations.

Initial	Schlick (2000)	Our Exp.
$\pi^1 = (0.5, 0.5)$	$\hat{\pi}^1 = (0.9999, 0.0001)$	$\hat{\pi}^1 = (0.9999, 0.0001)$
$A^1 = \begin{pmatrix} 0.2 & 0.8 & 0.0 \\ 0.5 & 0.0 & 0.5 \end{pmatrix}$	$\hat{A}^1 = \begin{pmatrix} 0.0001 & 0.9999 & 0.0000 \\ 0.0001 & 0.0000 & 0.9999 \end{pmatrix}$	$\hat{A}^1 = \begin{pmatrix} 0.0011 & 0.9988 & 0.0000 \\ 0.0001 & 0.0001 & 0.9998 \end{pmatrix}$

Table 5.2: The initial parameters at the root level, and their estimated results from (Schlick, 2000) and from our experiment.

Experiment 2

To demonstrate the advantage of our proposed method in comparison with the linear

¹¹In practice, the expected sufficient statistics for a parameter is initialised with a very small number, *not* from 0. This is usually referred to as adding a Dirichlet’s prior.

method (Murphy and Paskin, 2001) we construct two different topologies for testing. The first topology has four levels containing three states each (a total of 12 states excluding the top-level dummy state); the topology is fully connected, that is a state at each level is the parent of all the states at the level below. The second topology is constructed in a similar manner except it has five levels (and thus has 15 states). These topologies are similar to the one plotted in Figure-(5-5) in our earlier discussion. Figure-(5-21) shows

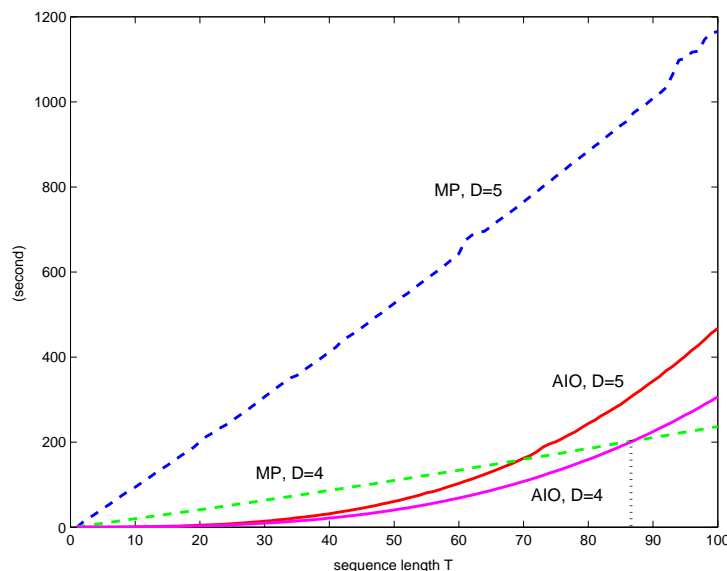


Figure 5-21: Computation time of MP method (Murphy and Paskin, 2001) versus our proposed Asymmetric Inside-Outside (AIO) method.

the computational time of the forward pass in one EM iteration using the two methods on two different topologies. While our method exhibits $O(T^3)$, it scales linearly when we increase the depth D of the model. Adding an extra level means that our method only has to deal with an extra three states, while the MP method (Murphy and Paskin, 2001) has to deal with three times the number of states due to its conversion to a tree structure (and thus essentially exponential in D).

5.8 Numerical Underflow and the Scaling Algorithm

It is well-known that as the length of the observation sequence increases, naive implementations of the HMMs will run into numerical underflow problems. To combat this problem, a method for numerical scaling in the flat Hidden Markov Model has been derived (Rabiner, 1989). It is thus imperative to address the same issues in the HHMM for it to be deployed in practice.

The set of variables defined in Equation-(5.43) when computing the sufficient statistics for θ reveals the source of the problem: both the numerators and denominators are joint probabilities of order $O(T)$ that quickly go to zero as T becomes large. We can rewrite the equations to compute the expected sufficient statistics for θ as:

$$\begin{aligned} \langle A_{i,j}^{d,p} \rangle &= \sum_{k=1}^K \sum_{t=2}^T \tilde{\xi}_t^{d,p}(i,j) & \langle \pi_i^{d,p} \rangle &= \sum_{k=1}^K \sum_{t=1}^T \tilde{\chi}_t^{d,p}(i) \\ \langle A_{i,\text{end}}^{d,p} \rangle &= \sum_{k=1}^K \sum_{t=1}^T \tilde{\xi}_t^{d,p}(i,\text{end}) & \langle B_{v|i} \rangle &= \sum_{k=1}^K \sum_{t=1}^T \Pr(y_t=v \mid x_t^D=i, \mathcal{O}) \tilde{\Gamma}_t^D(i) \end{aligned}$$

where the newly introduced set of scaled variables are defined as:

$$\tilde{\xi}_t^{d,p}(i,j) \triangleq \frac{\xi_t^{d,p}(i,j)}{\Pr(\mathcal{O})} = \Pr(x_t^d=p, x_t^{d+1}=j, x_{t-1}^{d+1}=i, e_{t-1}^{d:d+1}=01 \mid \mathcal{O}) \quad (5.71a)$$

$$\tilde{\xi}_t^{d,q}(i,\text{end}) \triangleq \frac{\xi_t^{d,q}(i,\text{end})}{\Pr(\mathcal{O})} = \Pr(x_t^d=p, x_t^{d+1}=i, e_t^{d:d+1}=11 \mid \mathcal{O}) \quad (5.71b)$$

$$\tilde{\chi}_t^{d,p}(i) \triangleq \frac{\chi_t^{d,p}(i)}{\Pr(\mathcal{O})} = \Pr(x_t^d=p, x_t^{d+1}=i, x_{t-1}^{d:d+1}=11 \mid \mathcal{O}) \quad (5.71c)$$

$$\tilde{\Gamma}_t^D(i) \triangleq \frac{\Gamma_t^D(i)}{\Pr(\mathcal{O})} = \Pr(x_t^D=i \mid \mathcal{O}) \quad (5.71d)$$

It is desirable to compute this set of scaled variables directly. To do so, we introduce the *scaling factor* at time t as follows:

$$\begin{aligned} \varphi_1 &\triangleq \Pr(y_1) \\ \varphi_t &\triangleq \Pr(y_t, e_{t-1}^1=0 \mid y_{1:t-1}, e_{1:t-2}^1=\mathbf{0}) \quad \text{for } t \geq 2 \end{aligned} \quad (5.72)$$

With this definition, a product of t consecutive scaling factors results in a simple joint probability of observations from 1 to t , in particular, the likelihood can be obtained as:

$$\begin{aligned} \prod_{k=1}^T \varphi_k &= \Pr(o_1) \Pr(o_2 \mid o_1) \dots \Pr(o_T \mid o_{T-1}) = \Pr(o_1, o_2, \dots, o_T) \\ &= \Pr(\mathcal{O}) \end{aligned} \quad (5.73)$$

We proceed to scale up each inside and outside variable *by the set of scaling factors in the*

observation range, that is:

$$\tilde{\alpha}_{l;r}^{d,p}(i) \triangleq \alpha_{l;r}^{d,p}(i) \left(\prod_{k=l}^r \varphi_k \right) \quad \tilde{\alpha}_{\circ l;r}^{d,p}(i) \triangleq \alpha_{\circ l;r}^{d,p}(i) \left(\prod_{k=l}^r \varphi_k \right) \quad (5.74a)$$

$$\tilde{\Delta}_{l;r}^{d,i} \triangleq \Delta_{l;r}^{d,i} \left(\prod_{k=l}^r \varphi_k \right) \quad \tilde{\Delta}_{\circ l;r}^{d,i} \triangleq \Delta_{\circ l;r}^{d,i} \left(\prod_{k=l}^r \varphi_k \right) \quad (5.74b)$$

$$\tilde{\tilde{\alpha}}_{l;r}^{d,p}(i) \triangleq \tilde{\alpha}_{l;r}^{d,p}(i) \left(\prod_{k=l}^r \varphi_k \right) \quad (5.74c)$$

$$\tilde{\lambda}_{l;r}^{d,p}(i) \triangleq \lambda_{l;r}^{d,p}(i) \left(\prod_{k=1}^{l-1} \varphi_k \prod_{k=r+1}^T \varphi_k \right) \quad \tilde{\Lambda}_{l;r}^{d,i} \triangleq \Lambda_{l;r}^{d,i} \left(\prod_{k=1}^{l-1} \varphi_k \prod_{t=r+1}^T \varphi_t \right) \quad (5.74d)$$

Since each of the scaled variables effectively carries the scaled factors within its range of observations, their product would carry the product of all the scaled factors. Thus, the set of equations to calculate the sufficient statistics variables $\left\{ \xi_t^{d,p}(i, j), \xi_t^{d,p}(i, \text{end}), \chi_t^{d,p}(i), \Gamma_t^D(i) \right\}$ in the unscaled case computed in Equations-(5.56–5.59) still holds if we replace each of the variables by its scaled counterpart. That is:

$$\tilde{\xi}_t^{d,p}(i, j) = \sum_{l \in \mathbf{L}_t^d} \sum_{r \in \mathbf{R}_{t+1}^{d+1}} \tilde{\lambda}_{l;r}^{d,p}(j) \left[\tilde{\alpha}_{l;t}^{d,p}(i) A_{i,j}^{d,p} \tilde{\Delta}_{t+1:r}^{d+1,j} \right] \quad (5.75a)$$

$$\tilde{\xi}_t^{d,p}(i, \text{end}) = \sum_{l \in \mathbf{L}_t^d} \left[\tilde{\alpha}_{l;t}^{d,p}(i) A_{i,\text{end}}^{d,p} \right] \tilde{\Lambda}_{l;t}^{d,p} \quad (5.75b)$$

$$\tilde{\chi}_t^{d,p}(i) = \pi_i^{d,p} \left[\sum_{r \in \mathbf{R}_t^{d+1}} \tilde{\lambda}_{t;r}^{d,p}(i) \tilde{\Delta}_{t:r}^{d+1,i} \right] \quad (5.75c)$$

$$\tilde{\Gamma}_t^D(i) = \sum_{p \in \text{pa}(i)} \left[\sum_{j \in \text{ch}(p)} \tilde{\xi}_{t-1}^{D-1,p}(j, i) + \tilde{\chi}_t^{D-1,p}(i) \right] \quad (5.75d)$$

The correctness of Equation-(5.75a), for example, can be formally verified by using the definitions of scaled variables:

$$\begin{aligned} \tilde{\lambda}_{l;r}^{d,p}(j) \left[\tilde{\alpha}_{l;t}^{d,p}(i) A_{i,j}^{d,p} \tilde{\Delta}_{t+1:r}^{d+1,j} \right] &= \frac{\lambda_{l;r}^{d,p}(j)}{\prod_{k=1}^{l-1} \varphi_k \prod_{k=r+1}^T \varphi_k} \left[\frac{\alpha_{l;t}^{d,p}(i)}{\prod_{k=l}^t \varphi_k} A_{i,j}^{d,p} \frac{\Delta_{t+1:r}^{d+1,j}}{\prod_{k=r+1}^r \varphi_k} \right] \\ &= \frac{\lambda_{l;r}^{d,p}(j) \left[\alpha_{l;t}^{d,p}(i) A_{i,j}^{d,p} \Delta_{t+1:r}^{d+1,j} \right]}{\prod_{k=1}^T \varphi_k} \end{aligned}$$

Summing over l and r on both sides of the above equations and noting that $\prod_{k=1}^T \varphi_k = \text{Pr}(\mathcal{O})$, the RHS becomes $\xi_t^{d,p}(i, j) / \text{Pr}(\mathcal{O}) \triangleq \tilde{\xi}_t^{d,p}(i, j)$ and thus Equation-(5.75a) is proved. Equations 5.75b and 5.58 can be verified similarly.

We now show how the auxiliary scaled variables defined in Equation-(5.74) can be computed via dynamic programming in a bottom-up, and left-right fashion. We first discuss

the group of inside variables $\left\{ \tilde{\alpha}, \tilde{\alpha}_{\phi}, \tilde{\Delta}, \tilde{\Delta}_{\phi}, \tilde{\alpha} \right\}$.

5.8.1 Calculating the scaled inside variables

We provide the calculation of these variables in an inductive way. Let us start with the recursive hypothesis that all the scaled inside variables have been computed up to time $r - 1$ at all level d . The objective is to compute these variables at time r using induction.

Knowing all information prior to r allows us to simply apply the recursion formulas to calculate the scaled variables with right index r *provided that we can compute the scaling factor φ_r at time r* . For example, when computing α by their scaled counterparts, we have:

$$\tilde{\alpha}_{l:r}^{d,p}(i) = \sum_{t=l}^r \tilde{\alpha}_{l:t}^{d,p}(i) \tilde{\Delta}_{t:r}^{d+1,i} \quad (5.76)$$

We note that in this equation the started-forward scaled variable $\tilde{\alpha}_{l:t}^{d,p}(i)$ only accounts for the observation from l to $t - 1$ and is, therefore, already computed by the recursive hypothesis¹². The problematic term is $\tilde{\Delta}_{t:r}^{d+1,i}$ as it involves information at time r , which can be rewritten as:

$$\tilde{\Delta}_{t:r}^{d+1,i} \triangleq \frac{\Delta_{t:r}^{d+1,i}}{\prod_{k=t}^r \varphi_r} = \begin{cases} \frac{B_{yr|i}}{\varphi_r} & \text{if } d+1 = D, t = r \\ \sum_{s \in \text{ch}(i)} \tilde{\alpha}_{t:r}^{d+1,i}(s) A_{s,\text{end}}^{d+2,i} & \text{if } d+1 < D \end{cases} \quad (5.77)$$

Clearly, if the scaling factor φ_r at time r is known, then $\tilde{\Delta}_{r:r}^{d+1,i}$ is computable and, therefore, $\tilde{\alpha}$ and $\tilde{\Delta}$ in Equation-(5.77)-Equation-(5.76) are computable in a bottom-up fashion through the following updates¹³:

$$\tilde{\Delta}_{r:r}^{D,i} \longrightarrow \tilde{\alpha}_{l:r}^{D-1,p}(i) \longrightarrow \tilde{\Delta}_{l:r}^{D-1,i} \longrightarrow \tilde{\alpha}_{l:r}^{D-2,p}(i) \longrightarrow \dots$$

Similar lines of arguments are also valid for the set of continuing inside variables $\left\{ \tilde{\alpha}_{\phi}, \tilde{\Delta}_{\phi} \right\}$.

In the next section, we detail the computation of the scaling variable φ_r .

¹²A trace back to its computation form in Equation-(5.61) can easily verify this fact.

¹³Where we recall that by the model assumption a state at production level D starts and ends in a single time slice therefore calculation for $\tilde{\alpha}$ at $D - 1$ is simplified to:

$$\tilde{\alpha}_{l:r}^{D-1,p}(i) = \tilde{\alpha}_{l:r}^{d,p}(i) \tilde{\Delta}_{r:r}^{d+1,i}$$

5.8.1.1 Calculating the scaling factor φ_r

We propose a two-stage procedure to calculate φ_r . First, a set of *partially* scaled variables, each of which effectively carries all necessary scaling factors, except at the very end φ_r , are calculated. Next, based on these partially scaled variables, φ_r is obtained by marginalising out necessary variables.

The partially scaled variables are defined from the scaled variables by a multiplication factor of φ_r :

$$\ddot{\alpha}_{l:r}^{d,p}(i) \triangleq \tilde{\alpha}_{l:r}^{d,p}(i) \times \varphi_r \qquad \ddot{\Delta}_{l:r}^{d,i} \triangleq \tilde{\Delta}_{l:r}^{d,i} \times \varphi_r \qquad (5.78a)$$

$$\ddot{\alpha}_{\circ:l:r}^{d,p}(i) \triangleq \tilde{\alpha}_{\circ:l:r}^{d,p}(i) \times \varphi_r \qquad \ddot{\Delta}_{\circ:l:r}^{d,i} \triangleq \tilde{\Delta}_{\circ:l:r}^{d,i} \times \varphi_r \qquad (5.78b)$$

Using these definitions and the definitions for their fully scaled versions in Equation-(5.74), we can immediately rewrite them as:

$$\ddot{\alpha}_{l:r}^{d,p}(i) \triangleq \tilde{\alpha}_{l:r}^{d,p}(i) \times \varphi_r = \sum_{t=l}^r \tilde{\alpha}_{l:t}^{d,p}(i) \left(\tilde{\Delta}_{t:r}^{d+1,i} \times \varphi_r \right) = \sum_{t=l}^r \tilde{\alpha}_{l:t}^{d,p}(i) \ddot{\Delta}_{t:r}^{d+1,i} \qquad (5.79a)$$

$$\ddot{\Delta}_{t:r}^{d+1,i} \triangleq \tilde{\Delta}_{t:r}^{d+1,i} \times \varphi_r = \begin{cases} B_{y_r|i} & \text{if } d+1 = D, t = r \\ \sum_{s \in \text{ch}(i)} \ddot{\alpha}_{t:r}^{d+1,i}(s) A_{s,\text{end}}^{d+2,i} & \text{if } d+1 < D \end{cases} \qquad (5.79b)$$

Given the recursive hypothesis, Equation-(5.79) are completely defined through the chain:

$$\ddot{\Delta}_{r:r}^{D,i} \longrightarrow \ddot{\alpha}_{l:r}^{D-1,p}(i) \longrightarrow \ddot{\Delta}_{l:r}^{D-1,i} \longrightarrow \ddot{\alpha}_{l:r}^{D-2,p}(i) \longrightarrow \dots$$

This time, however, the initialisation $\ddot{\Delta}_{r:r}^{D,i} = B_{y_r|i}$ is computable without the knowledge of φ_r . Similarly the pair $\{\ddot{\alpha}, \ddot{\Delta}\}$, is computed as:

$$\ddot{\alpha}_{l:r}^{d,p}(i) \triangleq \tilde{\alpha}_{l:r}^{d,p}(i) \times \varphi_r = \sum_{t=l}^r \tilde{\alpha}_{l:t}^{d,p}(i) \ddot{\Delta}_{\circ:t:r}^{d+1,i} \qquad (5.80a)$$

$$\ddot{\Delta}_{\circ:t:r}^{d+1,i} \triangleq \tilde{\Delta}_{\circ:t:r}^{d+1,i} \times \varphi_r = \begin{cases} 0 & \text{if } d+1 = D \\ \sum_{s \in \text{ch}(i)} \left(\ddot{\alpha}_{t:r}^{d+1,i}(s) (1 - A_{s,\text{end}}^{d+1,i}) + \ddot{\alpha}_{t:r}^{d+1,i}(s) \right) & \text{if } d < D \end{cases} \qquad (5.80b)$$

Given all the partially scaled variables computed till time r , the scaling factor φ_r is obtained by summing $\ddot{\alpha}$ and $\ddot{\Delta}$ at the top level:

$$\varphi_r = \sum_{i \in \mathcal{S}^2} \left(\ddot{\alpha}_{1:r}^{1,1}(i) + \ddot{\Delta}_{\circ:1:r}^{1,1}(i) \right) \qquad (5.81)$$

A proof for this equation is given in Appendix B. Finally, in our inductive approach,

at the very first time slice, $r = 1$, the set of partially scaled variables are equal to their corresponding unscaled versions. That is, for example:

$$\ddot{\alpha}_{1:1}^{d,p}(i) \triangleq \tilde{\alpha}_{1:1}^{d,p}(i) \times \varphi_1 = \frac{\alpha_{1:1}^{d,p}(i)}{\varphi_1} \times \varphi_1 = \alpha_{1:1}^{d,p}(i)$$

The pseudocode to calculate the pair of scaled variables $\{\tilde{\alpha}_{l:r}^{d,p}(i), \tilde{\vartheta}_{l:r}^{d,p}(i)\}$ is outlined in Algorithm 5.6.

Algorithm 5.6 Calculating the set of scaled inside variables

```

For  $r = 1$  to  $T$  (left-right) Do
  For  $d = D - 1$  downto 1 (bottom-up) and  $\forall p \in \mathcal{S}^d, i \in \text{ch}(p)$  Do
    For  $l = 1$  to  $\mathbf{R}_r^d$  (Equation-(5.55)) Do
      Calculate  $\ddot{\alpha}_{l:r}^{d,p}(i)$  from Equation-(5.79)
      Calculate  $\ddot{\vartheta}_{l:r}^{d,p}(i)$  from Equation-(5.80)
    EndFor
  EndFor (loop over  $d$ )
  Calculate the scaling factor  $\varphi_r$  from Equation-(5.81)
  For  $d = D - 1$  downto 1 (bottom-up) and  $\forall p \in \mathcal{S}^d, i \in \text{ch}(p)$  Do
    Calculate  $\tilde{\alpha}_{l:r}^{d,p}(i) = \frac{1}{\varphi_r} \ddot{\alpha}_{l:r}^{d,p}(i)$ 
    Calculate  $\tilde{\vartheta}_{l:r}^{d,p}(i) = \frac{1}{\varphi_r} \ddot{\vartheta}_{l:r}^{d,p}(i)$ 
  EndFor
EndFor

```

It is easy to verify that the complexity for this algorithm is $O(T^3 S b^2 D)$, which is the same as that of Algorithm 5.3 used to compute its unscaled counterparts. It is very important to note, however, that the order when performing the loop is different in Algorithm 5.6 and Algorithm 5.3.

5.8.2 Calculating the scaled outside variables

There is still the set of scaled outside variables $\{\tilde{\lambda}_{l;r}^{d,p}(i), \tilde{\Lambda}_{l;r}^{d,p}\}$ to compute. Fortunately, once the scaled inside variables are calculated, this set of variables can be obtained directly from their computational forms given in Equation-(5.65) by simply replacing the unscaled

variables by their corresponding scaled versions. That is:

$$\begin{aligned} \tilde{\Lambda}_{1;T}^{1,1} &= 1, & \tilde{\lambda}_{1;T}^{1,1}(i) &= A_{i,\text{end}}^{1,1} \\ \tilde{\lambda}_{1;r}^{1,1}(i) &= \sum_{t \in \mathbf{R}_1^1} \sum_{j \in \mathcal{S}^2} \tilde{\lambda}_{1;t}^{1,1}(j) \tilde{\Delta}_{r+1:t}^{2,j} A_{i,j}^{1,1} & \text{for } r < T \end{aligned} \quad (5.82a)$$

$$\tilde{\Lambda}_{l;r}^{d,p} = \sum_{q \in \text{pa}(p)} \sum_{t \in \mathbf{L}^d} \tilde{Q}_{t:l}^{d-1,q}(p) \tilde{\lambda}_{t;r}^{d-1,q}(p) \quad (5.82b)$$

$$\tilde{\lambda}_{l;r}^{d,p}(i) = \sum_{t \in \mathbf{R}_r^d} \sum_{j \in \text{ch}(p)} \tilde{\lambda}_{l;t}^{d,p}(j) A_{i,j}^{d,p} \tilde{\Delta}_{r+1:t}^{d+1,j} + \tilde{\Lambda}_{l;r}^{d,p} A_{i,\text{end}}^{d,p} \quad (5.82c)$$

The pseudocode to compute these variables follow the same procedure as presented in Algorithm 5.4 that used to compute their unscaled counterparts.

5.9 Closing Remarks

The Hierarchical HMM is an extension of the Hidden Markov Model to include a hierarchy of hidden states. This form of hierarchical modeling has been found useful in many applications and offers an integrated probabilistic framework to model a video at multiple levels of semantics. The state hierarchy in the original HHMM (Fine *et al.*, 1998) is however restricted to a tree structure. This prohibits two different states from having the same child, and thus does not allow for the sharing of common substructures in the model. In this chapter, we have presented a general HHMM in which the state hierarchy can be a lattice, which allows for the arbitrary sharing of substructures. We have presented a thorough study for this model including the problem of inference and learning. In addition, we provide a method for numerical scaling to avoid underflow, an important issue in dealing with long observation sequences.

Given the theoretical results for the HHMM, in the next chapter, we present its application to the problem of segmentation and narrative structure discovery in educational videos.

Chapter 6

Semantic Analysis with the Hierarchical HMM

The ability to represent and model the hierarchic nature of a video is of great interest. It enables segmentation, browsing, and retrieval of videos at semantic levels of different granularity. The majority of previous works (reviewed in Chapter 2) have modeled each semantic level separately, and pure heuristics are usually used to form a higher semantic unit. As we have highlighted in Chapter 2, the underlying problem with these approaches is that they do not capture the essential nature of the interaction across semantic layers, and these interactions are not incorporated during the learning process. The Hierarchical HMM provide an integrated probabilistic framework for modeling the interaction both across and within semantic layers. However, previous works using this model in video analysis have not retained structural information within the hierarchy explicitly, and thus the information about the structure is not effectively utilised in segmentation or classification. Furthermore, as we have argued, shared structures naturally embedded in the domain have not been exploited in previous research due to the lack of theoretical work for this model. Our previous chapter has filled this theoretical hole and this allows us to readily deploy this model for the problem of video content analysis.

In this chapter, we report two applications of the Hierarchical HMM to (1) segment a video into semantic (topical) segments, and (2) to automatically learn useful structural units. From the theoretical perspective, to add the segmentation ability to the model, we need to develop a Viterbi algorithm to compute the best sequence of states. In addition, we need to be able to handle the continuous observation. Our contributions from this chapter are: (1) a new Viterbi algorithm for the HHMM with the general state hierarchy developed in Chapter 5 for segmentation of a video and the parameter estimation for the continuous observation case, (2) a novel application of the HHMM to detect subtopic boundaries, and (3) a novel application of the HHMM to automatically discover semantic

units in a semi-supervised manner.

The remainder of the chapter is organised as follows. Section 6.1 presents the details of a generalised Viterbi algorithm developed for the HHMM. Next, we present a parameter estimation problem when the observation in the HHMM is modeled as a mixture of Gaussians in Section 6.2. The application of the HHMM for subtopic boundary detection is presented in Section 6.3. This is followed by another application of the HHMM for automatically learning meaningful structural units in educational films in Section 6.4. Finally, our closing remarks are contained in Section 6.5.

6.1 The Generalised Viterbi Algorithm for the HHMM

Indexing a video is to segment the video stream into meaningful indices. When the video is modeled as a hierarchical HMM, to perform segmentation we need an algorithm to infer the best sequences of states at different levels and their associated switching times. For the HHMM with the tree-structure topology, a generalised Viterbi has been developed in (Fine *et al.*, 1998). This algorithm is however *inapplicable* for the HHMM with a general state hierarchy. In this section, we provide a generalised Viterbi algorithm to handle such a case. We note that the Viterbi procedure in (Fine *et al.*, 1998) then becomes a special case of the generalised Viterbi developed in this section.

Given the observation sequence \mathcal{O} , the Viterbi objective is to find the corresponding hierarchical sequence of states $\{x_{1:T}^{1:D}\}$ and switching times $\{e_{1:T}^{1:D}\}$ which can best explain the observation:

$$\{x_{1:T}^{1:D}, e_{1:T}^{1:D}\}^* = \operatorname{argmax}_{\{x_{1:T}^{1:D}, e_{1:T}^{1:D}\}} \Pr(x_{1:T}^{1:D}, e_{1:T}^{1:D} \mid \mathcal{O}) \quad (6.1)$$

This requires two stages. First, at level d , for each pair of time indices (l, r) and parent-child (p, i) , we maintain the maximum probabilities over all possible configurations that account for the observation from $l \rightarrow r$ via the auxiliary variable $\delta_{l,r}^{d,p}(i)$, along with best recorded states and best switching times. This is done via dynamic programming. The second phase is a backtracking procedure using the dynamic tables in the previous stage to decode the best sequences of hierarchy of states and switching times.

Computing Maximum Probability Tables via Dynamic Programming

Using a similar idea to the method of calculating the asymmetric inside variable $\alpha_{l,r}^{d,p}(i)$ in Chapter 5, we use a *maximum (asymmetric) inside* variable $\delta_{l,r}^{d,p}(i)$ that keeps track of the maximum probability computed over all possible configurations of the hidden variables

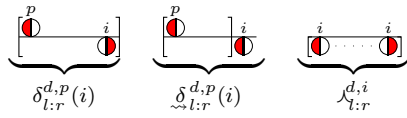


Figure 6-1: Diagrammatic visualisation for the set of ‘maximum’ variables.

‘inside’ the (asymmetric) boundary $\text{AB}_{l:r}^{d,p}(i)$. Denote this set of variables by $V_{l:r}^{d,p}(i) \triangleq \{\text{AI}_{l:r}^{d,p}(i) \setminus y_{l:r}\}$, the variable $\delta_{l:r}^{d,p}(i)$ is then defined as:

$$\delta_{l:r}^{d,p}(i) \triangleq \max_{V_{l:r}^{d,p}(i)} \Pr(y_{l:r}, x_r^{d+1} = i, e_{l:r-1}^d = \mathbf{0}, V_{l:r}^{d,p}(i) \mid \cdot x_l^d = p) \quad (6.2)$$

This is the key quantity to maintain the structure of the maximum configuration that accounts for the observation from l to r . Similar to $\alpha_{l:r}^{d,p}(i)$, $\delta_{l:r}^{d,p}(i)$ can be computed efficiently via dynamic programming algorithm. For convenience, we introduce two more auxiliary variables, the *started maximum (asymmetric) inside* variable $\tilde{\delta}_{l:r}^{d,p}(i)$, and the *maximum symmetric inside variable* $\lambda_{l:r}^{d,i}$ as follows:

$$\tilde{\delta}_{l:r}^{d,p}(i) \triangleq \max_{V_{l:r}^{d,p}(i)} \Pr(y_{l:r-1}, \cdot x_r^{d+1} = i, e_{l:r-1}^d = \mathbf{0}, V_{l:r}^{d,p}(i) \mid \cdot x_l^d = p) \quad (6.3a)$$

$$\lambda_{l:r}^{d,i} \triangleq \max_{W_{l:r}^{d,i}} \Pr(y_{l:r}, e_{l:r-1}^d = \mathbf{0}, e_l^d = 1, W_{l:r}^{d,i} \mid \cdot x_l^d = i) \quad (6.3b)$$

where in Equation-(6.3b), $W_{l:r}^{d,i} \triangleq \{\text{SB}_{l:r}^{d,i} \setminus y_{l:r}\}$ (ie: the set of all hidden variables ‘inside’ the symmetric boundary $\text{SB}_{l:r}^{d,i}$). We provide the visualisation for these variables in Figure-(6-1). Again, this visualisation offers us an intuitive way to convey the recursive relationships. For example, the strategy to compute $\delta_{l:r}^{d,p}(i)$ is by recursively decomposing it into two components: a started maximum inside probability $\tilde{\delta}_{l:t}^{d,p}(i)$ and a maximum symmetric inside probability $\lambda_{t:r}^{d+1,i}$ where t is the starting time of the children x_r^{d+1} . This results in the recursive formula:

$$\delta_{l:r}^{d,p}(i) = \max_{l \leq t \leq r} \left\{ \tilde{\delta}_{l:t}^{d,p}(i) \wedge_{t:r}^{d+1,i} \right\} \quad (6.4)$$

and diagrammatically: $\underbrace{\left[\begin{array}{c} p \\ \text{---} \\ i \end{array} \right]}_{\delta_{l:r}^{d,p}(i)} = \max_{l \leq t \leq r} \underbrace{\left[\begin{array}{c} p \\ \text{---} \\ i \end{array} \right]}_{\tilde{\delta}_{l:t}^{d,p}(i)} \times \underbrace{\left[\begin{array}{c} i \dots i \\ \text{---} \\ i \end{array} \right]}_{\lambda_{t:r}^{d+1,i}}$

At this stage, we also need to maintain the switching time t that results in a maximum in Equation-(6.4) for later backtracking purpose. We define the *switching time* variable $\psi_{l:r}^{d,p}(i)$ to keep track of t :

$$\psi_{l:r}^{d,p}(i) \triangleq \operatorname{argmax}_{l \leq t \leq r} \left\{ \tilde{\delta}_{l:t}^{d,p}(i) \wedge_{t:r}^{d+1,i} \right\} \quad (6.5)$$

Now let us turn to the computation of the variable $\tilde{\delta}_{l:r}^{d,i}(p)$. For the initial case when $r = l$,

by definition $\delta_{l:l}^{d,p}(i) \triangleq \pi_i^{d,p}$. For the case $r > l$, it is decomposed as shown below:

$$\delta_{l:r}^{d,p}(i) = \max_{j \in \text{ch}(p)} \left\{ \delta_{l:r-1}^{d,p}(j) A_{j,i}^{d,p} \right\} \quad (6.6)$$

$$\underbrace{\left[\overset{p}{\circ} \text{---} \overset{i}{\circ} \right]}_{\delta_{l:r}^{d,p}(i)} = \max_{j \in \text{ch}(p)} \underbrace{\left[\overset{p}{\circ} \text{---} \overset{j}{\circ} \right]}_{\delta_{l:r-1}^{d,p}(j)} \times \underbrace{\left[\overset{j}{\circ} \text{---} \overset{i}{\circ} \right]}_{A_{j,i}^{d,p}}$$

As can be seen, the recursion now appears in the calculation of $\delta_{l:r}^{d,p}(i)$ in Equation-(6.6) as it requires the result of $\delta_{l:r-1}^{d,p}(j)$ at previous time slices. At this stage, we need to keep track of the best switching children state j via the *switching state* variable¹ $\omega_{l:r}^{d,p}(i)$:

$$\omega_{l:r}^{d,p}(i) \triangleq \operatorname{argmax}_{j \in \text{ch}(d,p)} \left\{ \delta_{l:r-1}^{d,p}(j) A_{j,i}^{d,p} \right\} \quad (6.7)$$

The last computation has been left out so far is the maximum symmetric inside variable $\lambda_{l:r}^{d,i}$. This variable can be readily computed in a bottom-up approach. In the initial phase at the bottom level, by definition²: $\lambda_{t:t}^{D,i} \triangleq B_{yt|i}$. In the inductive step when $d > D$, it is given as:

$$\lambda_{l:r}^{d,i} = \max_{s \in \text{ch}(i)} \left\{ \delta_{l:r}^{d,i}(s) A_{s,\text{end}}^{d,i} \right\} \quad (6.8)$$

and diagrammatically: $\underbrace{\left[\overset{i}{\circ} \text{---} \dots \text{---} \overset{i}{\circ} \right]}_{\lambda_{l:r}^{d,i}} = \max_{s \in \text{ch}(i)} \underbrace{\left[\overset{i}{\circ} \text{---} \overset{s}{\circ} \right]}_{\delta_{l:r}^{d,i}(s)} \times \underbrace{\left[\overset{s}{\circ} \text{---} \overset{i}{\circ} \right]}_{A_{s,\text{end}}^{d,i}}$

We note that in the calculation of $\delta_{l:r}^{d,p}(i)$ at level d in Equation-(6.4), it requires the results of $\lambda_{t:r}^{d+1,i}$ *only* at the lower level $d+1$, therefore $\lambda_{t:r}^{d+i,i}$ is completely defined.

In summary, the set of equations (6.4), (6.6), (6.7), and (6.8) (and their initial computation) constitute a recursive algorithm to compute the maximum configuration table at different levels. The pseudocode is obtained similarly from Algorithm 5.3 which was used in Chapter 5 to compute the set of inside variables. This algorithm is performed via dynamic programming and in a bottom-up, left-right manner. During the calculation of this table we also keep track of extra information about the best switching times and states and defined in Equation-(6.5) and Equation-(6.7) respectively for later backtracking purpose.

Decoding the Best Sequence of States and Ending Status

In the second phase, the best sequence of states $\{x_{1:T}^{1:D}\}^*$ and ending statuses $\{e_{1:T}^{1:D}\}^*$

¹When $l = r$, $\omega_{l:r}^{d,p}(i)$ is undefined and set to 0.

²Note that at the bottom level, ie: $d = D$, by definition $e_t^D = 1$ for all t , and thus $\lambda_{l:r}^{D,i}$ is defined only for $l = r$.

are computed by a backtracking routine via $\psi_{l:r}^{d,p}(i)$ and $\omega_{l:r}^{d,p}(i)$. The pseudocode of the algorithm is outlined in Algorithm 6.1.

Algorithm 6.1 Backtracking for Generalised Viterbi.

Input: Values of $\delta_{l:r}^{d,p}(i)$, $\psi_{l:r}^{d,p}(i)$ and $\omega_{l:r}^{d,p}(i)$ (used globally) for all d, l, r, p and i . T is length of the observation and D is the depth of the model.

/ initialise at root level /

$\{\mathbf{s}_1^1, \boldsymbol{\tau}_1^1\} = \text{find-max-config}(1, T, 1, 1)$ (see Algorithm 6.2)

For d from 2 to $D - 1$

 set $\mathbf{s}^d, \boldsymbol{\tau}^d$ to empty

/ loop though best times at previous level /

 For n from 1 to length of $\boldsymbol{\tau}^{d-1}$

/ calculate started/ended times l, r /

 set $l = \boldsymbol{\tau}_n^d$

 If $n < \text{length of } \boldsymbol{\tau}^{d-1}$

 set $r = \boldsymbol{\tau}_{n+1}^{d-1} - 1$ Else set to T

$\{\mathbf{s}^*, \boldsymbol{\tau}^*\} = \text{find-max-config}(l, r, d, \mathbf{s}_n^{d-1})$

 set $\mathbf{s}^d = \mathbf{s}^d \odot \mathbf{s}^*, \boldsymbol{\tau}^d = \boldsymbol{\tau}^d \odot \boldsymbol{\tau}^*$

/ where \odot is the concatenation operator /

Output: Sequence of best states \mathbf{s}_m^d and its corresponding activation time $\boldsymbol{\tau}_m^d$ for $1 \leq d < D$.

In general, the algorithm decodes the best configuration in a top-down fashion and the key routine is $\text{find-max-config}(l, r, d, p)$, which finds the best sequences of states and times within time $l \rightarrow r$ given that state p is started at time l . The pseudo code for this routine is shown in Algorithm 6.2 with the aid of Figure-(6-2) to visualise the recursive structure in the finding the maximum configuration. The complexity for this Viterbi algorithm can be shown to have the same complexity as Algorithm 5.3 for the asymmetric inside variable α presented in Chapter 5.

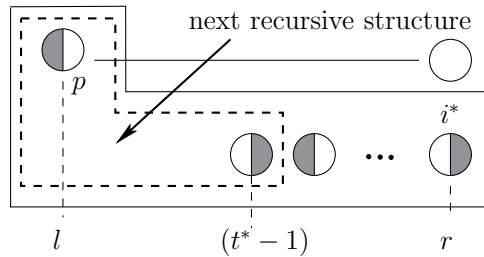


Figure 6-2: Visualisation of the recursive structure for routine $\text{find-max-config}(l, r, d, p)$.

In practice, to avoid numerical underflow, all calculations can be done in the log space, that is we maintain versions of logarithms of these variables instead. The set of equations then becomes:

Algorithm 6.2 Sub-routine $\{\mathbf{s}^*, \boldsymbol{\tau}^*\} = \text{find-max-config}(l, r, d, p)$

Return: the best sequence of state \mathbf{s}^* and times $\boldsymbol{\tau}^*$ at level d from $l \rightarrow r$ given that this segment is bounded by state p started at l at level d .

```

set  $\mathbf{s}^*$  and  $\boldsymbol{\tau}^*$  to empty
retrieve the best state:  $i^* = \operatorname{argmax}_i \delta_{l:r}^{d,p}(i)$ 
set  $\mathbf{s}^* = i^*$ ,  $\boldsymbol{\tau}^* = t^* = \psi_{l:r}^{d,p}(i^*)$ 
/ now find all other best states/times /
While  $t^* > l$  Do
  set  $i^* = \omega_{l:r}^{d,p}(i^*)$ 
   $\mathbf{s}^* = i^* \odot \mathbf{s}^*$ 
  set  $r = t^* - 1$  / update right boundary /
  / retrieve the next best time /
  set  $t^* = \psi_{l:r}^{d,p}(i^*)$ 
  set  $\boldsymbol{\tau}^* = t^* \odot \boldsymbol{\tau}^*$ 
EndWhile

```

In log-space	Corresponding Equation
$\log \delta_{l:r}^{d,p}(i) \triangleq \max_{l \leq t \leq r} \left\{ \log \delta_{l:t}^{d,p}(i) + \log \lambda_{t:r}^{d+1,i} \right\}$	Equation-(6.4)
$\psi_{l:r}^{d,p}(i) \triangleq \operatorname{argmax}_{l \leq t \leq r} \left\{ \log \delta_{l:t}^{d,p}(i) + \log \lambda_{t:r}^{d+1,i} \right\}$	Equation-(6.5)
$\log \delta_{l:r}^{d,p}(i) \triangleq \max_{j \in \text{ch}(p)} \left\{ \log \delta_{l:r-1}^{d,p}(j) + \log A_{j,i}^{d,p} \right\}$	Equation-(6.6)
$\omega_{l:r}^{d,p}(i) \triangleq \operatorname{argmax}_{j \in \text{ch}(p)} \left\{ \log \delta_{l:r-1}^{d,p}(j) + \log A_{j,i}^{d,p} \right\}$	Equation-(6.7)
$\log \lambda_{l:r}^{d,i} \triangleq \max_{s \in \text{ch}(i)} \left\{ \log \delta_{l:r}^{d,i}(s) + \log A_{s,\text{end}}^{d,i} \right\}$	Equation-(6.8)

6.2 Modeling Continuous Observations

In this section we discuss the situation when the emission observation probability is modeled as a mixture of Gaussians.

6.2.1 Mixture of Gaussians Emission Probability

The observation matrix B in the discrete case is replaced by: the mixing weight matrix $\boldsymbol{\kappa}$, and a set of means and covariance matrix $\{\mu, \Sigma\}$. The DBN network at the lowest level is modified as in Figure-(6-3), where a mixture variable z_t is added. Let M be the number of mixture components and $N = |\mathcal{S}^D|$ be the number of states at level D , then the mixing

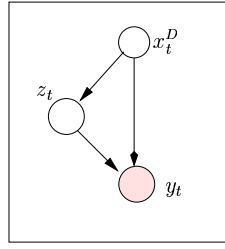


Figure 6-3: Graphical representation of the lowest level in the DBN structure with added mixture component z_t .

weight probability is defined as:

$$\kappa_{mi} \triangleq \Pr(z_t = m \mid x_t^D = i) \quad (6.9)$$

and at time t , for the mixture $z_t = m$ and the production state $x_t^D = i$, the emission probability for a real-valued observation y_t is given as:

$$\Pr(y_t \mid z_t = m, x_t = i) = \mathcal{N}(y_t, \mu_{mi}, \Sigma_{mi}) \quad (6.10)$$

where $\mathcal{N}(\cdot)$ is the multivariate Gaussian density function defined as:

$$\mathcal{N}(y, \mu, \Sigma) = (2\pi)^{-l/2} |\Sigma|^{-1/2} \exp \left\{ -\frac{1}{2} (y - \mu)^\top \Sigma^{-1} (y - \mu) \right\}$$

with l being the dimension of the observation vector y_t . When the mixture variable z_t is omitted, the emission probability is computed as a weighted mixture defined as:

$$\begin{aligned} \Pr(y_t \mid x_t^D = i) &= \sum_{m=1}^M \Pr(z_t = m \mid x_t^D = i) \Pr(y_t \mid z_t = m, x_t = i) \\ &= \sum_{m=1}^M \kappa_{mi} \mathcal{N}(y_t, \mu_{mi}, \Sigma_{mi}) \end{aligned} \quad (6.11)$$

6.2.2 Parameter Estimation

Let us now concentrate on the problem of estimating these new parameters. Assume data \mathcal{D} is given, expressing the complete logarithm likelihood and ignoring terms that are irrelevant to z_t and y_t , we have:

$$\ell_G^c(\mathcal{D} \mid \theta) = \sum_{t=1}^T \log \Pr(y_t \mid x_t, z_t, \mu, \Sigma) + \sum_{t=1}^T \log \Pr(z_t \mid x_t, \kappa)$$

Again we see that the parameter set has been decoupled. Using the ‘identity-checking trick’, looping over all possible values of z_t, x_t , we have:

$$\begin{aligned} \ell_G^c(\mathcal{D} | \theta) &= \sum_{t=1}^T \log \prod_{m=1}^M \prod_{i=1}^N \Pr(y_t | z_t = m, x_t^D = i)^{\delta_{z_t}^{(m)} \delta_{x_t^D}^{(i)}} + \sum_{t=1}^T \log \prod_{m=1}^M \prod_{i=1}^N \Pr(z_t = m | x_t^D = i)^{\delta_{z_t}^{(m)} \delta_{x_t^D}^{(i)}} \\ &= \sum_{\substack{1 \leq i \leq N \\ 1 \leq m \leq M}} \left[\sum_{t=1}^T \delta_{z_t}^{(m)} \delta_{x_t^D}^{(i)} \log \mathcal{N}(y_t, \mu_{mi}, \Sigma_{mi}) + \sum_{t=1}^T \left(\delta_{z_t}^{(m)} \delta_{x_t^D}^{(i)} \right) \log \kappa_{mi} \right] \end{aligned}$$

and the expected complete logarithm likelihood is given as:

$$\langle \ell_G^c(\mathcal{D} | \theta) \rangle = \sum_{\substack{1 \leq i \leq N \\ 1 \leq m \leq M}} \left[\sum_{t=1}^T \langle \delta_{z_t}^{(m)} \delta_{x_t^D}^{(i)} \rangle \log \mathcal{N}(y_t, \mu_{mi}, \Sigma_{mi}) + \sum_{t=1}^T \langle \delta_{z_t}^{(m)} \delta_{x_t^D}^{(i)} \rangle \log \kappa_{mi} \right]$$

where the expectation is calculated as:

$$\begin{aligned} \langle \delta_{z_t}^{(m)} \delta_{x_t^D}^{(i)} \rangle &= \Pr(z_t = m, x_t^D = i | \mathcal{O}) \\ &= \Pr(z_t = m | x_t^D = i, y_t) \Pr(x_t^D = i | \mathcal{O}) \\ &= \frac{\Pr(y_t | z_t = m, x_t^D = i) \Pr(z_t = m | x_t^D = i) \Pr(x_t^D = i, \mathcal{O})}{\Pr(y_t | x_t^D = i) \Pr(\mathcal{O})} \\ &= \frac{\Gamma_t^D(i) \kappa_{mi} \mathcal{N}(y_t, \mu_{mi}, \Sigma_{mi})}{\Pr(\mathcal{O}) \sum_{m=1}^M \kappa_{mi} \mathcal{N}(y_t, \mu_{mi}, \Sigma_{mi})} \end{aligned} \quad (6.12)$$

where $\Gamma_t^D(i) \triangleq \Pr(x_t^D = i, \mathcal{O})$ is the emission auxiliary variable which has been defined and computed in Chapter 5.

Re-estimating the Mixture Weight

Using the Lagrange multipliers, we maximise the likelihood function separately with the constraint:

$$\sum_{m=1}^M \kappa_{mi} = \sum_{m=1}^M \Pr(z_t = m | x_t = i) = 1, \quad \forall i$$

Using Theorem 5.3, the mixture matrix is ready to be re-estimated as:

$$\hat{\kappa}_{mi} = \frac{\sum_{t=1}^T \langle \delta_{z_t}^{(m)} \delta_{x_t^D}^{(i)} \rangle}{\sum_{m=1}^M \sum_{t=1}^T \langle \delta_{z_t}^{(m)} \delta_{x_t^D}^{(i)} \rangle} \quad (6.13)$$

Re-estimating Gaussian Parameters $\{\mu, \Sigma\}$

To estimate μ_{mi} , take the first derivative of the expected complete log likelihood w.r.t μ_{mi}

and ignore irrelevant terms, we have:

$$\begin{aligned} \frac{\partial \langle \ell \rangle}{\partial \mu_{mi}} &= \frac{\partial}{\partial \mu_{mi}} \left\{ -\frac{1}{2} \sum_{m=1}^M \sum_{i=1}^N \sum_{t=1}^T \left\langle \delta_{z_t}^{(m)} \delta_{x_t^D}^{(i)} \right\rangle (y_t - \mu_{mi})^\top \Sigma_{mi}^{-1} (y_t - \mu_{mi}) \right\} \\ &= \sum_{t=1}^T \left\langle \delta_{z_t}^{(m)} \delta_{x_t^D}^{(i)} \right\rangle (y_t - \mu_{mi})^\top \Sigma_{mi}^{-1} \end{aligned} \quad (6.14)$$

where we have used the result $\frac{\partial}{\partial x} (x^\top V x) = 2Vx$ if V is symmetric. Setting the derivative to zero, we obtain:

$$\hat{\mu}_{mi} = \frac{\sum_{t=1}^T \left\langle \delta_{z_t}^{(m)} \delta_{x_t^D}^{(i)} \right\rangle y_t}{\sum_{t=1}^T \left\langle \delta_{z_t}^{(m)} \delta_{x_t^D}^{(i)} \right\rangle} \quad (6.15)$$

We obtain the covariance matrix Σ_{mi} in a similar way with a trace ‘trick’ as in (Jordan, 2004). First, rewrite the completed logarithm likelihood with respect to Σ , we have:

$$\begin{aligned} \langle \ell(\Sigma | \mathcal{D}) \rangle &= \sum_{\substack{1 \leq i \leq N \\ 1 \leq m \leq M}} \left[\sum_{t=1}^T \left\langle \delta_{z_t}^{(m)} \delta_{x_t^D}^{(i)} \right\rangle \left(-\frac{1}{2} \log |\Sigma_{mi}| - \frac{1}{2} (y_t - \mu_{mi})^\top \Sigma_{mi}^{-1} (y_t - \mu_{mi}) \right) \right] \\ &= \sum_{\substack{1 \leq i \leq N \\ 1 \leq m \leq M}} \left[\sum_{t=1}^T \left\langle \delta_{z_t}^{(m)} \delta_{x_t^D}^{(i)} \right\rangle \left(\frac{1}{2} \log |\Sigma_{mi}^{-1}| - \frac{1}{2} \text{tr} [(y_t - \mu_{mi})^\top \Sigma_{mi}^{-1} (y_t - \mu_{mi})] \right) \right] \\ &= \sum_{\substack{1 \leq i \leq N \\ 1 \leq m \leq M}} \left[\sum_{t=1}^T \left\langle \delta_{z_t}^{(m)} \delta_{x_t^D}^{(i)} \right\rangle \left(\frac{1}{2} \log |\Sigma_{mi}^{-1}| - \frac{1}{2} \text{tr} [(y_t - \mu_{mi})^\top (y_t - \mu_{mi}) \Sigma_{mi}^{-1}] \right) \right] \end{aligned} \quad (6.16)$$

where we have used the fact that $(y_t - \mu_{mi})^\top \Sigma_{mi}^{-1} (y_t - \mu_{mi})$ is a scalar; $|\Sigma|^{-1} = |\Sigma^{-1}|$ and the cyclical permutation property of the trace function, ie: $\text{tr} [x^\top A x] = \text{tr} [x x^\top A]$. Taking the first derivative of $\langle \ell(\Sigma | \mathcal{D}) \rangle$ with respect to Σ_{mi}^{-1} , we have:

$$\frac{\partial \langle \ell(\Sigma | \mathcal{D}) \rangle}{\partial \Sigma_{mi}^{-1}} = \sum_{t=1}^T \left\langle \delta_{z_t}^{(m)} \delta_{x_t^D}^{(i)} \right\rangle \left(\frac{1}{2} \Sigma_{mi} - \frac{1}{2} (y_t - \mu_{mi})(y_t - \mu_{mi})^\top \right) \quad (6.17)$$

where we remember that Σ_{mi} is symmetric, $\frac{\partial}{\partial U} \log |U| = (U^{-1})^\top$ and $\frac{\partial}{\partial U} x^\top U x = \frac{\partial}{\partial U} \text{tr} [x^\top U x] = x x^\top$. Finally, setting to zero, we obtain the maximum likelihood estimated covariance:

$$\hat{\Sigma}_{mi} = \frac{\sum_{t=1}^T \left\langle \delta_{z_t}^{(m)} \delta_{x_t^D}^{(i)} \right\rangle (y_t - \mu_{mi})(y_t - \mu_{mi})^\top}{\sum_{t=1}^T \left\langle \delta_{z_t}^{(m)} \delta_{x_t^D}^{(i)} \right\rangle} \quad (6.18)$$

In the case we have K iid. sequences of observation $\{\mathcal{O}^{(1)}, \dots, \mathcal{O}^{(K)}\}$, the set of estimation formulae is given below, where we drop the index k and implicitly understand that terms

inside the bracket are referring to the k th observation sequence.

$$\begin{aligned}
\hat{\kappa}_{mi} &= \frac{\sum_{k=1}^K \left[\sum_{t=1}^T \left\langle \delta_{z_t}^{(m)} \delta_{x_t^D}^{(i)} \right\rangle \right]}{\sum_{m=1}^M \sum_{k=1}^K \left[\sum_{t=1}^T \left\langle \delta_{z_t}^{(m)} \delta_{x_t^D}^{(i)} \right\rangle \right]} \\
\hat{\mu}_{mi} &= \frac{\sum_{k=1}^K \left[\sum_{t=1}^T \left\langle \delta_{z_t}^{(m)} \delta_{x_t^D}^{(i)} \right\rangle y_t \right]}{\sum_{k=1}^K \left[\sum_{t=1}^T \left\langle \delta_{z_t}^{(m)} \delta_{x_t^D}^{(i)} \right\rangle \right]} \\
\hat{\Sigma}_{mi} &= \frac{\sum_{k=1}^K \left[\sum_{t=1}^T \left\langle \delta_{z_t}^{(m)} \delta_{x_t^D}^{(i)} \right\rangle (y_t - \mu_{mi})(y_t - \mu_{mi})^T \right]}{\sum_{k=1}^K \left[\sum_{t=1}^T \left\langle \delta_{z_t}^{(m)} \delta_{x_t^D}^{(i)} \right\rangle \right]} \tag{6.19}
\end{aligned}$$

In fact, in this case, since these observation sequences are assumed to be independently and identically distributed (iid) the expected complete logarithm likelihood will become:

$$\langle \ell(\theta; \mathcal{D}) \rangle = \sum_{\substack{1 \leq i \leq N \\ 1 \leq m \leq M}} \left[\sum_{k=1}^K \sum_{t=1}^T \left\langle \delta_{z_t}^{(m)} \delta_{x_t^D}^{(i)} \right\rangle \log \mathcal{N}(y_t, \mu_{mi}, \Sigma_{mi}) + \sum_{k=1}^K \sum_{t=1}^T \left\langle \delta_{z_t}^{(m)} \delta_{x_t^D}^{(i)} \right\rangle \log \kappa_{mi} \right]$$

Following the same steps as we have done before for a single observation sequence and adding the sum over K we obtain the results as presented.

6.3 Subtopic Boundaries Detection with the HHMM

In this section, we present a framework that uses the HHMM to detect subtopic transition boundaries in a sub-class of educational videos, namely safety and training videos (cf. Section 3.1). We organise this section as follows. In Subsection 6.3.1, we utilise the grammar knowledge of educational videos to construct the topology for the HHMM. Next, the details of the training phase are given in Section 6.3.2. Finally, detection results and discussion are provided in Section 6.3.3.

6.3.1 Incorporating Prior Knowledge into the Topology

An intrinsic functionality of educational videos is to ‘teach’, and therefore structuralising the content and building meaningful indices are important to improve the learning experience. Materials delivered in an educational video might vary widely to suit different purposes. However, when restricted to instructional and safety videos, the content organisation is relatively simple. Subjects are arranged into a sequence of subtopics starting with a few introduction shots. Literature in this field (Herman, 1965) offers further insight into how a subtopic is constructed. Generally, there are three presentational styles: (1) *direct*

instruction, (2) *on-screen* instruction, and (3) *illustrative* instruction (see Figure-(6-4)). In *direct* instruction, the video-maker chooses to present a subtopic by means of text captions and voice over. In *on-screen* instruction, s/he decides to appear on the camera to talk directly to the viewers. Lastly in *illustrative* instruction, illustrative examples are the major mode of presentation to convey the subject material with the possible appearance of the anchor-person (narrators). These presentational styles shares certain similar semantic concepts at the lower levels such as the introduction shot (Figure-(6-4)).

Our aim is to apply the HHMM model, taking advantage of the shared units, to segment an educational video into high-levels of abstraction – ie: detection of subtopic transitions in this case, which presents the same problem addressed in Section 4.2 of Chapter 3. We construct a 3-level HHMM as follows. The root level represents the entire video, followed by three states at the next level, each of which corresponds to one subtopic presentational style. The production level includes four states, corresponding to four semantic concepts at the shot level: (1) the introduction, (2) instruction delivered by means of captioned texts, (3) instruction delivered directly by the presenter and (4) illustrative example (Figure-(6-4)).

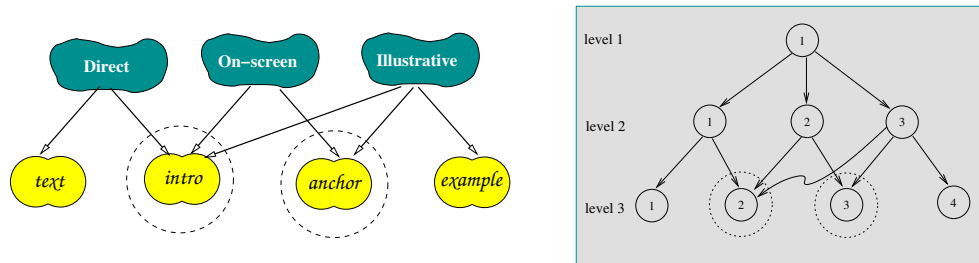


Figure 6-4: Structure of subtopic generating process with assumed hidden ‘styles’; and its mapping to a topology for the HHMM. Shared structures are identified with an extra dotted circle.

6.3.2 Training the Model

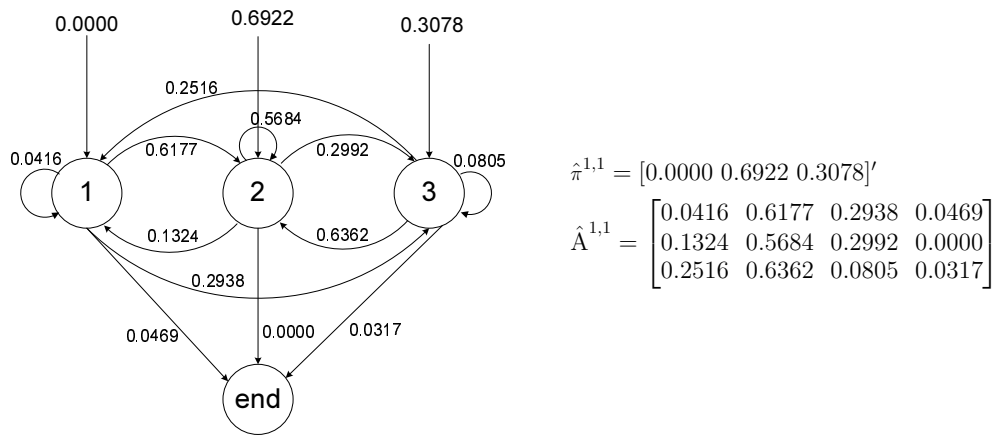
Given a training set of $N = 8$ videos, we extract features from each video and use them as input observation sequences to the EM parameter learning algorithm to estimate a new model parameter. This new parameter set will be used in the second phase to segment a video based on results from the generalised Viterbi decoding algorithm.

The data set includes eight instructional and safety videos, whose subtopics span a variety of subjects such as how to exercise safety measure at home, in the office, or at the workplace. Shot indices are assumed to be available, which are first detected by a commercial software and then errors are manually corrected. In the training phase, each video yields

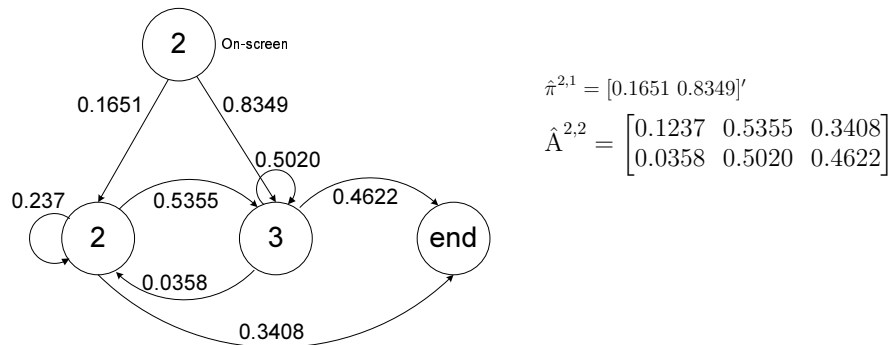
an observation sequence with each shot-based feature vector \vec{o} being a column vector of seven elements:

$$\vec{o} = [o_1, o_2, o_3, o_4, o_5, o_6, o_7]^t$$

From the visual stream, we extract three features (o_1, o_2, o_3) , including the face-content-ratio (FCR), text-content-ratio (TCR) and average motion based on camera pan and tilt. The other four features (o_4, o_5, o_6, o_7) namely music-ratio, speech-ratio, silence-ratio and NL-ratio are from the audio track. Feature music-ratio, for example, is calculated as the ratio of number of clips classified as music to the total number of audio clips in the shot. Details of these visual and aural feature sets can be further found in Section 3.3 of Chapter 3. The trained parameters at the top level are shown in Figure-(6-5). At the



(a) The root has three children states corresponding to three presentational styles.



(b) State 2 at the second level has two children corresponding to 'intro' and 'anchor'.

Figure 6-5: Re-estimated parameters after training for the model at the top two layers.

production level of the trained model, the estimated matrices $\hat{\mu}$ and $\hat{\Sigma}$ can be examined to get an idea about the semantics. Figure-(6-6), for instance, shows the estimated mean

values for different features with respect to the ‘introduction’ state, which is intended to model the ‘style’ of shots used to introduce a new subtopic. As can be seen, this state is

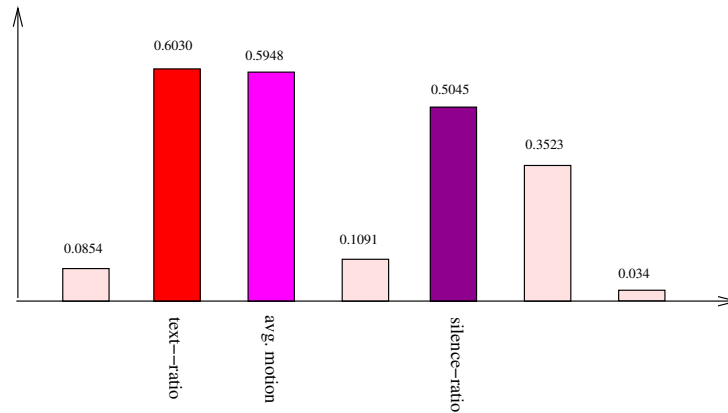


Figure 6-6: Visualisation of the re-estimated mean value $\hat{\mu}_{21}$ for ‘introduction state’ (state 2 at the production level).

‘sensitive’, ie: it will yield a high probability, to shots with displayed captioned texts and no audio. When compared with the groundtruth, we observe that this is indeed a major kind of shot that demarcates subtopics.

6.3.3 Subtopic Detection Results and Discussion

To evaluate the detection performance, we manually watch and segment each video into subtopics. In some cases, this information is available directly from the video manuals. This results in a total of 75 indices. We use two well-known metrics, namely *recall* and *precision* to measure the performance of the detection. To perform segmentation, we first run the Viterbi algorithm to get the time indices for which a state at the subtopic level (ie: level two) makes the transition. Let τ be such an index, we then examine the state x_{τ}^3 , which is the corresponding state at the production level being called. If this state coincides with the introduction shot (ie: = 2), then τ is recorded as a subtopic transition.

The entire segmentation results are reported in Table-(6.1). Calculation yields a recall of 77.3% and a precision of 70.7%. Given that the segmentation has been done in a completely unsupervised manner, ie: there are no hints in the training data as to what is a subtopic boundary, the result demonstrates the validity of the HHMM-based detection scheme. Compared with the subtopic detection scheme developed in Chapter 4, this framework clearly demonstrates its advantage. Even though marginally close in the performance, this framework doesnot require any groundtruth information about subtopic boundaries nor their related information such as the distribution of the distance between a local

Video	GroundTruth	TP	Errors	
			FN	FP
1. Eye Safety	10	8	2	4
2. Foot Safety	9	6	3	3
3. Hand Safety	8	7	1	0
4. Head Safety	5	3	2	4
5. Elect/Elecs Safety	7	6	1	10
6. Stop Burning	19	15	4	2
7. Welding Safety	10	7	3	0
8. SlipStripFall	7	6	1	1
Total	75	58	17	24

Table 6.1: Detection results.

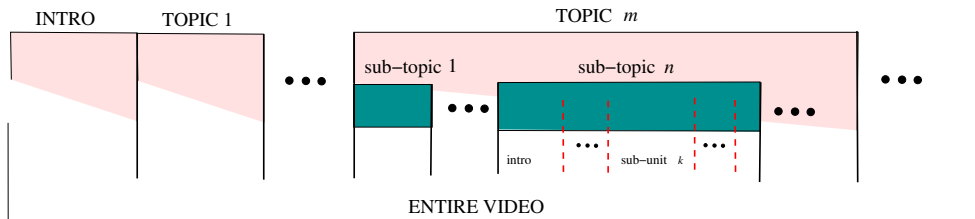


Figure 6-7: Further insight into the structure of an educational video.

minimum of the content density function and the subtopic boundary as needed previously in Section 4.2.4.

Discussion

Table-(6.1) reveals that over-segmentation is the major source of error causing a degradation in precision; and the high number of ‘misses’ (false negatives) cause a low recall rate. The resulting false negatives are not surprising since a topic is introduced in numerous ways, but the estimated model has learned only a subset of these methods of introduction. To overcome this, we obviously need a more complex model structure, which will be considered in our future work. The fact that the detector usually over-segments a video (eg: video 5) is worth further discussion. A close analysis discloses that while these (over-segmenting) indices do not match the groundtruth, they frequently map to the lower level of sub-structures within a topic, such as segments emphasising a safety message (for example, this happens many times in video 5). Figure-(6-7) depicts an insight into the structure of an educational video and illustrates this problem. The vertical solid lines have been the target of our detection, while dashed-lines correspond to the over-segmented indices from the detector. This fact suggests that the model might be utilised to exploit further structure in subtopics.

In summary, this section has presented a framework for subtopic boundary detection using

the HHMM. An important aspect of video data when considering its content organisation is the shared structures embedded in the data. This suggests a natural mapping to the Hierarchical HMM, which we have utilised to model the topic structures in educational videos. The experimental results have demonstrated the usefulness of the detection scheme.

6.4 Automatically Learning Structural Units with the HHMM

In this experiment, we apply the HHMM to automatically map semantic concepts to the model states at different levels of abstraction. Taking the advantages of the expressiveness of shared structures, we construct a three-level HHMM with a ‘true’ model in mind, ie: the topology of the HHMM is constructed in such a way that it maps into the hypothesised hierarchy as in Figure-(6-8). That is, for example, at the production level, we hypothesise that the model can ‘perfectly’ learn to map each state to an elementary feature such as face, text or speech, from which higher levels of descriptions such as an *on-screen* narration section can be built. Given this topology, a three-level HHMM is subsequently formed.

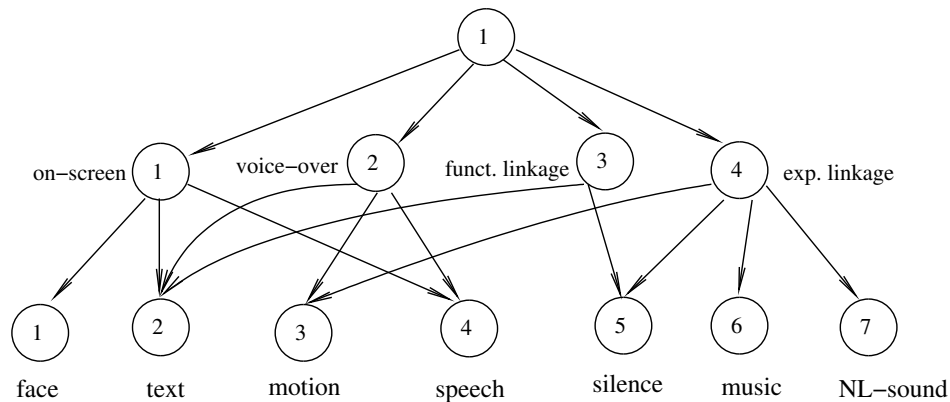


Figure 6-8: The hierarchy of connected concepts which is used as topological specification for the corresponding HHMM (*links from state 1, 2 (level two) to state 7 (level three) are not shown for readability*).

The production level includes seven states directly attached to the observed features (recall that these are shot-based features). The next level has four states, which in our opinion, adequately reflects the number of higher level units in this experiment. Finally, the root covers the entire video. We first randomly initialise the parameter of the model and then use the EM to learn a new set of model parameters.

The feature vector used in this experiment is the same as that used in the previous experiment. It includes seven features computed at the shot level. Three of them are from the

visual stream, namely: face-content-ratio, text-content-ratio, and average motion based on camera pan and tilt estimation; and the other four features are extracted from the sound track including: music-ratio, speech-ratio, silence-ratio, and non-literal sound (NL-) ratio.

Let V be any arbitrary video, and T be the number of shots in V . Feature extraction from V will result in a sequence of observations of length T where each observation is a column vector of seven features. We collect nine videos in total to use for training purposes with T ranging from 124 to 245. We analyse the learned model and present the results at two levels.

6.4.1 Semantic Mappings at the Production Level

We use the mean matrix at the production level to understand the mapping between the production level states and the feature vector components. Figure-(6-9) shows the visualisation of the means. Thus, for example, state 1 is strongly linked to face and speech, and a little bit of motion. This state corresponds to direct and assisted narration sections. State 2 corresponds to speech and motion, and thus to voice-over sections. State 3 corresponds to NL-sound, and thus to expressive linkage sections and so on. In Table-(6.2), we summarise this result of mapping the production level states to the structural units manually crafted in our earlier work. The mean values learned at the production level are visualised in Figure-(6-9).

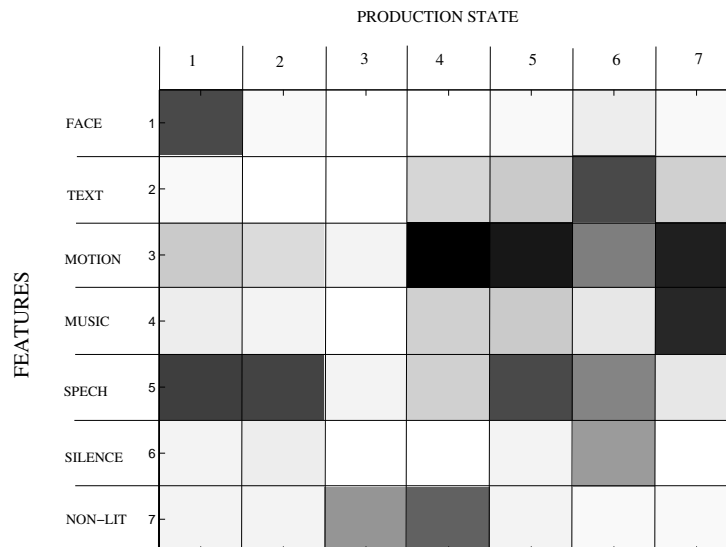


Figure 6-9: Gray-scaled visualisation of the mean values of seven features learned at the production level.

State	State Feature	Structural Units
1	face, speech, motion	direct-, assisted- narration
2	speech, motion	<i>pure</i> voice-over (VO)
3	NL-sound	expressive linkage
4	motion, NL-sound, text	linkage, VO with texts
5	motion, speech, music	expressive linkage
6	text, motion, speech	VO, functional linkage
7	motion, music	dramatic/expressive linkage

Table 6.2: Deduced semantic mappings at production (shot) level.

6.4.2 Semantic Mappings at the Upper Level

The middle level of topology (Figure-(6-8)) contains four states. Combining the information of the ‘parent-child’ relationship specified in the topology and the results from Table 6.2, we can interpret the meaning of each state. However, this must be done in conjunction with the estimated π matrix since it specifies the probability of a child being called.

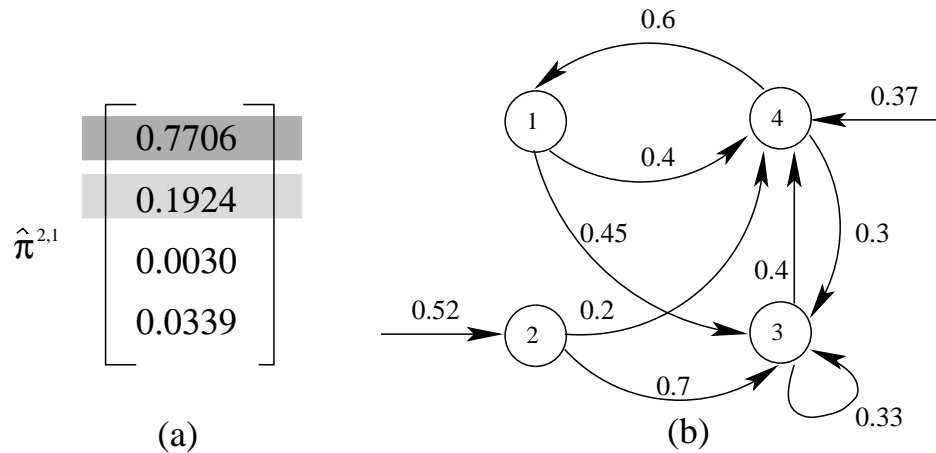


Figure 6-10: Estimated transition probability at the upper level. Only dominant probabilities are shown.

For example, from Figure-(6-8), state 1 has four children at the production level, namely 1, 2, 4 and 7. However, the estimated initial probability $\hat{\pi}^{2,1}$ shown in Figure-6-10(a) indicates that child states 4 and 7 are almost disconnected from their parent state 1. This allows us to attach this parent state with only the on-screen section or with the pure voice-over section (with a lower probability). Similar analysis shows that state 2 is connected with expressive linkage sections or voice-over; state 3 is connected with expressive voice-over (ie: voice over with no texts and lots of motion); state 4 is connected with functional linkage or sometimes with voice-over sections. Overlapping is observed with the voice-

over sections, however, they have different probabilities. A transition matrix for these four states is shown in Figure-6-10(b). While we cannot deduce the complete ‘style’ of educational video from this result, it does suggest a ‘common’ presentational style for this genre of video. In this case, written in regular grammar, it is³:

$$[2|4][4,1]^*[3]^+[4,1]^*[1,4]^*[3]^+$$

At this stage, we cannot make conclusions about the power of this type of grammar⁴. However, our initial results from the Viterbi decoding suggests a very interesting point. The sequence of states and time indices decoded at the middle level for the video ‘Eye-Safety’ is: $\begin{bmatrix} 4 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 5 \end{bmatrix} \begin{bmatrix} 4 \\ 14 \end{bmatrix} \begin{bmatrix} 1 \\ 15 \end{bmatrix} \begin{bmatrix} 4 \\ 16 \end{bmatrix} \begin{bmatrix} 3 \\ 17 \end{bmatrix} \begin{bmatrix} 4 \\ 18 \end{bmatrix} \begin{bmatrix} 1 \\ 22 \end{bmatrix} \begin{bmatrix} 4 \\ 23 \end{bmatrix} \begin{bmatrix} 1 \\ 24 \end{bmatrix} \begin{bmatrix} 4 \\ 25 \end{bmatrix} \begin{bmatrix} 3 \\ 28 \end{bmatrix} \begin{bmatrix} 4 \\ 44 \end{bmatrix} \begin{bmatrix} 1 \\ 45 \end{bmatrix} \begin{bmatrix} 4 \\ 49 \end{bmatrix} \begin{bmatrix} 1 \\ 50 \end{bmatrix} \begin{bmatrix} 3 \\ 52 \end{bmatrix}$
 $\begin{bmatrix} 4 \\ 57 \end{bmatrix} \begin{bmatrix} 1 \\ 58 \end{bmatrix} \begin{bmatrix} 4 \\ 65 \end{bmatrix} \begin{bmatrix} 1 \\ 67 \end{bmatrix} \begin{bmatrix} 4 \\ 70 \end{bmatrix} \begin{bmatrix} 3 \\ 71 \end{bmatrix} \begin{bmatrix} 4 \\ 72 \end{bmatrix} \begin{bmatrix} 3 \\ 73 \end{bmatrix} \begin{bmatrix} 1 \\ 91 \end{bmatrix} \begin{bmatrix} 3 \\ 94 \end{bmatrix} \begin{bmatrix} 4 \\ 103 \end{bmatrix} \begin{bmatrix} 1 \\ 104 \end{bmatrix} \begin{bmatrix} 3 \\ 109 \end{bmatrix} \begin{bmatrix} 4 \\ 117 \end{bmatrix} \begin{bmatrix} 1 \\ 118 \end{bmatrix} \begin{bmatrix} 3 \\ 119 \end{bmatrix} \begin{bmatrix} 1 \\ 124 \end{bmatrix}$, where each entry is a two-tuple $\begin{bmatrix} s \\ t \end{bmatrix}$ where s is the state activated at time t . A useful state pattern from the above the grammar is $[41]$ or $[14]$ (ie: a state 1 is followed by 4 or vice versa). We observe this pattern in this video at times 16, 23, 25, 45, ... or 118. These correspond to subtopic boundaries. In fact, when compared with the groundtruth for the video ‘Eye-Safety’, all subtopic boundaries correspond to this pattern, except one miss and one false positive.

To sum up, in this section we have utilised the hierarchic modeling of the HHMM to automatically exploit the structural units in an educational video. We have analysed the hierarchy constructed in this data-driven approach, and our results indicate the promise of this approach for mining semantics and the discovery of structural information in video data.

6.5 Closing Remarks

In this chapter, we have demonstrated the application of the Hierarchical HMM to the problem of semantic analysis for educational films. From the theoretical aspect, we develop a generalised Viterbi algorithm needed in the segmentation stage when the problem is based on a HHMM framework. In addition, we also address the parameter estimation problem when the observation is continuous. We then report two applications of the HHMMs: the first is for the problem of subtopic detection, and the second is for automatically learning meaningful semantic structures in the videos. While these results are specific to the educational domain, they have demonstrated the feasibility of potential larger scale deployment of the HHMM-based framework to hierarchical modeling of videos.

³Note that together with the probabilities specified in Figure-6-10(b), this grammar should be viewed as Probabilistic Regular Grammar.

⁴We also notice the absence of state 2 in the middle part of this grammar. This is somewhat due to the rare appearance of expressive linkage sections.

Chapter 7

Conclusions

This thesis has presented an investigation of Film Grammar based and probabilistic methods for the analysis of educational videos, a domain as yet scarcely explored, and of great interest to e-learning and training services.

Our contributions from this thesis can be summarised into two parts: (1) a Film Grammar based analysis to the problem of annotation and segmentation of educational films presented in Chapter 3 and Chapter 4, and (2) the theoretical extension to the HHMM to handle shared structures in Chapter 5, and its application to segmenting and learning semantic concepts in educational films presented in Chapter 6.

Chapter 3 contributes our first study to understanding the specific ‘grammars’ of educational films and proposes a hierarchy of meaningful narrative structures for this film genre. The hierarchy contains nine useful semantic concepts organised into two layers which provides vital information of the content of educational material. The top layer includes main narrative structures such as *on-screen narration* or *voice-over*, and the lower level includes finer narrative divisions such as *direct narration* or *voice-over with texts*. We then develop a set of useful visual and aural features which are used in a study to learn decision trees for automatically classifying a shot into one of the labels contained in the hierarchy. The first experimental results have shown that important structures at the top level have been recognised with an acceptable accuracy, while the performances within a group (eg: voice-over narration) are poor. We then propose a two-tiered classification procedure, which has demonstrated improvements in the classification performance.

Continuing on from Chapter 3, we delve deeper into Film Grammar in Chapter 4 to seek higher-order semantics in educational films, hypothesising their narrative ‘effects’, and using them for segmentation of topical content at two conceptual levels: *main topic* and *subtopic*. We first propose a computational form for the extraction of the *content density* function as a measure of the ‘information delivery rate’, and then study the be-

haviours of this function in relation to the subtopic boundary transitions. Two (heuristic and Bayesian) methods are then proposed for the problem of subtopic detection. The experimental results have demonstrated the validity of these algorithms, in which the probabilistic method performs slightly better with a recall of 78.3% and a precision of 83.4%.

Next, we extract two useful expressive elements relating to the ‘mediation process’ of the video-makers, namely the *thematic* and *dramatic* functions. In an experiment, we set out to study key contributing elements to these functions and then formulate their computational forms. Incorporating the knowledge of educational films, the thematic function is then used, in conjunction with the content density function, in an edge-based detection algorithm to segment a video into main topics and subtopics. The experimental results on ten industrial instructional videos report a recall of 75.8% and a precision of 86.2% in detecting main topic boundaries. This has demonstrated the usefulness of our expressive function set and how the knowledge of film analysis can provide important clues about the structure of the video.

Our next major contribution contained in Chapter 5 is the theoretical extension to the original Hierarchical Hidden Markov Model in (Fine *et al.*, 1998) to have arbitrary shared substructures. This approach is motivated by our key observation that video domain does not only possess a natural hierarchical decomposition, but also many substructures are naturally shared and inherited in the hierarchy. We have thoroughly investigated this model. We first introduce the concept of ‘shared’ structures and provide formal definitions of the extended HHMM that generalises the original HHMM (Fine *et al.*, 1998). We then discuss the problem of inference and learning in this model. In particular, our key contributions from this chapter are the following:

- A set of conditional independencies formally exploited in the DBN representation of the HHMMs. This provides a mathematical background for many inference algorithms developed at the later stage.
- A novel Asymmetric Inside-Outside algorithm to perform inference on the model, and an EM parameter estimation procedure for learning.
- A novel scaling algorithm is developed to avoid numerical underflow, an issue that needs to be solved in order for the HHMM to deal with long observation sequences.

We have shown that with the AIO algorithm, we can achieve the same complexity for inference and learning for the extended HHMM compared with the results in the original paper (Fine *et al.*, 1998).

Our final contribution is contained in Chapter 6 in which the HHMM is applied to the problem of semantic analysis in educational videos. In the first part of this chapter, we present a novel generalised decoding Viterbi algorithm needed to enable the segmentation process in the HHMM by computing the best sequence of states and their corresponding ending status. Parameter estimation when the emission probability is modeled as a mixture of Gaussians is also addressed. These issues need to be solved so that the HHMM can be used to segment real video data when the observations (eg: colour, sound energy, shot length, etc.) are continuous in nature. We then present two applications of the HHMM. In the first application, we tailor a three-layer HHMM whose topology is constructed based on our prior domain knowledge, to detect subtopic boundary transitions. Even though the model is simple, the initial results are promising with 77.3% for recall and 70.7% for precision when testing on a set of eight videos with similar styles of presentation. In the second application of the HHMM, we aim to exploit the expressiveness of shared structures in the HHMM to automatically learn meaningful structural units in educational videos. The topology in this experiment is richer than in the first one with tighter shared structures. It is however, again constructed based on our assumption about the prior structural information in relation to the videos. The experimental results have shown that many meaningful concepts can be hierarchically learned from this model. For example, at the bottom level, we can deduce concepts such as ‘expressive linkage’ or ‘face and motion’ (a kind of on-screen narration) and at the top level concepts such as ‘pure voice-over’ or ‘functional linkage’ can be inferred. Some overlaps in semantic interpretation do, however, exist. Nevertheless, the experiment demonstrates the feasibility and promise of a semantic discovery framework based on the HHMM.

7.1 Future Directions

In this section, we discuss some speculative ideas and possible future extensions to the work presented here.

Modern techniques allow multimedia data nowadays to include extra information such as closed captions that contain the scripts of the video. It is worth investigating how to take into account this extra information into account for the problem of annotation and segmentation. When no closed captions are available, the use of speech recognition to spot key words presents another aspect to explore. For example, in safety videos, keywords such as ‘**remember**’ or ‘**should**’ are used repeatedly to emphasise instructional messages.

Closely related to educational videos investigated in this thesis are other informational categories of films such as documentaries. It would be interesting to see how methods and

techniques developed in this thesis can be applied and/or generalised to these domains. The dramatic and thematic functions, for example, will be useful to uncover the content and structure of a documentary.

The work done in this thesis implies the potential construction of content management systems for educational videos that allow users to browse in a table-of-content style or formulate semantically rich queries such as “find me all on-screen segments” or “find me segments where instructional messages are given directly”.

With respect to the work done for HHMMs in this thesis, there are a number of useful directions that can be considered for future work. Firstly, the EM learning for the HHMM in this thesis is solved in the most general case, assuming the ending status of every sub-HMM is unknown and thus treating $\{e_{1:T}^{2:D-1}\}$ as hidden variables. This learning framework can thus be readily extended to solve for the *partially observed time indices* case. That is, in the training data, together with the observation $\{y_{1:T}\}$, an arbitrary *subset* of $\{e_{1:T}^{2:D-1}\}$ is also observed. This will result in an attractive framework in which (sparse) information about the ending status of any sub-HMM is incorporated directly in the model, and thus helps to improve the learning accuracy and training speed. It is not difficult to see that when the *complete* set of ending variables $\{e_{1:T}^{2:D-1}\}$ is observed, training the HHMM in this case can be done in a level-by-level manner, similar to the common method of ‘stacking’ HMMs as reviewed in Subsection 2.3.3.4. The training data in this case is, however, required to be *completely pre-segmented* at all levels (but not necessarily annotated), which is, in practice, either not available, or extremely time-consuming to prepare. Training the HHMM with partially observed time indices *does not require the data to be completely pre-segmented*, and any temporal information that is available will be taken care of. This extension would also present a very attractive model in other domains such as behaviour recognition or robot navigation when the temporal information of an action or behaviour is observed partially.

The second open issue is the problem of *model selection* for the HHMM to automatically learn the topology of the model. The applications of the HHMM in this thesis have used the domain knowledge of educational videos to help in constructing the topology. To deploy the HHMM in a larger scale and across different domains, it would be attractive if the topology of the HHMM can be automatically learned from training data. This presents, in general, the same problem of model selection in Bayesian Networks. Using the MCMC technique with some split and merge procedure as presented in (Xie *et al.*, 2002a) or using Gibbs sampling offers some immediate solutions. Alternatively, one can take advantage of modeling the shared structures in this thesis to learn the topology in a semi-supervised manner as follows. First, given the depth D and the number of states at each level, we construct a *fully shared topology*, ie: all of the states at the lower level are

shared by all states at the upper level. The AIO-algorithm is then used to estimate the parameters from the training data. Based on the estimated parameters, the topology will be pruned and/or refined.

Lastly, it would be interesting to study the relationship between Probabilistic Context Free Grammar (PCFG) and the HHMM. If it can be shown (which is our conjecture) that there is a one-to-one mapping between PCFG (or at least a subclass of PCFG) and the HHMM, then work developed in this thesis would imply an alternative way to the traditional ways of modeling, learning and parsing in the PCFG community. A very interesting fact that would emerge is the scaling algorithm developed in this thesis. To the best of our knowledge, such scaling schemes have never been addressed in the computational linguistics research community.

Appendix A

Proof of Selected Theorems

Proof for Theorem 5.3 (page 111)

Let \vec{c} and \vec{z} be two M -dimension vectors, whose elements are non-negative: $\vec{c} = \{c_i \mid c_i \geq 0\}_{i=1}^M$, $\vec{z} = \{z_i \mid z_i \geq 0\}_{i=1}^M$. Let the objective function be:

$$f(\vec{z}) = \vec{c} \cdot \log \vec{z} = \sum_{i=1}^M c_i \log z_i$$

with the constraint: $\sum_{i=1}^M z_i = 1$. Then $f(\vec{z})$ is maximised for: $\hat{z}_i = c_i / \sum_{i=1}^M c_i$.

Proof. Adding the Lagrange multiplier λ into the objective function, we have:

$$f(\vec{z}) = \sum_{i=1}^M c_i \log z_i + \lambda(1 - \sum_{i=1}^M z_i)$$

Taking derivative with respect to each z_i and set to 0, we have:

$$\begin{aligned} \frac{\partial f(\vec{z})}{\partial z_i} &= \frac{c_i}{z_i} - \lambda = 0 \\ \implies z_i &= \frac{c_i}{\lambda} \\ \implies 1 &= \sum_{i=1}^M z_i = \sum_{i=1}^M \frac{c_i}{\lambda} \implies \lambda = \sum_{i=1}^M c_i \end{aligned} \tag{A.1}$$

Substituting λ back into Equation-(A.1), we have: $\hat{z}_i = \frac{c_i}{\lambda} = \frac{c_i}{\sum_{i=1}^M c_i}$ ■

Appendix B

Proofs of Formulas

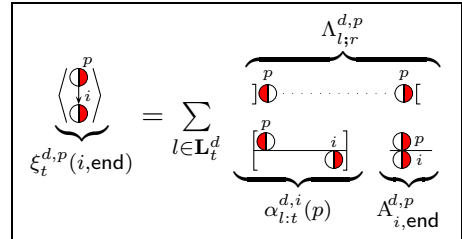
In this appendix, we provide formal proofs for the set of unproven formulas developed in Chapter 5. In addition, where possible we will show the calculation in matrix forms so that it can be used in Matlab efficiently. To avoid lengthy expansion, we frequently factorise a joint probability according to the known conditional independencies. For example, if we know $A \perp\!\!\!\perp \{C, D\} \mid B$ and $B \perp\!\!\!\perp D \mid C$, then we write:

$$\begin{aligned} \Pr(A, B, C, D) &\stackrel{(a)}{=} \Pr(A \mid \text{REST}) \Pr(B, C, D) \\ &\stackrel{(b)}{=} \Pr(A \mid B) \Pr(B \mid \text{REST}) \Pr(C \mid D) \Pr(D) \\ &= \Pr(A \mid B) \Pr(B \mid C) \Pr(C \mid D) \Pr(D) \end{aligned}$$

where the term ‘REST’ in step (a) is implicitly understood to be $\{B, C, D\}$ and REST in step (b) is $\{C, D\}$.

Equation-(5.57) (page 125)

$$\xi_t^{d,p}(i, \text{end}) = \sum_{l \in \mathbf{L}_t^d} \left[\alpha_{l:t}^{d,p}(i) A_{i,\text{end}}^{d,p} \right] \Lambda_{l;t}^{d,p}$$



Proof. Using the definition, summing over the starting time l of the parent p and factorising

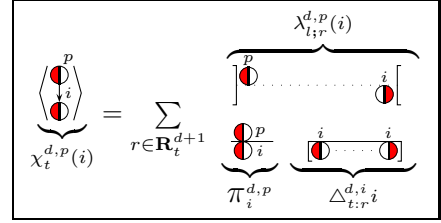
according to the diagram, we have:

$$\begin{aligned}
\xi_t^{d,p}(i, \text{end}) &\triangleq \Pr(x_t^d = p, x_t^{d+1} = i, e_t^{d:d+1} = 11, \mathcal{O}) = \sum_{l \in \mathbf{L}_t^d} \Pr(x_t^d = p, \cdot \tau_t^d = l, x_t^{d+1} = i, e_t^{d:d+1} = 11, \mathcal{O}) \\
&\stackrel{(a)}{=} \sum_{l \in \mathbf{L}_t^d} \Pr(\mathcal{O}_{l;t}^{\text{out}} \mid \text{REST}) \Pr(x_t^d = p, e_{l;t-1}^d = \mathbf{0}, x_t^{d+1} = i, e_t^{d:d+1} = 11, \mathcal{O}_{l;t}^{\text{in}}) \\
&\stackrel{(b)}{=} \sum_{l \in \mathbf{L}_t^d} \frac{\Lambda_{l;t}^{d,p}}{\Pr(\cdot x_t^d = p)} \Pr(e_t^d = 1 \mid \text{REST}) \Pr(\mathcal{O}_{l;t}^{\text{in}}, x_t^{d+1} = i, e_{l;t-1}^d = \mathbf{0} \mid \cdot x_t^d = p) \Pr(\cdot x_t^d = p) \\
&= \sum_{l \in \mathbf{L}_t^d} \Lambda_{l;t}^{d,p} A_{i,\text{end}}^{d,p} \alpha_{l;t}^{d,p}(i)
\end{aligned}$$

where in step (a) clearly the observation ‘outside’ $\mathcal{O}_{l;t}^{\text{out}}$ only depends on the symmetric boundary ‘guarded’ by p (by Theorem 5.1), therefore $\Pr(\mathcal{O}_{l;t}^{\text{out}} \mid \text{REST}) = \Lambda_{l;t}^{d,p} / \Pr(\cdot x_t^d = p)$; and in step (b) by Lemma 5.1, $\Pr(e_t^d = 1 \mid \text{REST}) = A_{i,\text{end}}^{d,p}$. ■

Equation-(5.58) (page 125)

$$\chi_t^{d,p}(i) = \pi_i^{d,p} \left[\sum_{r \in \mathbf{R}_t^{d+1}} \lambda_{t;r}^{d,p}(i) \Delta_{t;r}^{d+1,i} \right]$$



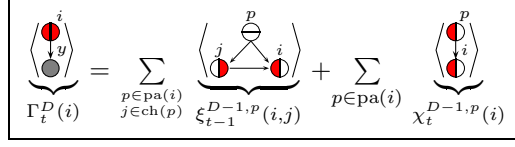
Proof. Using the definition of the vertical transition probability variable $\chi_t^{d,p}(i)$, and summing over the ending time of the child i , we have:

$$\begin{aligned}
\chi_t^{d,p}(i) &\triangleq \Pr(x_t^d = p, x_t^{d+1} = i, e_{t-1}^{d:d+1} = 11, \mathcal{O}) = \sum_{r \in \mathbf{R}_t^{d+1}} \Pr(x_t^d = p, x_t^{d+1} = i, \tau_t^{d+1} = r, e_{t-1}^{d:d+1} = 11, \mathcal{O}) \\
&\stackrel{(a)}{=} \sum_{r \in \mathbf{R}_t^{d+1}} \Pr(\mathcal{O}_{t;r}^{\text{out}} \mid \text{REST}) \Pr(x_t^d = p, x_t^{d+1} = i, e_{t;r-1}^{d+1} = \mathbf{0}, e_{t-1}^{d:d+1} = 11, \mathcal{O}_{t;r}^{\text{in}}) \\
&\stackrel{(b)}{=} \sum_{r \in \mathbf{R}_t^{d+1}} \frac{\lambda_{t;r}^{d,p}(i)}{\Pr(\cdot x_t^d = p)} \underbrace{\Pr(\mathcal{O}_{t;r}^{\text{in}}, e_{t;r-1}^{d+1} = \mathbf{0}, e_r^{d+1} = 1 \mid \text{REST})}_{\Delta_{t;r}^{d+1,i}} \Pr(x_t^{d+1} = i \mid \cdot x_t^d = p) \Pr(\cdot x_t^d = p) \\
&= \sum_{r \in \mathbf{R}_t^{d+1}} \lambda_{t;r}^{d,p}(i) \Delta_{t;r}^{d+1,i} \pi_i^{d,p}
\end{aligned}$$

Again, by Theorem 5.2 in step (a) the ‘outside’ observation $\mathcal{O}_{t;r}^{\text{out}}$ only depends on the asymmetric boundary ‘guarded’ by p and i and hence the term $\lambda_{t;r}^{d,p}(i)$ is recovered in the next step; and in step (b) the symmetric inside variable $\Delta_{t;r}^{d+1,i}$ is obtained by Lemma 5.1. ■

Equation-(5.59) (page 125)

$$\Gamma_t^D(i) = \sum_{p \in \text{pa}(i)} \left(\sum_{j \in \text{ch}(p)} \xi_{t-1}^{D-1,p}(j, i) + \chi_t^{D-1,p}(i) \right)$$



Proof. We sum over the parent p of i and consider the value of e_{t-1}^{D-1} , that is we write:

$$\begin{aligned} \Gamma_t^D(i) &\triangleq \Pr(x_t^D = i, \mathcal{O}) \\ &= \sum_{p \in \text{pa}(i)} (\Pr(x_t^{D-1} = p, e_{t-1}^{D-1} = 0, x_t^{D-1} = i, \mathcal{O}) + \Pr(x_t^{D-1} = p, e_{t-1}^{D-1} = 1, x_t^{D-1} = i, \mathcal{O})) \end{aligned}$$

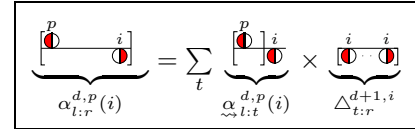
The first term (when $e_{t-1}^{D-1} = 1$) corresponds to $\chi_t^{D-1,p}(i)$. For the second term (when $e_{t-1}^{D-1} = 0$), we sum over the state j before i and note that by the model assumption a state at level D starts and ends in a single time slice:

$$\begin{aligned} \Pr(x_t^{D-1} = p, e_{t-1}^{D-1} = 1, x_t^{D-1} = i, \mathcal{O}) &= \sum_{j \in \text{ch}(p)} \Pr(x_t^{D-1} = p, e_{t-1}^{D-1} = 0, x_{t-1}^{D-1} = j, x_t^{D-1} = i, \mathcal{O}) \\ &= \sum_{j \in \text{ch}(p)} \xi_{t-1}^{D-1,p}(j, i) \end{aligned}$$

Substituting back into the previous equation, we obtain the required proof. ■

Equation-(5.60) (page 126)

$$\alpha_{l:r}^{d,p}(i) = \sum_{t=l}^r \alpha_{l:t}^{d,p}(i) \Delta_{t:r}^{d+1,i}$$



Proof. Summing over the starting time t of the child i and following the factorisation form in

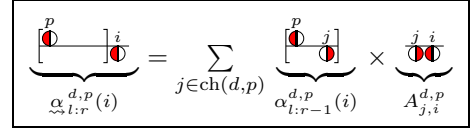
the diagram, we write:

$$\begin{aligned}
\alpha_{l:r}^{d,p}(i) &\triangleq \Pr(y_{l:r}, x_r^{d+1} = i, e_{l:r-1}^d = \mathbf{0} \mid \cdot x_l^d = p) \\
&= \sum_{l \leq t \in \mathbf{L}_r^{d+1}} \Pr(y_{l:r}, \cdot x_t^{d+1} = i, e_{t:r-1}^{d+1} = \mathbf{0}, e_r^{d+1} = 1, e_{l:r-1}^d = \mathbf{0} \mid \cdot x_l^d = p) \\
&= \sum_{l \leq t \in \mathbf{L}_r^{d+1}} \Pr(y_{t:r}, e_{t:r-1}^{d+1} = \mathbf{0}, e_r^{d+1} = 1 \mid \text{REST}) \Pr(y_{l:t-1}, \cdot x_t^{d+1} = i, e_{l:r-1}^d = \mathbf{0} \mid \cdot x_l^d = p) \\
&= \sum_{l \leq t \in \mathbf{L}_r^{d+1}} \underset{\sim}{\alpha}_{l:t}^{d,p}(i) \Delta_{t:r}^{d+1,i}
\end{aligned}$$

Note that the recursion occurs at two places: at previous time-slice in $\underset{\sim}{\alpha}_{l:t}^{d,p}(i)$, and at the lower level at $\Delta_{t:r}^{d+1,i}$. ■

Equation-(5.61) (page 126)

$$\underset{\sim}{\alpha}_{l:r}^{d,p}(i) = \sum_{j \in \text{ch}(d,p)} \alpha_{l:r-1}^{d,p}(j) A_{j,i}^{d,p}$$



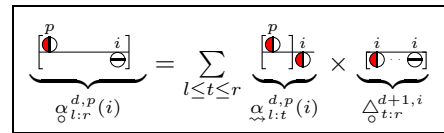
Proof. This started-forward variable $\underset{\sim}{\alpha}$ should be viewed as a *convenient* quantity. It is calculated directly from α and involves no recursion. The diagram suggests an immediate answer:

$$\begin{aligned}
\underset{\sim}{\alpha}_{l:r}^{d,p}(i) &\triangleq \Pr(y_{l:r-1}, \cdot x_r^{d+1} = i, e_{l:r-1}^d = \mathbf{0} \mid \cdot x_l^d = p) \\
&= \sum_{j \in \text{ch}(d,p)} \Pr(x_{r-1}^{d+1} = j, y_{l:r-1}, \cdot x_r^{d+1} = i, e_{l:r-1}^d = \mathbf{0} \mid \cdot x_l^d = p) \\
&= \sum_{j \in \text{ch}(d,p)} \Pr(e_{r-1}^d = 0, \cdot x_r^{d+1} = i \mid \text{REST}) \Pr(y_{l:r-1}, x_{r-1}^{d+1} = j, e_{l:r-2}^d = \mathbf{0} \mid \cdot x_l^d = p) \\
&= \sum_{j \in \text{ch}(d,p)} \alpha_{l:r-1}^{d,p}(j) A_{j,i}^{d,p}
\end{aligned}$$

In matrix form: $\underset{\sim}{\alpha}_{l:r}^{d,p} = \alpha_{l:r-1}^{d,p} A_{[i]}^{d,p}$ ■

Equation-(5.68) (page 131)

$$\underset{\circ}{\alpha}_{l:r}^{d,p}(i) = \sum_{t=l}^r \underset{\sim}{\alpha}_{l:t}^{d,p}(i) \Delta_{\circ t:r}^{d+1,i}$$



Proof. Following the same line of algebra as in the case of the finished version, we have:

$$\begin{aligned}
\alpha_{l:r}^{d,p}(i) &\triangleq \Pr(y_{l:r}, x_r^{\circ d+1} = i, e_{l:r-1}^d = \mathbf{0} \mid \cdot x_l^d = p) \\
&= \sum_{t=l}^r \Pr(y_{l:r}, \cdot x_t^{d+1} = i, e_{t:r-1}^{d+1} = \mathbf{0}, e_r^{d+1} = 0, e_{l:r-1}^d = \mathbf{0} \mid \cdot x_l^d = p) \\
&= \sum_{t=l}^r \Pr(y_{t:r}, e_{t:r-1}^{d+1} = \mathbf{0}, e_r^{d+1} = 0 \mid \text{REST}) \Pr(y_{l:t-1}, \cdot x_t^{d+1} = i, e_{l:r-1:d}^0 \mid \cdot x_l^d = p) \\
&= \sum_{t=l}^r \alpha_{l:t}^{d,p}(i) \triangle_{\circ t:r}^{d+1,i}
\end{aligned}$$

Clearly, the results are the same as in the case of $\alpha_{l:r}^{d,p}(i)$, except that Δ is replaced by \triangle_{\circ} . \blacksquare

Equation-(5.62b) (page 127)

$$\Delta_{l:r}^{d,i} = \sum_{s \in \text{ch}(i)} \alpha_{l:r}^{d,i}(s) A_{s,\text{end}}^{d,i}$$

Proof. Summing over the child s of state i at time r and following the form of factorisation in the diagram, we have:

$$\begin{aligned}
\Delta_{l:r}^{d,i} &\triangleq \Pr(y_{l:r}, e_{l:r-1}^d = \mathbf{0}, e_r^d = 1 \mid \cdot x_l^d = i) \\
&= \sum_{s \in \text{ch}(i)} \Pr(y_{l:r}, e_{l:r-1}^d = \mathbf{0}, e_r^d = 1, x_r^{d+1} = s \mid \cdot x_l^d = i) \\
&= \sum_{s \in \text{ch}(i)} \Pr(e_r^d = 1 \mid \text{REST}) \Pr(y_{l:r}, x_r^{d+1} = s, e_{l:r-1}^d = \mathbf{0} \mid \cdot x_l^d = i) \\
&= \sum_{s \in \text{ch}(i)} \alpha_{l:r}^{d,i}(s) A_{s,\text{end}}^{d,i}
\end{aligned}$$

Again there is no recursion in Δ itself, it is computed directly from α . And, in matrix form:

$$\Delta_{l:r}^{d,i} = \alpha_{l:r}^{d,i} A_{[\text{end}]}^{d,i}. \quad \blacksquare$$

Equation-(5.70) (page 131)

$$\Delta_{\circ l:r}^{d,i} = \begin{cases} 0 & \text{if } d = D \\ \sum_{s \in \text{ch}(i)} \alpha_{l:r}^{d,i}(s) (1 - A_{s,\text{end}}^{d,i}) + \alpha_{l:r}^{d,i}(s) & \text{if } d < D \end{cases}$$

Proof. When $d = D$, by definition (Equation-(5.69)), we have:

$$\Delta_{\circlearrowleft l:r}^{D,i} = \Pr(y_r, e_r^D = 0 \mid x_r^D = i) = 0$$

It equates to zero since $e_r^D = 0$ is *inconsistent* with the model definition which states that $e_t^D = 1, \forall t$. When $d < D$, we sum over the child-state s of i at time r and consider the binary value of e_r^{d+1} . If $e_r^{d+1} = 1$, the form of factorisation is given as in the diagram, in which $\alpha_{l:r}^{d,i}(s)$ is recovered; and when $e_r^{d+1} = 0$, it reduces to $\alpha_{l:r}^{d,i}(s)$. The algebra is:

$$\begin{aligned} \Delta_{\circlearrowleft l:r}^{d,i} &\triangleq \Pr(y_{l:r}, e_{l:r}^d = \mathbf{0} \mid \cdot x_l^d = i) = \sum_{s \in \text{ch}(i)} \Pr(y_{l:r}, e_{l:r}^d = \mathbf{0}, x_r^{d+1} = s \mid \cdot x_l^d = i) \\ &= \sum_{s \in \text{ch}(i)} \left[\underbrace{\Pr(e_r^{d+1} = 1, y_{l:r}, e_{l:r}^d = \mathbf{0}, x_r^{d+1} = s \mid \cdot x_l^d = i)}_A + \underbrace{\Pr(e_r^{d+1} = 0, y_{l:r}, e_{l:r}^d = \mathbf{0}, x_r^{d+1} = s \mid \cdot x_l^d = i)}_B \right] \end{aligned}$$

Now factorise A according to the diagram:

$$\begin{aligned} A &= \Pr(y_{l:r}, e_{l:r-1}^d = \mathbf{0}, e_r^d = \mathbf{0}, x_r^{d+1} = s, e_r^{d+1} = 1 \mid \cdot x_l^d = i) \\ &= \Pr(e_r^d = 0 \mid \text{REST}) \Pr(y_{l:r}, e_{l:r-1}^d = \mathbf{0}, x_r^{d+1} = s \mid \cdot x_l^d = i) = \left(1 - A_{s,\text{end}}^{d,i}\right) \alpha_{l:r}^{d,i}(s) \end{aligned}$$

And in B , since $e_r^{d+1} = 0$ implies $e_r^d = 0$, therefore:

$$\begin{aligned} B &= \Pr(y_{l:r}, e_{l:r-1}^d = \mathbf{0}, e_r^d = \mathbf{0}, x_r^{d+1} = s, e_r^{d+1} = 0 \mid \cdot x_l^d = i) \\ &= \Pr(y_{l:r}, e_{l:r-1}^d = \mathbf{0}, x_r^{d+1} = s \mid \cdot x_l^d = i) = \alpha_{l:r}^{d,i}(s) \end{aligned}$$

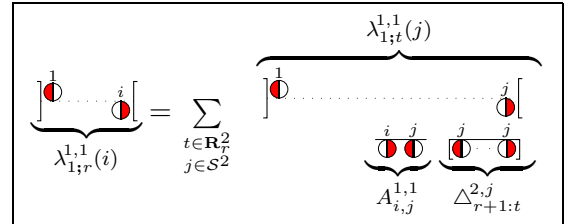
Combining results from A and B , the proof is attained. In matrix form, we write:

$$\Delta_{\circlearrowleft l:r}^{d,i} = \alpha_{l:r}^{d,i} \left(1 - A_{[\text{end}]}^{d,i}\right) + \text{sum} \left(\alpha_{l:r}^{d,i}\right)$$

where the $\text{sum}(\cdot)$ operator is the same as $\sum_{s \in \text{ch}(i)} \alpha_{l:r}^{d,i}(s)$. ■

Equation-(5.63c) (page 128)

$$\lambda_{1;r}^{1,1}(i) = \sum_{t \in \mathbf{R}_r^2} \sum_{j \in \mathbf{S}^2} \lambda_{1;t}^{1,1}(j) \Delta_{r+1:t}^{2,j} A_{i,j}^{1,1} \quad \text{for } r < T$$



Proof. Writing down the definition in Equation-(5.47c), and making a note that the root state

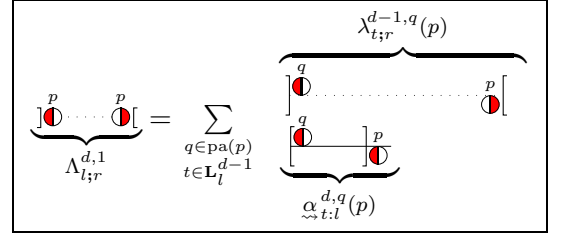
always starts at the first time slice, ie: $\Pr(\cdot x_1^1 = 1) = 1$, we have:

$$\begin{aligned}
\lambda_{1;r}^{1,1}(i) &\triangleq \Pr(e_T^1 = 1, y_{r+1:T} \mid \cdot x_1^1 = 1, e_{1:r-1}^1 = \mathbf{0}) \Pr(\cdot x_1^1 = 1) \\
&= \sum_{j \in \mathcal{S}^2} \Pr(e_T^1 = 1, y_{r+1:T}, \cdot x_{r+1}^2 = j \mid \cdot x_1^1 = 1, e_{1:r-1}^1 = \mathbf{0}) \\
&= \sum_j \sum_{t \in \mathbf{R}_r^2} \Pr(e_T^1 = 1, y_{r+1:T}, \cdot x_{r+1}^2 = j, e_{r+1:t-1}^2 = \mathbf{0}, e_t^2 = 1 \mid \cdot x_1^1 = 1, e_{1:r-1}^1 = \mathbf{0}) \\
&\stackrel{(a)}{=} \sum_{j,t} \underbrace{\Pr(y_{t+1:T}, e_T^1 = 1 \mid \text{REST})}_{\lambda_{1;t}^{1,1}(j)} \Pr(y_{r+1:t}, \cdot x_{r+1}^2 = j, e_{r+1:t-1}^2 = \mathbf{0}, e_t^2 = 1 \mid \cdot x_1^1 = 1, x_r^2 = i) \\
&\stackrel{(b)}{=} \sum_{j,t} \lambda_{1;t}^{1,1}(j) \underbrace{\Pr(y_{r+1:t}, e_{r+1:t-1}^2 = \mathbf{0}, e_t^2 = 1 \mid \text{REST})}_{\Delta_{r+1:t}^{2,j}} \Pr(\cdot x_{r+1}^2 = j \mid \underbrace{\cdot x_1^1 = 1, x_r^2 = i, e_{1:r-1}^1 = \mathbf{0}}_W) \\
&= \sum_{j,t} \lambda_{1;t}^{1,1}(j) \Delta_{r+1:t}^{2,j} \left[\underbrace{\Pr(\cdot x_{t+1}^2 = j, e_r^1 = 0 \mid W)}_{A_{i,j}^{1,1}} + \underbrace{\Pr(\cdot x_{t+1}^2 = j, e_r^1 = 1 \mid W)}_{= 0 \text{ (since } e_r^1 = 0 \text{ by definition)}} \right] \\
&= \sum_{t \in \mathbf{R}_r^2} \sum_{j \in \mathcal{S}^2} \lambda_{1;t}^{1,1}(j) \Delta_{r+1:t}^{2,j} A_{i,j}^{1,1}
\end{aligned}$$

We note that the terms $\lambda_{1;t}^{1,1}(j)$ and $\Delta_{r+1:t}^{2,j}$ recovered in steps (a) and (b) respectively are due to Theorem 5.2 and Theorem 5.1. They guarantee the fact that when conditioning on the ‘REST’, only the boundaries, $\text{AB}_{1:t}^{1,1}(j)$ and $\text{SB}_{r+1:t}^{2,j}$, are needed. \blacksquare

Equation-(5.64) (page 128)

$$\Lambda_{l;r}^{d,p} = \sum_{q \in \text{pa}(p)} \sum_{t \in \mathbf{L}_l^{d-1}} \alpha_{t:l}^{d-1,q}(p) \lambda_{t;r}^{d-1,q}(p) \text{ for } d > 1$$



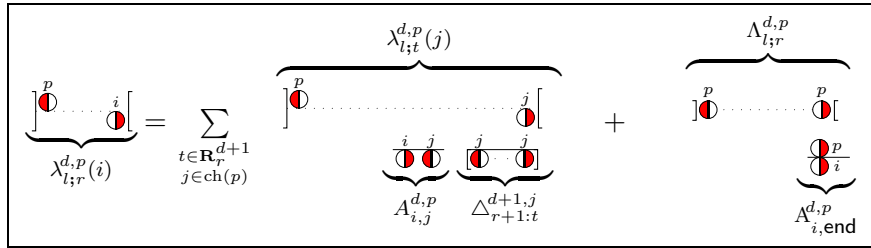
Proof. Writing down the definition in Equation-(5.49) and letting $W \triangleq \Pr(\cdot x_l^d = p)$ we have:

$$\begin{aligned}
\Lambda_{l;r}^{d,p} &\triangleq \Pr(\mathcal{O}_{l;r}^{\text{out}} \mid \cdot x_l^d = p, e_{l:r-1}^d = \mathbf{0}, e_r^d = 1) W \\
&= \sum_{q \in \text{pa}(p)} \sum_{t \in \mathbf{L}_l^{d-1}} \Pr(\mathcal{O}_{l;r}^{\text{out}}, \cdot x_t^{d-1} = q, e_{t:r-1}^{d-1} = \mathbf{0} \mid \cdot x_l^d = p, e_{l:r-1}^d = \mathbf{0}, e_r^d = 1) W \\
&\stackrel{(a)}{=} \sum_{q,t} \Pr(\mathcal{O}_{l;t}^{\text{out}} \mid \text{REST}) \underbrace{\Pr(\mathcal{O}_{t:l-1}^{\text{in}}, \cdot x_t^{d-1} = q, e_{t:l-1}^{d-1} = \mathbf{0})}_{\text{past}} \underbrace{\Pr(\cdot x_l^d = p)}_{\text{present}} \underbrace{\Pr(e_{l:r-1}^d = \mathbf{0}, e_r^d = 1)}_{\text{future}} \Pr(x_l^d = p) \\
&= \sum_{q,t} \Pr(\mathcal{O}_{t:l-1}^{\text{in}}, \cdot x_t^{d-1} = q, e_{t:l-1}^{d-1} = \mathbf{0}, \cdot x_l^d = p) \Pr(\mathcal{O}_{l;t}^{\text{out}} \mid \text{REST}) \\
&= \sum_{q,t} \underbrace{\Pr(\mathcal{O}_{t:l-1}^{\text{in}}, e_{t:l-1}^{d-1} = \mathbf{0}, \cdot x_l^d = p \mid \cdot x_t^{d-1} = q)}_{\alpha_{l;t}^{d,q}(p)} \underbrace{\Pr(x_t^{d-1} = q) \Pr(\mathcal{O}_{l;t}^{\text{out}} \mid \cdot x_t^{d-1} = q, e_{t:r-1}^{d-1} = \mathbf{0}, x_r^d = p)}_{\lambda_{l;r}^{d-1,q}(p)} \\
&= \sum_{q \in \text{pa}(p)} \sum_{t \in \mathbf{L}_l^{d-1}} \alpha_{l;t}^{d-1,q}(p) \lambda_{l;r}^{d-1,q}(p)
\end{aligned}$$

where in step (a) we have applied Lemma 5.1. ■

Equation-(5.65) (page 129)

$$\lambda_{l;r}^{d,p}(i) = \sum_{t \in \mathbf{R}_r^{d+1}} \sum_{j \in \text{ch}(p)} \lambda_{l;t}^{d,p}(j) A_{i,j}^{d,p} \Delta_{r+1:t}^{d+1,j} + \Lambda_{l;r}^{d,p} A_{i,\text{end}}^{d,p}$$



Proof. The proof consists of two parts corresponding to the left and right diagrams for $e_r^d = 0$ and $e_r^d = 1$ respectively when $\lambda_{l;r}^{d,p}(i)$ is re-written from Equation-(5.47c) as:

$$\begin{aligned}
\lambda_{l;r}^{d,p}(i) &\triangleq \Pr(\mathcal{O}_{l;r}^{\text{out}} \mid \cdot x_l^d = p, e_{l:r-1}^d = \mathbf{0}, x_r^{d+1} = i) \Pr(\cdot x_l^d = p) \\
&= \underbrace{\Pr(\mathcal{O}_{l;r}^{\text{out}} e_r^d = 0 \mid \cdot x_l^d = p, e_{l:r-1}^d = \mathbf{0}, x_r^{d+1} = i) \Pr(\cdot x_l^d = p)}_A \\
&\quad + \underbrace{\Pr(\mathcal{O}_{l;r}^{\text{out}} e_r^d = 1 \mid \cdot x_l^d = p, e_{l:r-1}^d = \mathbf{0}, x_r^{d+1} = i) \Pr(\cdot x_l^d = p)}_B
\end{aligned} \tag{B.1}$$

Following the factorisation suggested in left diagram we have:

$$\begin{aligned}
A &= \sum_{\substack{t \in \mathbf{R}_r^{d+1} \\ j \in \text{ch}(p)}} \Pr(\mathcal{O}_{l;r}^{\text{out}}, e_r^d = 0, \underbrace{\cdot x_{r+1}^{d+1} = j, \tau_{r+1}^{d+1} = t}_X \mid \underbrace{\cdot x_l^d = p, e_{l;r-1}^d = \mathbf{0}, x_r^{d+1} = i}_Y) \Pr(\cdot x_l^d = p) \\
&\stackrel{(a)}{=} \sum_{\substack{t \in \mathbf{R}_r^{d+1} \\ j \in \text{ch}(p)}} \underbrace{\Pr(\mathcal{O}_{l;t}^{\text{out}} \mid \text{REST}) \Pr(\cdot x_l^d = p) \Pr(\text{REST})}_{\lambda_{l;t}^{d,p}(j)} = \sum_{\substack{t \in \mathbf{R}_r^{d+1} \\ j \in \text{ch}(p)}} \lambda_{l;t}^{d,p}(j) \Pr(\text{REST}) \tag{B.2}
\end{aligned}$$

where $\text{REST} \triangleq \{\mathcal{O}_{r+1;t}^{\text{in}}, X, Y\}$, and again in step (a) we use Theorem 5.2 to recover the term $\lambda_{l;t}^{d,p}(j)$ and note that $\mathcal{O}_{l;r}^{\text{out}} = \{\mathcal{O}_{l;t}^{\text{out}} \cup \mathcal{O}_{r+1;t}^{\text{in}}\}$. Next, we factorise the term $\Pr(\text{REST})$ as follows:

$$\begin{aligned}
\Pr(\text{REST}) &= \Pr(\mathcal{O}_{r+1;t}^{\text{in}}, \tau_{r+1}^{d+1} = t \mid \cdot x_{r+1}^{d+1} = j, \text{REST} \setminus \tau) \Pr(x_{r+1}^{d+1} = j, e_r^d = 0 \mid x_{r+1}^{d+1} = i, x_r^d = p, \tau_r^d = t) \\
&= \Delta_{r+1;t}^{d+1,j} A_{i,j}^{d,p} \tag{B.3}
\end{aligned}$$

Substituting Equation-(B.3) into Equation-(B.2) we have:

$$A = \sum_{\substack{t \in \mathbf{R}_r^{d+1} \\ j \in \text{ch}(p)}} \lambda_{l;t}^{d,p}(j) \Delta_{r+1;t}^{d+1,j} A_{i,j}^{d,p} \tag{B.4}$$

Now let us turn to the case when $e_r^d = 1$ and compute B :

$$\begin{aligned}
B &= \Pr(\mathcal{O}_{l;r}^{\text{out}}, e_r^d = 1 \mid \cdot x_l^d = p, e_{l;r-1}^d = \mathbf{0}, x_r^{d+1} = i) \Pr(\cdot x_l^d = p) \\
&\stackrel{(a)}{=} \underbrace{\Pr(\mathcal{O}_{l;r}^{\text{out}} \mid \cdot x_l^d = p, e_{l;r-1}^d = \mathbf{0}, e_r^d = 1, \tau_r^{d+1} = i)}_{\Lambda_{l;r}^{d,p}} \Pr(\cdot x_l^d = p) \underbrace{\Pr(e_r^d = 1 \mid x_r^{d+1} = i, x_r^d = p)}_{A_{i,\text{end}}^{d,p}} \\
&= \Lambda_{l;r}^{d,p} A_{i,\text{end}}^{d,p} \tag{B.5}
\end{aligned}$$

where in step (a) we have used Lemma 5.2 to recover the term $\Lambda_{l;r}^{d,p}$. Substituting Equation-(B.4) and Equation-(B.5) into Equation-(B.1) the proof is obtained. \blacksquare

Equation-(5.81) (page 138)

The scaling factor φ_r at time r is calculated based on the partially scaled variables $\ddot{\alpha}$ and $\ddot{\phi}$ as follows:

$$\varphi_r = \sum_{i \in \mathcal{S}^2} \left(\ddot{\alpha}_{1;r}^{1,1}(i) + \ddot{\phi}_{1;r}^{1,1}(i) \right)$$

Proof. By definition:

$$\begin{aligned}
\sum_{i \in \mathcal{S}^2} \ddot{\alpha}_{1;r}^{1,1}(i) &= \sum_{i \in \mathcal{S}^2} \frac{\alpha_{1;r}^{1,1}(i)}{\prod_{t=1}^{r-1} \varphi_t} = \sum_{i \in \mathcal{S}^2} \frac{\Pr(y_{1:r}, x_r^2 = i, e_r^2 = 1, e_{1:r-1}^1 = \mathbf{0})}{\Pr(y_{1:r-1}, e_{1:r-2}^1 = \mathbf{0})} \\
&= \Pr(y_r, e_r^2 = 1, e_{r-1}^1 = 0 \mid y_{1:r-1}, e_{1:r-2}^1 = \mathbf{0})
\end{aligned}$$

$$\text{and similarly: } \sum_{i \in \mathcal{S}^2} \ddot{\phi}_{1;r}^{1,1}(i) = \Pr(y_r, e_r^2 = 0, e_{r-1}^1 = 0 \mid y_{1:r-1}, e_{1:r-2}^1 = \mathbf{0})$$

Therefore, $\sum_{i \in \mathcal{S}^2} \left(\ddot{\alpha}_{1:r}^{1,1}(i) + \ddot{\alpha}_{1:r}^{1,1}(i) \right) = \Pr(y_r, e_{r-1}^1 = 0 \mid y_{1:r-1}, e_{1:r-2}^1 = \mathbf{0}) \triangleq \varphi_r$ ■

Appendix C

Sufficient Statistics via Parameter Tying Transformation

In this appendix, we provide an alternative way to derive the sufficient statistics (SS) for the Hierarchical HMMS via a formal exploitation of ‘tying’ parameters in the exponential family. It is well-known that the DBN is a special case of BN with a ‘tied-parameter’, where the parameters get replicated over time. This phenomenon is also known as the ‘homogeneity’ property, ie: the parameter is invariant with time. Therefore, a DBN, in principle, can be viewed as a general BN, where the SS are derived for the BN and then a ‘tie’ operator is applied (eg: sum over time) to compute the SS for the DBN.

The end-state variables e_t^d in the DBN structure of the HHMM is, however, rather special. These variables are special because they cannot take on any arbitrary values and must satisfy the conditions in Equation-(5.6a) and Equation-(5.6b). A combination of, for example, $\{e_t^d = 1, e_t^{d+1} = 0\}$ is invalid. Furthermore, when they are observed certain (conditional) dependencies are simplified (as shown in Figure-(5-6(a)) and Figure-(5-7)), and therefore, the DBN network is simplified and collapsed into a simpler structure. Because of this fact, we call them the *context specific independent* (CSI) variables¹.

Let us first formally study the behaviour of the tying operator with respect to the SS for the exponential family.

¹DBN with CSI variables is a known issue and has been investigated in previous works (eg: Boutilier *et al.* (1996)).

C.0.1 Tying Transformation Theorem in the Exponential Family

Theorem C.1 (sufficient statistics in “tying” transformation) *Let λ be the full parameter of an exponential distribution, and γ be its compact parameter. Let Γ be the tie-transformation between γ and λ , and satisfy:*

$$\lambda = \Gamma(\gamma) = \mathbf{\Gamma} \cdot \gamma$$

then

$$T_\varphi = \mathbf{\Gamma}^\top \cdot T_\gamma$$

where $\mathbf{\Gamma}$, and $\mathbf{\Gamma}^\top$ are respectively the tie-transformation matrix and its transpose; T_λ and T_γ are the sufficient statistics for λ and γ respectively.

Proof. The proof for this theorem can be obtained easily by simple manipulation as follows. By definition of the exponential family (cf. Section 5.3.1), we have:

$$\Pr(x | \lambda) = h(x) \exp \{ \lambda^\top \cdot T(x) - A(\lambda) \}, \text{ where } T_\lambda = T(x)$$

Substituting λ by $(\mathbf{\Gamma} \cdot \gamma)$ we have:

$$\begin{aligned} \Pr(x | \gamma) &= h'(x) \exp \{ (\mathbf{\Gamma} \cdot \gamma)^\top \cdot T(x) - A'(\gamma) \} \\ &= h'(x) \exp \{ \gamma^\top \cdot (\mathbf{\Gamma}^\top \cdot T(x)) - A'(\gamma) \} \\ &= h'(x) \exp \{ \gamma^\top \cdot T'(x) - A'(\gamma) \} \end{aligned}$$

where $T_\gamma = T'(x) = \mathbf{\Gamma}^\top \cdot T(x)$, therefore: $T_\gamma = \mathbf{\Gamma}^\top \cdot T_\lambda$ ■

C.0.2 Tying Parameters in the DBN Structure of the HHMM

We know that the BN belongs to the family of exponential distributions (eg: see Dan (1998)). Using the tie-parameter ‘trick’ in Theorem C.1, we define $\theta_{\text{DBN}}, \theta_{\text{BN}}$ to be the parameters of the DBN (homogeneity exists) and of the DBN when viewed as a general BN (ie: homogeneity is lifted). The original parameters for the HHMM can then be viewed as the compact parameter being ‘tied’ from θ_{DBN} , and θ_{BN} in two steps:

$$\theta = \Gamma_{\text{CSI}}(\theta_{\text{DBN}}) = \mathbf{\Gamma}_{\text{CSI}} \cdot \theta_{\text{DBN}} \tag{C.1}$$

$$\theta_{\text{DBN}} = \Gamma_{\text{TIME}}(\theta_{\text{BN}}) = \mathbf{\Gamma}_{\text{TIME}} \cdot \theta_{\text{BN}} \tag{C.2}$$

where, roughly speaking, Γ_{CSI} compacts θ_{DBN} to θ via CSI transformation (ie: only valid for certain configurations of CSI variables), and Γ_{TIME} compacts θ_{BN} to θ_{DBN} via time (ie: parameters get replicated over time).

If we know the operations $\Gamma_{\text{CSI}}, \Gamma_{\text{TIME}}$ and the \mathbb{T} for θ_{BN} , then Theorem C.1 allows us to compute the \mathbb{T} for the HHMM parameter set θ as:

$$\mathbb{T} \langle \theta \rangle = \Gamma_{\text{CSI}}^{\mathbb{T}} \cdot \mathbb{T} \langle \theta_{\text{DBN}} \rangle \quad (\text{C.3})$$

$$\mathbb{T} \langle \theta_{\text{DBN}} \rangle = \Gamma_{\text{TIME}}^{\mathbb{T}} \cdot \mathbb{T} \langle \theta_{\text{BN}} \rangle \quad (\text{C.4})$$

C.0.2.1 The CSI-transformation

As mentioned, the Γ_{CSI} expands θ to θ_{DBN} by allowing only certain configurations of the CSI variables $e_{1:T}^{1:D}$. To be precise, these variables must satisfy conditions in Equation-(5.6a) and Equation-(5.6b). Let $\theta_{\text{DBN}} \triangleq \{A_{\text{DBN}}, A_{\text{[end]}}^{\text{DBN}}, B_{\text{DBN}}\}$, where these parameters are defined directly for each type of clique $\{z, \pi_z\}$ in the DBN network structure.

$$\begin{aligned} A_{\text{DBN}} &\triangleq \Pr(x_t^{d+1} \mid e_{t+1}^d, x_t^d, e_{t-1}^{d+1}, x_{t-1}^{d+1}) \\ A_{\text{[end]}}^{\text{DBN}} &\triangleq \Pr(e_t^d \mid x_t^d, x_t^{d+1}, e_t^{d+1}) \\ B_{\text{DBN}} &\triangleq \Pr(y_t \mid x_t^D) \end{aligned}$$

Note that the parameters on the LHS are independent of t . The Γ_{CSI} transforms each component of θ separately and does so only for valid configurations of $\{e_{1:T}^{1:D}\}$. For example:

$$\begin{aligned} B &\xrightarrow{\Gamma_{\text{CSI}}} B_{\text{DBN}} \\ A &\xrightarrow{\Gamma_{\text{CSI}}} \begin{cases} A_{\text{DBN}} & \text{for } e^{d:d+1} = 01 \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

Using Theorem C.1, we obtain the relation in $\mathbb{T} \langle \theta \rangle = \Gamma_{\text{CSI}}^{\mathbb{T}} \cdot \mathbb{T} \langle \theta_{\text{DBN}} \rangle$ as:

$$\begin{aligned} \mathbb{T} \langle B \rangle &= \mathbb{T} \langle B_{\text{DBN}} \rangle \\ \mathbb{T} \langle A \rangle &= \mathbb{T} \langle A_{\text{DBN}} \rangle \Big|_{\text{when } t > 1, e_t^{d:d+1} = 01} \\ \mathbb{T} \langle A_{\text{[end]}} \rangle &= \mathbb{T} \langle A_{\text{[end]}}^{\text{DBN}} \rangle \Big|_{\text{when } t < T, e_t^{d+1} = 1} \\ \mathbb{T} \langle \pi \rangle &= \mathbb{T} \langle A_{\text{DBN}} \rangle \Big|_{\text{when } t=1} + \sum_{x_{t-1}^{d+1}} \mathbb{T} \langle A_{\text{DBN}} \rangle \Big|_{\text{when } t > 1, e_t^{d:d+1} = 11} \end{aligned}$$

C.0.2.2 The Time-transformation

The Γ_{TIME} expands θ_{DBN} to θ_{BN} by repeating the same conditional probability over time for the same ‘type’ of clique. With the assumption that the root level is fixed, there are four main types of cliques identified from the DBN (cf. Figure-(5-8)) of the HHMM as shown in Figure-(C-1):

$$C_{(x_1^{d+1})} \triangleq \{x_1^{d+1}, x_1^d\} \quad \text{for } 1 \leq d \leq D \quad (\text{C.5a})$$

$$C_{(y_t)} \triangleq \{y_t, x_t^D\} \quad \text{for } 1 \leq t \leq T \quad (\text{C.5b})$$

$$C_{(x_t^{d+1})} \triangleq \{x_t^{d+1}, x_t^d, x_{t-1}^{d+1}, e_{t-1}^d, e_{t-1}^{d+1}\} \quad \text{for } 2 \leq t \leq T, 1 \leq d \leq D \quad (\text{C.5c})$$

$$C_{(e_t^{d+1})} \triangleq \{x_t^{d+1}, e_t^d, x_t^d, e_{t-1}^{d+1}\} \quad \text{for } 1 \leq t < T, 1 < d \leq D \quad (\text{C.5d})$$

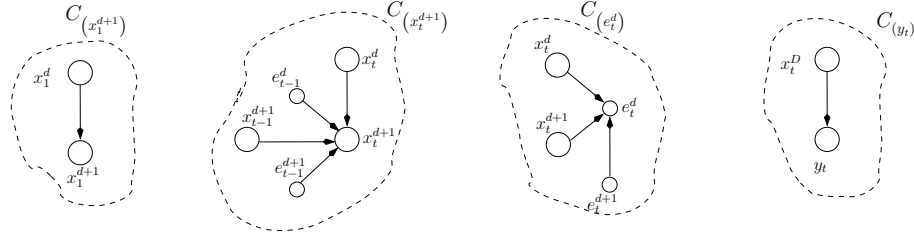


Figure C-1: Four main types of cliques identified in the DBN structure of the HHMM, which is essentially the same as Figure-(5-9).

The sufficient statistics for in the DBN are computed via Γ_{TIME}^T and given as:

$$\begin{aligned} \mathbf{T}_{\text{DBN}} \langle A \rangle |_{\text{for } t=1} &= \mathbf{T}_{\text{BN}} \langle C_{(x_1^{d+1})} \rangle & \mathbf{T}_{\text{DBN}} \langle A \rangle |_{\text{for } t>1} &= \sum_{t=2}^T \mathbf{T}_{\text{BN}} \langle C_{(x_t^{d+1})} \rangle \\ \mathbf{T}_{\text{DBN}} \langle A_{[\text{end}]} \rangle |_{\text{for } t>1} &= \sum_{t=1}^{T-1} \mathbf{T}_{\text{BN}} \langle C_{(e_t^{d+1})} \rangle & \mathbf{T}_{\text{DBN}} \langle B \rangle &= \sum_{t=1}^T \mathbf{T}_{\text{BN}} \langle C_{(y_t)} \rangle \end{aligned}$$

where we write $\mathbf{T}_W \langle \tau \rangle$ to refer to the sufficient statistics of τ discussed in the context of W .

C.0.3 Sufficient Statistics in the BN (θ_{BN})

Let us now focus on deriving the T for the BN form. When in this form, the BN is parameterised by a set of conditional probabilities over each clique: $\Pr(z | \pi_z)$ where z is

either x_t^d, e_t^d or y_t for $1 \leq t \leq T$ and $1 \leq d \leq D$. Computing T_{BN} with fully observed data is as simple as computing the marginal counts.

Assume that our observed data set $\mathcal{D} = \{\mathcal{D}^{(1)}, \mathcal{D}^{(2)}, \dots, \mathcal{D}^{(K)}\}$ is composed of K iid. sequences $\mathcal{D}^{(k)}$. For the k -th observation sequence, we denote $C_{(z)}^{(k)}$ to be the set of values in $\mathcal{D}^{(k)}$ observed for clique $C_{(z)}$. We introduce the indicator function $\mathbb{I}\left[\begin{smallmatrix} C_{(z)}^{(k)} \\ C_{(z)} \end{smallmatrix}\right]$ which returns 1 if $C_{(z)}^{(k)}$ is the same as $C_{(z)}$ and returns 0 otherwise²; and when the context is clear we use $\mathbb{I}_{C_{(z)}}^{(k)}$ in place of $\mathbb{I}\left[\begin{smallmatrix} C_{(z)}^{(k)} \\ C_{(z)} \end{smallmatrix}\right]$. In general, the indicator function of M arguments is defined as:

$$\mathbb{I}\left[\begin{smallmatrix} \{i_1, i_2, \dots, i_M\} \\ \{z_1, z_2, \dots, z_M\} \end{smallmatrix}\right] \triangleq \delta_{z_1}^{(i_1)} \times \delta_{z_2}^{(i_2)} \dots \times \delta_{z_M}^{(i_M)} = \prod_{k=1}^M \delta_{z_k}^{(i_k)}$$

Using these notations, the marginal counts for each clique becomes its SS and is computed as a counting process :

$$T_{\text{BN}} \langle C_{(z)} \rangle = \text{cnt} \left(C_{(z)}, \mathcal{D} \right) = \sum_{k=1}^K \mathbb{I}_{C_{(z)}}^{(k)} \quad \text{for } z \in \left\{ x_t^{d+1}, e_t^d, y_t \right\} \quad (\text{C.6a})$$

The sufficient statistics for θ_{DBN} is achieved via $\Gamma_{\text{TIME}}^{\text{T}}$ applied on T_{BN} as detailed in Section C.0.2.2. For example:

$$T_{\text{DBN}} \langle A \rangle |_{\text{for } t > 1} = \sum_{t=2}^T T_{\text{BN}} \langle C_{(x_t^d)} \rangle = \sum_{k=1}^K \sum_{t=2}^T \mathbb{I}_{C_{(x_t^d)}}^{(k)}$$

Finally, the sufficient statistics for the HHMM is computed via $\Gamma_{\text{CSI}}^{\text{T}}$ applied on T_{DBN} as detailed in Section C.0.2.1. For example, the sufficient statistics for the transition matrix A is:

$$T \langle A \rangle = \sum_{k=1}^K \sum_{t=2}^T \mathbb{I}_{C_{(x_t^d)}}^{(k)} \times \mathbb{I}\left[\begin{smallmatrix} 01 \\ e_{t-1}^{d:d+1} \end{smallmatrix}\right] \quad \text{and thus: } T \langle A_{i,j}^{d,p} \rangle = \sum_{k=1}^K \sum_{t=2}^T \delta_{x_t^d}^{(p)} \delta_{x_{t-1:t}^{d+1}}^{(i,j)} \delta_{e_{t-1}^{d:d+1}}^{(0,1)} \quad (\text{C.7a})$$

$$T \langle A_{[\text{end}]} \rangle = \sum_{k=1}^K \sum_{t=1}^{T-1} \mathbb{I}_{C_{(e_t^d)}}^{(k)} \times \mathbb{I}\left[\begin{smallmatrix} 11 \\ e_t^{d:d+1} \end{smallmatrix}\right] \quad \text{and thus: } T \langle A_{i,\text{end}}^{d,p} \rangle = \sum_{k=1}^K \sum_{t=1}^{T-1} \delta_{e_t^d}^{(1)} \delta_{x_t^{d:d+1}}^{(p,i)} \quad (\text{C.7b})$$

These sufficient statistics equations are identical to the ones derived in Equation-(5.24) and Equation-(5.25) obtained from the expression of the log-likelihood in Section 5.3.2.

²To be precise, we are comparing $C_{(z)}^{(k)}$ against a particular configuration of clique $C_{(z)}$ of interest.

Bibliography

- Adams, B. (2003a). *Mapping the Semantic Landscape of Film: Computational Extraction of Indices through Film Grammar*. Ph.D. thesis, Curtin University of Technology.
- Adams, B. (2003b). Where does computational media aesthetics fit? *IEEE Multimedia Magazine, Special Edition on Computational Media Aesthetics*, pages 18–27.
- Adams, B., Dorai, C., and Venkatesh, S. (2000). Role of shot length in characterizing tempo and dramatic story sections in motion pictures. In *IEEE Pacific Rim Conference on Multimedia 2000*, pages 54–57, Sydney, Australia.
- Adams, B., Dorai, C., and Venkatesh, S. (2001). Automated film rhythm extraction for scene analysis. In *IEEE International Conference on Multimedia and Expo*, pages 1056–1059, Tokyo, Japan.
- Adams, B., Dorai, C., and Venkatesh, S. (2002a). Finding the beat: An analysis of the rhythmic elements of motion pictures. In *Fifth Asian Conference on Computer Vision*, Melbourne, Australia.
- Adams, B., Dorai, C., and Venkatesh, S. (2002b). Towards automatic extraction of expressive elements from motion pictures: Tempo. *IEEE Transactions on Multimedia*, **4**(4), 472–481.
- Ahanger, G. and Little, T. D. C. (1996). A survey of technologies for parsing and indexing digital video. *Journal of Visual Communication and Image Representation*, **7**(1), 28–43.
- Aigrain, P., Jolly, P., and Longueville, V. (1998). Medium knowledge-based macro-segmentation of video into sequences. In M. Maybury, editor, *Intelligent Multimedia Information Retrieval*, pages 159–174. AAAI Press/MIT Press.
- Altunbasak, Y. (2000). A statistical approach to threshold selection in temporal video segmentation algorithms. In *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, volume 6, pages 2421–2424, Istanbul, Turkey.
- Antani, S., Kasturi, R., and Jain, R. (2002). A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video. *Pattern Recognition*, **35**, 945–965.

- Arijon, D. (1976). *Grammar of the Film Language*. Focal Press Limited.
- Ariki, Y., Shibutani, A., and Sugiyama, Y. (1997). Classification and retrieval of TV Sports News by DCT features. In *International Symposium on Information System and Technologies for Network Society*, pages 269–272.
- Barnard, M., Odobez, J.-M., and Bengio, S. (2003). Multi-modal audio-visual event recognition for football analysis. In *IEEE Workshop on Neural Networks for Signal Processing*, Toulouse.
- Bertini, M., Bimbo, A. D., and Pala, P. (2000). Content-based annotation and retrieval of news videos. In *International Conference on Multimedia and Expo*, pages 479–482.
- Bilmes, J. A. (1998). A gentle tutorial on the EM algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Technical report, University of Berkeley.
- Bojic, N. and Pang, K. K. (2000). Adaptive skin segmentation for head and shoulder video sequence. In *SPIE Visual Communication and Image Processing*, pages 704–711, Perth, Australia.
- Bonet, J. D. and Viola, P. (1998). Texture recognition using a non-parametric multi-scale statistical model. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 641–647, Santa Barbara, CA USA.
- Boreczky, J. and Wilcox, L. D. (1998). A Hidden Markov Model for video segmentation using audio and image features. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, volume 6, pages 3741–3744, Seattle, USA.
- Boutilier, C., Friedman, N., Goldszmidt, M., and Koller, D. (1996). Context-specific independence in Bayesian networks. In *Uncertainty in Artificial Intelligence*, pages 115–123, Portland, Oregon.
- Boykin, S. and Merlino, A. (2000). Machine learning of event segmentation for news on demand. *Communications of the ACM*, **43**(2), 35–41.
- Braddeley, W. H. (1975). *The Technique of Documentary Film Production*. Focal Press, London.
- Brants, T. (1999). Cascaded markov models. In *Proc. of 9th Conf. of the European Chapter of the Association for Computational Linguistics*, pages 118–125, Bergen, Norway.
- Bui, H., Venkatesh, S., and West, G. (2000). On the recognition of abstract Markov policies. In *Proceedings of the National Conference on Artificial Intelligence*, pages 524–530.

- Calic, J. and Izquierdo, E. (2002). Efficient keyframe extraction and video analysis. In *Int. Symposium on Information Technology*, pages 28–33, Las Vegas, USA.
- Chai, D. and Bouzerdoum, A. (2000). A bayesian approach to skin color classification in YCbCr color space. In *IEEE Region 10 Conference*, volume 2, pages 421–424, Kuala Lumpur, Malaysia.
- Chai, D. and Ngan, K. N. (1999). Face segmentation using skin color map in video phone applications. *IEEE Transactions on Circuits and Systems for Video Technology*, **9**(4), 551–564.
- Chen, L. and Ozsu, M. (2002). Rule-based scene extraction from video. In *Procs. of International Conference on Image Processing*, volume 2, pages II-737–II-740 vol.2. TY - CONF.
- Churchill, G. (1992). Hidden Markov chains and the analysis of genome structure. *Computers & Chemistry*, **16**(2), 107–115.
- Colombo, C., Bimbo, A. D., and Pala, P. (1999). Semantics in visual information retrieval. *IEEE Multimedia*, **6**(3), 38–53.
- Dan, G. (1998). Graphical models and exponential families. In *Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence*, pages 156–165, San Francisco, CA. Morgan Kaufmann Publishers.
- Davis, D. (1969). *The Grammar of Television Production*. Redwood Press Limited, Trowbridge & London.
- Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via EM algorithm (with discussion). *Journal of the Royal Statistical Society*, **39**(1), 1–38.
- Deriche, R. (1992). Recursively implementing the gaussian and its derivatives. In *International Conf. on Image Processing*, pages 263–267, Singapore.
- Dimitrova, N., Zhang, J.-J., Shahraray, B., Sezan, I., Huang, T., and Zakhor, A. (2002). Applications of video-content analysis and retrieval. *IEEE Multimedia*, **9**(3), 42–55.
- Divakaran, A., Radhakrishnan, R., and Peker, K. (2002). Motion activity-based extraction of keyframes from video shots. In *IEEE Int. Conf. on Image Processing*, pages 932–935, Rochester, NY.
- Dorai, C. and Venkatesh, S. (2001a). Bridging the semantic gap in content management systems: Computational media aesthetics. *International Conference on Computational Semiotics in Games and New Media*, pages 94–99.

- Dorai, C. and Venkatesh, S. (2001b). Computational media aesthetics: Finding meaning beautiful. *IEEE Multimedia*, **8**(4), 10–12.
- Dorai, C. and Venkatesh, S., editors (2002). *Media Computing: Computational Media Aesthetics*. The Kluwer International Series in Video Computing. Kluwer Academic Publishers.
- Dorai, C., Aradhye, H., and Shim, J.-C. (2001). End-to-end videotext recognition for multimedia content analysis. In *IEEE International Conference on Multimedia and Expo*, pages 479–482.
- Dorai, C., Mauthe, A., Nack, F., Rutledge, L., Sikora, T., and Zettl, H. (2002). Media semantics: Who needs it and why? In *ACM Int. Conf. on Multimedia*, pages 580–853, Juan-les-Pins, France.
- Doulamis, N., Doulamis, A., Avrithis, Y., and Killias, S. (1999). A stochastic framework for optimal keyframe extraction from MPEG video databases. In *The Third Workshop on Multimedia Signal Processing*, pages 141–146.
- Drew, M. S. and Au, J. (2000). Video keyframe production by efficient clustering of compressed chromaticity signatures. In *Procs. of ACM Multimedia*, pages 365–367, Los Angeles, CA.
- Feraud, R., Bernier, O. J., Viallet, J.-E., and Collobert, M. (2001). A fast and accurate face detector based on neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **23**(1), 42–53.
- Ferman, A. M. and Tekalp, A. M. (1997). Multiscale content extraction and representation for video indexing. In *Multimedia Storage and Archival Systems II*, volume SPIE-3229, pages 23–31, Dalls.
- Ferman, A. M. and Tekalp, A. M. (1999). Probabilistic analysis and extraction of video content. In *International Conference on Image Processing*, volume 2, pages 91–95.
- Fine, S., Singer, Y., and Tishby, N. (1998). The hierarchical hidden markov model: Analysis and applications. *Machine Learning*, **32**(1), 41–62.
- Fischer, S., Lienhart, R., and Effelsberg, W. (1995). Automatic recognition of film genres. In *The Third ACM International Multimedia Conference and Exhibition*, pages 367–368, New York. ACM Press.
- Foss, B. (1992). *The Filmmaking: Narrative and Structural Techniques*. Silman-James Press, Los Angeles.
- Garcia, C. and Tziritas, G. (1999). Face detection using quantized skin color regions merging and wavelet packet analysis. *IEEE Transactions on Multimedia*, **1**(3), 264–277.

- Gibert, X., Li, H., and Doermann, D. (2003). Sports video classification using HMMs. In *IEEE Int. Conf. on Multimedia and Expo*, volume 2, pages 345–348. TY - CONF.
- Govindaraju, V. (1996). Locating human faces in photographs. *International Journal of Computer Vision*, **19**(2), 129–146.
- Gu, L., Tsui, K., and Keightley, D. (1997). Dissolve detection in MPEG compressed video. In *Proc. of IEEE International Conference on Intelligent Processing Systems*, volume 2, pages 1692–1696.
- Gunsel, B. and Tekalp, A. M. (1998). Content-based video abstraction. In *IEEE Int. Conf. on Image Processing*, volume 3, pages 128–132, Chicago, Illinois, USA.
- Hanjalic, A. (2002). Shot-boundary detection: Unraveled and resolved? *IEEE Transaction in Circuits and Systems for Video Technology*, **12**(2), 90–105.
- Hanjalic, A. and Zhang, H. (1999a). An integrated scheme for automated video abstraction. *IEEE Transaction in Circuits and Systems for Video Technology*, **9**(8), 1280–1289.
- Hanjalic, A. and Zhang, H. (1999b). Optimal shot boundary detection based on robust statistical models. In *IEEE Int. Conf. on Multimedia Computing and Systems*, volume 2, pages 710–714.
- Hanjalic, A., Lagendijk, R. L., and Biemond, J. (1998). A new method for keyframe based video content representation. In A. Smeulders and R. Jain, editors, *Image Databases and Multimedia Search*, page Ch. 5. World Scientific Singapore.
- Hanjalic, A., Lagendijk, R. L., and Biemond, J. (1999a). Automated high-level movie segmentation for advanced video retrieval systems. *IEEE Transactions in Circuits and Systems for Video Technology*, **9**(4), 580–588.
- Hanjalic, A., Biemond, J., and Lagendijk, R. (1999b). Automatically segmenting movies into logical units. In *Proc. of the Third Int. Conf. on Visual Information and Information Systems*, pages 229–236, Amsterdam.
- Heckerman, D., Jordan, M. I., and Smyth, P. (1997). Probabilistic independence networks for hidden markov probability models. *Neural Computation*, **9**(2), 227–269.
- Herman, L. (1965). *Educational Films: Writing, Directing, and Producing for Classroom, Television, and Industry*. Crown Publishers, New York.
- Hjelmas, E. and Low, B. K. (2001). Face detection: a survey. *Computer Vision and Image Understanding*, **83**(3), 236–274.
- Hsu, R., Abdel-Mottaleb, M., and Jain, A. (2001). Face detection in color images. In *IEEE International Conference on Image Processing*, volume 1, pages 1046–1049, Thessaloniki, Greece.

- Huang, J., Liu, Z., and Wang, Y. (1998). Integration of audio and visual information for content-based video segmentation. In *Int. Conf. on Image Processing*, volume 3, pages 526–529.
- Huang, J., Liu, Z., and Wang, Y. (1999a). Integration of multimodal features for video scene classification based on HMM. In *IEEE Signal Processing Society: Workshop on Multimedia Signal Processing*, pages 53–58, Copenhagen, Denmark.
- Huang, J., Liu, Z., and Wang, Y. (2000). Joint video scene segmentation and classification based on hidden Markov model. In *IEEE Int. Conf. on Multimedia and Expo*, pages 1551–1554, New York, USA.
- Huang, Q., Liu, Z., Rosenberg, A., Gibbon, D., and Shahraray, B. (1999b). Automated generation of news content hierarchy by integrating audio, text, and video information. In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 6, pages 3025–3028.
- Ide, I., Yamamoto, K., and Tanaka, H. (1998). Automatic video indexing based on shot classification. In *The First Int. Conf. on Advanced Multimedia Content Processing*, pages 99–114, Osaka, Japan.
- Isenhour, J. P. (1975). The effects of context and order in film editing. *AV Communication Review*, **23**(1), 69–79.
- Iurgel, U., Meermeier, R., Eickeler, S., and Rigoll, G. (2001). New approaches to audio-visual segmentation of TV news for automatic topic retrieval. In *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, volume 3, pages 1397–1400, Salt Lake City, Utah.
- Iyengar, G. and Lippman, A. (1998). Models for automatic classification of video sequences. In *Storage and Retrieval for Image and Video Databases*, pages 216–227.
- Jain, A. K., Vailaya, A., and Xiong, W. (1999). Query by video clip. *Multimedia Systems*, **7**(5), 368–384.
- Jain, A. K., Duin, R. P. W., and Mao, J. (2000). Statistical pattern recognition: A review. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, **22**(1), 4–37.
- Jain, R. (2001). Structuralizing multimedia data. *IEEE Multimedia*, pages 1–2.
- Jain, R. and Hampapur, A. (1994). Metadata in video databases. *ACM Special Interest Group on Management of Data*, **23**(4), 27–33.
- Jasinschi, R. S., Dimitrova, N., McGee, T., Agnihotri, L., Zimmerman, J., and Li, D. (2001). Integrated multimedia processing for topic segmentation and classification. In *Proceedings of IEEE International Conference on Image Processing*, volume 3, pages 366–369, Thessaloniki, Greece.

- Jensen, F. V. (1996). *An Introduction to Bayesian Networks*. Springer Verlag.
- Jordan, M. I. (2004). *Introduction to Graphical Models*. MIT Press, Cambridge, MA. Forthcoming.
- Jung, K., Kim, K. I., and Jain, A. K. (2004). Text information extraction in images and video: a survey. *Pattern Recognition*, **37**, 977–997.
- Kang, H.-B. (2003). Affective content detection using HMMs. In *Proc. of 11th ACM International Conference on Multimedia*, pages 259–262, Berkeley, USA.
- Kankanhalli, M. S. and Chua, T.-S. (2000). Video modeling using strata-based annotation. *IEEE Multimedia*, **7**(1), 68–74.
- Kijak, E., Oisel, L., and Gros, P. (2003a). Hierarchical structure analysis of sport videos using HMMs. In *Int. Conf. on Image Processing*, volume 2, pages II–1025–8 vol.3. TY - CONF.
- Kijak, E., Gravier, G., Gros, P., Oisel, L., and Bimbot, F. (2003b). HMM based structuring of tennis videos using visual and audio cues. In *IEEE Int. Conf. on Multimedia and Expo*, volume 3, pages III–309–12, Baltimore, USA.
- Kim, C. and Hwang, J. (2000). An integrated scheme for object-based video abstraction. In *Procs. of ACM Multimedia*, pages 303–311, Los Angeles, CA.
- Kotropoulos, C. and Pitas, L. (1997). Rule-based face detection in frontal views. *Proceedings of the 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, **4**, 25–37.
- Kriby, M. and Sirovich, L. (1990). Application of the karhunen-loeve procedure for the characterization of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **12**(1), 103–108.
- Lari, K. and Young, S. J. (1990). The estimation of stochastic context-free grammars using the inside-outside algorithm. In *Computer Speech and Language*, pages 35–56.
- Lee, J. C.-M., Li, Q., and Xiong, W. (1992). VIMS: A video information management system. *Multimedia Tools and Applications*, (9), 1–25.
- Li, Y., Wan, X., and Kuo, C.-C. J. (2001). Introduction to content-based image retrieval: Overview of key techniques. In B. a. Castelli, editor, *Image Databases: Search and Retrieval of Digital Imagery*, pages 261–284. John Wiley & Sons, Inc.
- Lienhart, R. (2001). Reliable transition detection in videos: A survey and practitioner’s guide. *International Journal of Image and Graphics*, **1**(3), 469–486.

- Lin, T. and Zhang, H. J. (2000). Automatic video scene extraction by shot grouping. *Pattern Recognition*, **4**, 39–42.
- Liu, X. and Chen, T. (2002). Shot boundary detection using temporal statistics modeling. In *Proceedings of Acoustics, Speech, and Signal Processing*, volume 4, pages 3389–3392.
- Liu, X., Zhuang, Y., and Pan, Y. (1999). A new approach to retrieve video by example video clip. In *Proceeding of The 7th ACM International Multimedia Conference*, Orlando, Florida, USA.
- Liu, Z. and Huang, Q. (1999). Detecting news reporting using audio/visual information. In *International Conference on Image Processing*, pages 24–28, Kobe, Japan.
- Liu, Z., Huang, J., Wang, Y., and Chen, T. (1997). Audio feature extraction and analysis for scene classification. In *IEEE First Workshop on Multimedia Signal Processing*, pages 343–348, Princeton, NJ, USA.
- Liu, Z., Huang, J., and Wang, Y. (1998). Classification of TV programs based on audio information using hidden markov model. In *IEEE Signal Processing Society 1998 Workshop on Multimedia Signal Processing*, pages 27–32.
- Lu, C., Drew, M. S., and Au, J. (2003). An automatic video classification system based on a combination of HMM and video summarization. *International Journal of Smart Engineering and System Design*, **5**(1), 33–45.
- Luhr, S., Bui, H. H., Venkatesh, S., and West, G. (2003). Recognition of human activity through hierarchical stochastic learning. In *Int. Conf. on Pervasive Computing and Communication*, pages 416–422.
- Ma, Y.-F., Lu, L., Zhang, H.-J., and Li, M. (2002). A user attention model for video summarization. In *Proc. of ACM Multimedia*, Juan Les Pin, France.
- Mahmood, T. S. and Srinivasan, S. (2000). Detecting topical events in digital video. In *ACM Multimedia*, pages 85–94.
- Mediaware-Company (1999). Mediaware solution webflix professional V1.5.3. <http://www.mediaware.com.au/webflix.html>.
- Menser, B. and Wien, M. (2000). Segmentation and tracking of facial regions in color image sequences. In *SPIE Visual Communication and Image Processing*, volume 4067, pages 731–740, Perth, Australia.
- Merlino, A., Morey, D., and Maybury, M. T. (1997). Broadcast news navigation using story segmentation. *ACM Multimedia*, pages 381–391.

- Moghaddam, B. and Pentland, A. (1997). Probabilistic visual learning for object representation. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, **19**(7), 696–710.
- Monaco, J. (1977). *How to Read a Film: the Art, Technology, Language, History and Theory of Film and Media*. Oxford University Press.
- Moncrieff, S. (2004). *Computational Approaches for the Extraction of Semantics Through Film Audio*. Ph.D. thesis, Curtin University of Technology.
- Moncrieff, S., Dorai, C., and Venkatesh, S. (2001a). Affect computing in films through sound energy dynamics. In *The 9th ACM International Conference on Multimedia*, pages 525–527, Ottawa, Canada.
- Moncrieff, S., Dorai, C., and Venkatesh, S. (2001b). Detecting indexical signs in film audio for scene interpretation. In *IEEE International Conference on Multimedia and Expo*, pages 989–992, Tokyo, Japan.
- Murphy, K. (2001). Representing and learning hierarchical structure in sequential data. Unpublished manuscript, Available at: <http://www-anw.cs.umass.edu/~cs691t/SS02/readings.html>.
- Murphy, K. and Nefian, A. (2001). Embedded graphical models. Unpublished manuscript, Available at: citeseer.ist.psu.edu/murphy01embedded.html.
- Murphy, K. and Paskin, M. (2001). Linear-time inference in hierarchical HMMs. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, Cambridge, MA. MIT Press.
- Naphade, M. and Huang, T. (2000a). Inferring semantic concepts for video indexing and retrieval. *IEEE International Conference on Image Processing*, **3**, 766 – 769.
- Naphade, M. and Huang, T. (2000b). A probabilistic framework for semantic indexing and retrieval in video. In *IEEE International Conference on Multimedia and Expo*, volume 1, pages 475 – 478, New York.
- Naphade, M., Kristjansson, T., Frey, B., and Huang, T. (1998). Probabilistic multimedia objects (multijects): a novel approach to video indexing and retrieval in multimedia systems. In *Proc. IEEE International Conference on Image Processing*, volume 3, pages 536–540, Chicago, IL.
- Naphade, M. R. and Huang, T. S. (2002). Discovering recurrent events in video using unsupervised methods. In *Int. Conf. on Image Processing*, volume 2, pages 13–16, Rochester, NY, USA.

- Naphade, M. R., Garg, A., and Huang, T. S. (2001). Duration dependent input output markov models for audio-visual event detection. *IEEE International Conference on Multimedia and Expo*, pages 253 – 256.
- Ngo, C.-W., Zhang, H.-J., and Pong, T.-C. (2001). Recent advances in content based video analysis. *International Journal of Image and Graphics*, **1**(3), 445–468.
- Nitta, N., Babaguchi, N., and Kitahashi, T. (2002). Story based representation for broadcasted sports video and automatic story segmentation. In *Proceedings of IEEE International Conference on Multimedia and Expo*, volume 1, pages 813–816.
- Nowak, R. and Scoot, C. (2003). The fisher-neyman factorization theorem.
- Osuna, E., Freund, R., and Girosi, F. (1997). Training support vector machines: an application to face detection. *International Conference on Computer Vision and Pattern Recognition*, pages 130–136.
- Pala, P. and Santini, S. (1999). Image retrieval by shape and texture. *Pattern Recognition*, **32**(3), 517–527.
- Park, H. and Lee, S. (1996). Off-line recognition of large-set handwritten characters with multiple hidden Markov models. *Pattern Recognition*, **29**(6), 231–244.
- Patel, N. V. and Sethi, I. K. (1996). Compressed video processing for cut detection. In *IEE Proceedings: Vision, Image and Signal Processing*, pages 315–322, 134.
- Pearl, J. (1998). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. The Morgan Kaufmann Series in Representation and Reasoning. Morgan Kaufmann Publishers, Inc., San Francisco, 2nd edition.
- Pearl, J. (2004). *Causality : Models, Reasoning, and Inference*. Cambridge University Press.
- Penn, R. (1971). Effects of motion and cutting-rate in motion pictures. *AV Communication Review*, **19**(1), 29–49.
- Pfeiffer, S., Lienhart, R., Fischer, S., and Effelsberg, W. (1996). Abstracting digital video automatically. *Journal of Visual Communication and Image Representation*, **7**(4), 345–353.
- Pham, T. V., Worring, M., and Smeulders, A. W. M. (2001). Face detection by aggregated bayesian network classifiers. In *Machine Learning and Data Mining in Pattern Recognition*, volume 23, pages 249–262.
- Phung, D. Q. (2001). *An Investigation into Audio for Content Annotation*. honours, Honours thesis, Department of Computing, Curtin University of Technology.

- Phung, D. Q., Dorai, C., and Venkatesh, S. (2002). Video genre categorization using audio wavelet coefficients. In *The Fifth Asian Conference on Computer Vision*, pages 69–74, Melbourne, Australia.
- Prescher, D. (2003). A short tutorial on the expectation-maximization algorithm.
- Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, California, USA.
- Rabiger, M. (1998). *Directing the Documentary*. Focal Press, Boston, 3rd edition.
- Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. In *Procs. IEEE*, volume 77, pages 257–286.
- Radev, I., Pissinou, N., and Makki, K. (1999). Film video modeling. In *Workshop on Knowledge and Data Engineering Exchange*, pages 122–128, Chicago, USA.
- Rasheed, Z., Sheikh, Y., and Shah, M. (2003a). On the use of computable features for film classification. *IEEE Transactions on Circuits and Systems for Video Technology*, **1**(11), 1–11.
- Rasheed, Z., Sheikh, Y., and Shah, M. (2003b). Semantic film preview classification using low-level computable features. In *Int. Workshop on Multimedia Data and Document Engineering*, Berlin, Germany.
- Rowley, H. A., Baluja, S., and Kanade, T. (1998). Neutral network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**(1), 23–38.
- Rui, R., Huang, T. S., and Mehrotra, S. (1998). Exploring video structure beyond the shots. In *Proceedings of IEEE International Conference on Multimedia Computing and Systems*, pages 237–240.
- Rui, Y., Huang, T. S., and Chang, S.-F. (1999). Image retrieval: Current technique, promising directions, and open issues. *Journal of Visual Communication and Image Representation*, **10**, 39–62.
- Sanchez, J. M., Binefa, X., and Kender, J. (2002). Coupled Markov Chains for video contents characterization. In *IEEE Int. Conf. on Pattern Recognition*, volume 2, pages 461–464, Quebec, Canada.
- Schlick, C. (2000). Fine’s algorithm for hierarchical HMMs. Technical report, Research Establishment for Applied Science (FGAN).
- Shearer, K., Dorai, C., and Venkatesh, S. (2000). Incorporating domain knowledge with video and voice data analysis. In *Workshop on Multimedia Data Mining*, Boston, USA.

- Shim, J.-C., Dorai, C., and Bolle, R. (1998). Automatic text extraction from video for content-based annotation and retrieval. In *International Conference on Pattern Recognition*, volume 1, pages 618–620, Brisbane, Australia.
- Sirohey, S. A. (1993). Human face segmentation and identification. Technical Report CS-TR-3176, University of Maryland.
- Skounakis, M., Craven, M., and Ray, S. (2003). Hierarchical hidden markov models for information extraction. In *Procs. of the Eighteen International Joint Conference on Artificial Intelligence*, Acapulco, Mexico.
- Smeulders, A., Worring, M., Santini, S., and Gupta, A. (2000). Content-based image retrieval at the end of the early years. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, **22**(12), 1349–1380.
- Smoliar, S. and Zhang, H. (1994). Content-based video indexing and retrieval. *IEEE Multimedia Magazine*, **1**(2), 62–72.
- Smyth, P., Heckerman, D., and Jordan, M. (1997). Probabilistic independence networks for hidden markov probability models. *Neural Computation*, **9**(2), 227–269.
- Snoek, C. G. and Worring, M. (2000). Multimodal video indexing: A review of the state-of-the-art. *Multimedia Tools and Applications*. In Press.
- Snoek, C. G. M., Worring, M., and Hauptmann, A. G. (2004). Detection of TV news monologues by style analysis. In *IEEE International Conference on Multimedia and Expo*, Taipei, Taiwan.
- Sobchack, T. and Sobchack, V. (1987). *An introduction to Film*. Scott, Foresman and Company.
- Srinivasan, M. V., Venkatesh, S., and Hoise, R. (1997). Qualitative estimation of camera motion parameters from video sequences. *Pattern Recognition*, **30**(4), 593–606.
- Sundaram, H. (2002). *Segmentation, Structure Detection and Summarization of Multimedia Sequences*. Ph.D. thesis, Columbia University.
- Sundaram, H. and Chang, S. (2000a). Determining computable scenes in films and their structure using audio-visual memory models. In *ACM Int. Conf. on Multimedia*, pages 94–104.
- Sundaram, H. and Chang, S.-F. (2000b). Video scene segmentation using video and audio features. In *International Conference on Multimedia and Expo*, volume 2, pages 1145–1148, New York, USA.
- Sundaram, H. and Chang, S.-F. (2002). Computable scenes and structures in films. *IEEE Transactions in Multimedia*, **4**(4), 482–491.

- Sung, K. K. and Poggio, T. (1998). Example-based learning for view-based human face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**(1), 39–51.
- Tan, Y. P., Saur, D. D., Kulkarni, S. R., and Ramadge, P. J. (1999). A framework for measuring video similarity and its application to video query by example. In *Int. Conf. on Image Processing*, pages 106–110, Japan.
- Taniguchi, Y. (1995). An intuitive and efficient access interface to real-time incoming video based on automatic indexing. In *Procs. of ACM Multimedia*, pages 25–33, San Francisco, CA.
- Taskiran, C., Pollak, I., Bouman, C. A., and Delp, E. J. (2003). Stochastic models of video structure for program genre detection. In *SPIE Conf. on Visual Content Processing Representation*, volume 4315, pages 571–578, Madrid, Spain.
- Theocharous, G. and Mahadevan, S. (2002). Learning the hierarchical structure of spatial environments using multiresolution statistical models. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 1038–1043.
- Truong, B. T. (2004). *An Investigation into Structural and Expressive Elements in Film*. Ph.D. thesis, Curtin University of Technology.
- Truong, B. T., Dorai, C., and Venkatesh, S. (2000a). Automatic genre identification for content-based video categorization. In *Proceedings of the 15th International Conference on Pattern Recognition*, volume II, pages 230–233, Barcelona, Spain.
- Truong, B. T., Dorai, C., and Venkatesh, S. (2000b). Improved fade and dissolve detection for reliable video segmentation. In *Proceedings of the International Conference on Image Processing*, volume III, pages 961–964, Vancouver, Canada.
- Truong, B. T., Dorai, C., and Venkatesh, S. (2000c). New enhancements to cut, fade, and dissolve detection processes in video segmentation. In *Proceedings of the 8th ACM International Conference on Multimedia*, pages 219–227, Los Angeles, California.
- Truong, B. T., Venkatesh, S., and Dorai, C. (2002a). Application of computational media aesthetics methodology to extracting color semantics in film. In *ACM International Conference on Multimedia*, pages 339–342, Juan Les Pins, France.
- Truong, B. T., Dorai, C., and Venkatesh, S. (2002b). Automatic scene extraction in motion pictures. *IEEE Transactions in Circuits and Systems for Video Technology*, **13**(1), 5–10.
- Turk, M. A. and Pentland, A. P. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, **3**(1), 71–96.

- Vasconcelos, N. and Lippman, A. (1998a). A bayesian framework for content-based indexing and retrieval. In *Proceedings Data Compression Conference*, page 580, Snowbird, UT , USA. TY - CONF.
- Vasconcelos, N. and Lippman, A. (1998b). A bayesian framework for semantic content characterization. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 566–571, Santa Barbara.
- Vasconcelos, N. and Lippman, A. (1998c). Bayesian modeling of video editing and structure: Semantic features for video summarization and browsing. In *International Conference on Image Processing*, pages 153–157, Chicago, USA.
- Vasconcelos, N. and Lippman, A. (2000a). A probabilistic architecture for content-based image retrieval. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, **1**, 216–221.
- Vasconcelos, N. and Lippman, A. (2000b). Statistical models of video structure for content analysis and characterization. *IEEE Transaction on Image Processing*, **9**(1), 3–19.
- Vendrig, J. and Worring, M. (2002). Systematic evaluation of logical story unit segmentation. *IEEE Transactions on Multimedia*, **4**(4), 492–499.
- Walls, F., Jin, H., Sista, S., and Schwartz, R. (1999). Topic detection in broadcast news. In *Proceedings of the DARPR Broadcast News Workshop*, pages 193–198.
- Wang, C., Wang, Y., Liu, H., and He, Y. (2003). Automatic story segmentation of news video based on audio-visual features and text information. In *Int. Conf. on Machine Learning and Cybernetics*, volume 5, pages 3008–3011.
- Wang, J., Chua, T.-S., and Chen, L. (2001). Cinematic-based model for scene boundary detection. In *The Eight Conference on Multimedia Modeling*, Amsterdam, Netherland.
- Wei, G., Agnihotri, L., and Dimitrova, N. (2000). TV program classification based on face and text processing. In *IEEE Int. Conf. on Multimedia and Expo*, volume 3, pages 1345–1348, New York.
- Wolf, W. (1996). Keyframe selection by motion analysis. In *Procs. of the Int. Conf. on Acoustics, Speech, and Signal Processing*, volume 2, pages 1228–1231.
- Wolf, W. (1997). Hidden Markov models parsing of video programs. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, volume 4, pages 2609–2611.
- Wu, J. K. and Kankanhalli, M. S. (2000). *Perspectives on Content-Based Multimedia Systems*. The Kluwer International Series on Information Retrieval. Kluwer Academic Publishers.

- Xie, L. and Chang, S.-F. (2003). Unsupervised mining of statistical temporal structures in video. In A. Rosenfeld, D. Doreman, and D. Dementhons, editors, *Video Mining*. Kluwer Academic Publishers.
- Xie, L., Chang, S.-F., Divakaran, A., and Sun, H. (2002a). Learning hierarchical hidden markov models for unsupervised structure discovery from video. Technical report, Columbia University.
- Xie, L., Chang, S.-F., Divakaran, A., and Sun, H. (2002b). Structure analysis of soccer video with hidden Markov models. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, volume 4, pages IV-4096 – IV-4099, Orlando, Finland.
- Xie, L., Chang, S.-F., Divakaran, A., and Sun, H. (2004). Structure analysis of soccer video with domain knowledge hidden Markov models. *Pattern Recognition Letters*, (25), 767–775.
- Xiong, W., Lee, J., and Ma, R. (1997). Automatic video data structuring through shot partitioning and keyframe computing. *Machine Vision and Applications*, **10**(2), 51–65.
- Xu, G. and Sugimoto, T. (1998). Rits eye: A software-based system for realtime face detection and tracking using pan-tilt-zoom controllable camera. *International Conference on Pattern Recognition*, **2**, 1194–1197.
- Yamato, J., Ohya, J., and Ishii, K. (1992). Recognizing human action in time-sequential images using hidden markov model. *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pages 379–385.
- Yang, J. and Waibel, A. (1996). A real-time face tracker. In *IEEE Workshop on Applications of Computer Vision*, pages 142–147, Sarasota, Florida, USA.
- Yang, J., Stiefelhagen, R., Meier, U., and Waibel, A. (1998). Real-time face and facial feature tracking and applications. *Auditory-Visual Speech Processing*, pages 79–84.
- Yang, M.-H., Kriegman, D. J., and Ahuja, N. (2002). Detecting faces in images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**(1), 34–58.
- Yeu, B. L. and Liu, B. (1995). Rapid scene analysis on compressed video. *IEEE Transaction in Circuits and Systems for Video Technology*, **2**, 533–544.
- Yeung, M. M. and Liu, B. (1995). Efficient matching and clustering of shots. In *Proceedings of IEEE Conference on Image Processing*, volume 2, pages 338–341.
- Yeung, M. M. and Yeo, B. L. (1996). Time-constrained clustering for segmentation of video into story units. In *International Conference on Pattern Recognition*, pages 375–380.
- Yeung, M. M., Yeo, B. L., and Liu, B. (1996). Extracting story units from long programs for video browsing and navigation. In *IEEE Proceedings of Multimedia*, pages 296–305.

- Yong, Y. (1999). An accurate and robust method for detecting video shot boundaries. In *IEEE Int. Conf. on Multimedia Computing and Systems*, volume 1, pages 850–854.
- Yoshitaka, A., Ishii, T., Hirakawa, M., and Ichikawa, T. (1997). Content-based retrieval of video data by the grammar of film. In *Procs. of Symposium on Visual Languages*, pages 310–317, Isle of Capri, Italy.
- Yu, X.-D., Wang, L., Tian, Q., and Xue, P. (2004). Multi-level video representation with application to keyframe extraction. In *Int. Conf. on Multimedia Modeling*, pages 117–121, Brisbane, Australia.
- Yuan, J., Tian, Q., and Ranganath, S. (2004). Fast and robust search method for short video clips from large video collection. In *Int. Conf. on Pattern Recognition*, pages 866–869.
- Zettl, H. (1999). *Sight, Sound, Motion: Applied Media Aesthetics*. Wadsworth Publishing Company.
- Zhai, Y., Rasheed, Z., and Shah, M. (2004). A framework for semantic classification of scenes using finite state machines. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 279–288.
- Zhang, H., Wu, J., Zhong, D., and Smoliar, S. (1997). An integrated system for content-based video retrieval and browsing. *Pattern Recognition*, **30**(4), 643–658.
- Zhang, H. J., Kankanhalli, A., and Smoliar, S. W. (1993). Automatic partitioning of full motion video. *Multimedia Systems*, **1**(1), 10–28.
- Zhu, X., Wu, L., Xue, X., Lu, X., and Fan, J. (2001). Automatic scene detection in news program by integrating visual feature and rules. In *IEEE Pacific-Rim Conference on Multimedia*, pages 837–842, Beijing, China.
- Zhuang, Y., Rui, Y., Huang, T., and Mehrotra, S. (1998). Adaptive key frame extraction using unsupervised clustering. In *Int. Conf. on Image Processing*, pages 866–870, Chicago, Illinois.